Siebel

REST API Guide

December 2024

Siebel
REST API Guide

December 2024

Part Number: F84072-03

Authors: Siebel Information Development Team

Contents

Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at *https://docs.oracle.com/*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an email to: *oracle_fusion_applications_help_ww_grp@oracle.com*.

1 What's New in This Release

What's New in Siebel REST API Guide, Siebel CRM 24.12 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 24.12 Update

Topic	Description
Purging Old Versions of Product, Class, and Attribute	New topic. Describes how product administrators can purge or delete old versions of product, class, and attribute while retaining specified number of versions.

What's New in Siebel REST API Guide, Siebel CRM 24.1 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 24.1 Update

Topic	Description
Using the PageSize Parameter for Parent/Child/Grandchild Records	New topic. Describes how to use the `PageSize` parameter to set the limit for the number of records returned from a query for every level of the hierarchy.

What's New in Siebel REST API Guide, Siebel CRM 23.7 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 23.7 Update

Topic	Description
Using Siebel REST API to Perform CRUD Operations on Hierarchies of Siebel Object Records	New topic. Describes how to use the Siebel REST API to create, update, and delete entire hierarchies (i.e. parent, child and grandchild) of Siebel object records in a single request.

What's New in Siebel REST API Guide, Siebel CRM 23.5 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 23.5 Update

Topic	Description
Supporting RESTful Access to Siebel Business Objects Dynamically	New topic. Describes how to use the Dynamic IO feature to enable RESTful access of Business Objects without the need to create Base Integration Objects (Base IO).
Configuring REST Inbound Authentication Parameters	Modified topic. Added information about the "Allow anonymous inbound REST requests" check box, which specifies whether you want to allow inbound REST requests without authentication.

What's New in Siebel REST API Guide, Siebel CRM 23.3 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 23.3 Update

Topic	Description
Overview of the REST Outbound Proxy Business Service	Modified topic. Added the list of parameters required for outgoing REST requests using Basic Authentication to avoid an error from an external system.

What's New in Siebel REST API Guide, Siebel CRM 23.1 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 23.1 Update

Topic	Description
Added the following topics under the "Using Siebel REST API For Siebel Clinical" chapter: • *Inserting SVT from External Application to Siebel Clinical* • *Querying Sites in Siebel Clinical* • *Querying Item Library in Siebel Clinical* • *Upserting Source Data Verification in Siebel Clinical* • *Marking Visits as Planned or UnPlanned* • *Using Inbound Subject API in Siebel Clinical*	Modified chapter. This chapter describes how to use inbound web services for integration with the Clinical One platform to create and update site, subject visit template, subject enrollment, and subject activity information.

What's New in Siebel REST API Guide, Siebel CRM 22.12 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.12 Update

Topic	Description
Using Existing Integration Objects in Outbound REST Services	New topic. Describes how to use existing integration objects to support outbound web services in Siebel CRM.
Overview of the REST Outbound Filter Service	Updated topic. Added information on using OAuth with Siebel REST API.

What's New in Siebel REST API Guide, Siebel CRM 22.11 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.11 Update

Topic	Description
About Getting the Siebel REST API Specification in the OpenAPI 3.0 Standard Using Describe	New topic. Provides information about using the **`describe`** URI parameter to get the Siebel REST API specification in the OpenAPI 3.0 standard format.
About Getting the Siebel REST API Specification in the OpenAPI 2.0 Standard Using Describe	Modified topic. Updated URI for OpenAPI 2.0.
Using the Describe Parameter to Return a Catalog of Repository Objects	Modified topic. Updated URI for OpenAPI 2.0 and added example for OpenAPI 3.0.
Using the Describe Parameter to Return the Siebel Base Business Object Catalog	Modified topic. Updated URI for OpenAPI 2.0 and added example for OpenAPI 3.0.
Using the Describe Parameter to Return Business Component Metadata	Modified topic. Updated URI for OpenAPI 2.0 and added example for OpenAPI 3.0.
Using the Describe Parameter to Return Child Business Component Metadata	Modified topic. Updated URI for OpenAPI 2.0 and added example for OpenAPI 3.0.
Using the Describe Parameter to Return Methods of a Business Service	Modified topic. Updated URI for OpenAPI 2.0 and added example for OpenAPI 3.0.
Using the Describe Parameter to Return a Catalog of Paths for All Available Business Services	Modified topic. Updated URI for OpenAPI 2.0 and added example for OpenAPI 3.0.
Refreshing the Cache	Modified topic. RefreshCache API now supports versioned object key.

What's New in Siebel REST API Guide, Siebel CRM 22.10 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.10 Update

Topic	Description
Using Siebel REST API with Product Administration	Modified topic. Removed the **`SortSpec`** parameter from the request parameters description table for the following operations: • SWI Catalog Admin - QueryPage • SWI Product - QueryPage • SWI Product Class - QueryPage • SWI Product Attribute - QueryPage • SWIProductLine - QueryPage

Topic	Description
	• SWI Product Base - QueryPage • ISS Promotion Definition Loader QP - QueryPage

What's New in Siebel REST API Guide, Siebel CRM 22.9 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.9 Update

Topic	Description
Using Siebel REST Outbound Services	Modified chapter. OpenAPI 3.0 is now supported for outbound REST API.
SWI Cfg Object Broker	New topic. Describes the inbound REST API service used to query a complete set of versioned objects with focus on the Object Id, External Integration Id, and name mapping.
SWI Product Line Operations	Modified topic. Describes how to query for existing product lines using the QueryPage method.

What's New in Siebel REST API Guide, Siebel CRM 22.8 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.8 Update

Topic	Description
Using Siebel REST API for Siebel Loyalty	New chapter. Describes Siebel Representational State Transfer (REST) Services for Siebel Loyalty users and how to use them.

What's New in Siebel REST API Guide, Siebel CRM 22.7 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.7 Update

Topic	Description
About Siebel CRM REST API Supported Resources	Modified topic. Siebel REST API now supports the Siebel Business Object Entitlement Templates and its associated Siebel Business Components.

What's New in Siebel REST API Guide, Siebel CRM 22.5 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.5 Update

Topic	Description
Resetting State Attributes in Product Administration	New topic. Describes how to reset state attributes in the SWI External Integration Service business service using the ResetSessionCache method.
ISS Promotion Definition Loader QP	New topic. Describes the REST API service used to perform queries for existing product promotions using QueryPage method on the ISS Promotion WS integration object.
Using Siebel REST API For Siebel Pricing Administration	New chapter. Provides a list of REST APIs for self-service commerce using Siebel Customer Order Management.

What's New in Siebel REST API Guide, Siebel CRM 22.4 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.4 Update

Topic	Description
About URI Parameters	Modified topic. Includes the following new parameters: • ExecutionMode • ExcludeEmptyFieldsInResponse
Configuring Preprocessing and Postprocessing Steps in Inbound REST API	New topic. Describes how to configure preprocessing and postprocessing steps to facilitate any state variable changes during inbound REST API.
Using Siebel REST API to Access Siebel Repository Resources JSON Examples	Modified chapter. Includes the following REST API operations: • *Querying Records Using the ExecutionMode Parameter* • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Using Siebel REST API to Access Siebel Repository Data XML Examples	Modified chapter. Includes the following REST API operations: • *Querying Records Using the ExecutionMode Parameter* • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Using Siebel REST API to Access Siebel Business Objects JSON Examples	Modified chapter. Includes the following REST API operations: • *Querying Records Using the ExecutionMode Parameter* • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Using Siebel REST API to Access Siebel CRM Business Objects XML Examples	Modified chapter. Includes the following REST API operations: • *Querying Records Using the ExecutionMode Parameter* • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Using Siebel REST API to Access Siebel Business Services JSON Examples	Modified chapter. Includes the following REST API operation: • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Using Siebel REST API to Access Siebel Business Services XML Examples	Modified chapter. Includes the following REST API operation: • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Using Siebel REST API to Access Workflows JSON Examples	Modified chapter. Includes the following REST API operation: • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Using Siebel REST API to Access Workflows XML Example	Modified chapter. Includes the following REST API operation: • *Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter*
Overview of Generated Repository Integration Objects	Modified topic. Describes how to disable default outbound URL encoding using the disableURLEncoding parameter.

What's New in Siebel REST API Guide, Siebel CRM 22.3 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.3 Update

Topic	Description
Using the Siebel REST API to Access Workflows	New topic. Describes how to access workflows in Siebel REST API.

What's New in Siebel REST API Guide, Siebel CRM 22.2 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 22.2 Update

Topic	Description
About Importing Lists Of Values (LOVs) Using REST API	New topic. Describes how to use the LOV Import Service to load LOVs into the Siebel CRM database.

What's New in Siebel REST API Guide, Siebel CRM 21.12 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 21.12 Update

Topic	Description
About User Authentication	Modified topic. This topic has been updated to provide new information on using and configuring OAuth authentication in REST API.
Using the Siebel REST API to Access Repository Resources	Modified chapter. All information pertaining to accessing repository resources is available in: • *Using Siebel REST API to Access Siebel Repository Resources JSON Examples* • *Using Siebel REST API to Access Siebel Repository Data XML Examples*
Using the Siebel REST API to Access Business Objects	Modified chapter. All information pertaining to accessing business objects is available in: • *Using Siebel REST API to Access Siebel Business Objects JSON Examples* • *Using Siebel REST API to Access Siebel CRM Business Objects XML Examples*
Using the Siebel REST API to Access Business Services	Modified chapter. All information pertaining to accessing business services is available in: • *Using Siebel REST API to Access Siebel Business Services JSON Examples* • *Using Siebel REST API to Access Siebel Business Services XML Examples*
Using the Siebel REST API to Access Attachments	New chapter. Describes how Siebel REST API supports attaching external documents and files to records.
Returning a Promotion Definition	Removed topic. This topic is now available in the chapter: *Using Siebel REST API For Siebel Product Administration*. **Note:** Two variations of this topic are now available: • *Returning a Promotion Definition by ID* • *Returning a Promotion Definition by Name*
Using Siebel REST API For Siebel Product Administration	New chapter. Describes Siebel Representational State Transfer (REST) Services for Siebel Product Administration users and how to use them.
Overview of the REST Outbound Proxy Business Service	Modified chapter. The Business Arguments section clarifies security level operations in JSON specifications.
Overview of Generated Repository Integration Objects	Modified chapter. This topic has a new section describing how to use the SiebelHierInput parameter to allow additional integration object and integration component wrappers.
Overview of Custom Headers in REST Outbound Proxy Business Service Methods	Modified chapter. This topic provides new information on the following custom header prefixes: • HDR. • HDR_ • siebel_transport_header

What's New in Siebel REST API Guide, Siebel CRM 21.9 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 21.9 Update

Topic	Description
Considerations When Using REST Outbound in Siebel	Modified topic. You can now import JSON files using both Siebel Tools and the Web Service Wizard in Web tools.
Overview of the REST Outbound Proxy Business Service	Modified topic. Added parameters that can be used with business service method arguments.
Overview of Generated Repository Integration Objects	Modified topic. This topic has been updated to provide clearer explanations and examples.
Adding Custom Headers in REST Outbound Proxy Business Service Methods	Modified topic. Added specific examples of the data type required when creating a use property in a new custom header.

What's New in Siebel REST API Guide, Siebel CRM 21.7 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 21.7 Update

Topic	Description
Overview of OAuth 2.0 Authentication in REST API	Modified topic. OAuth 2.0 is not supported for outbound REST API.
Using the Describe Parameter to Return the OpenAPI Specification of a Business Service Method	New topic. Describes how to use the describe parameter to return the OpenAPI specification of a business service method, its input and output arguments, and integration objects, if applicable.
Using the Developer Workspace Parameter to Preview Changes to a Business Service Without Compiling to the Repository	New topic. Describes how to use the developer workspace parameter to preview any changes you make to a business service, before you deliver them to the MAIN workspace.

Topic	Description
Using the Describe Workspace Parameter to Preview the Modified OpenAPI Specification of a Business Service Method Without Compiling to the Repository	New topic. Describes how to use the developer workspace parameter to preview the modified OpenAPI specification of changes to a business service method, before you deliver them to the MAIN workspace.

What's New in Siebel REST API Guide, Siebel CRM 21.6 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 21.6 Update

Topic	Description
About Siebel CRM REST API Supported Resources	Modified topic. Siebel Loyalty business objects are also supported in Siebel REST API.
Accessing a Siebel Business Service with the matchrequestformat Parameter	Modified topic. In Siebel CRM 21.6 Update and later, the matchrequestformat parameter replaces the legacy parameter.
Using Siebel REST API For Siebel Telco	New chapter. Provides a list of REST APIs for self-service commerce using Siebel Customer Order Management.

What's New in Siebel REST API Guide, Siebel CRM 21.4 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 21.4 Update

Topic	Description
Multiple topics	In Siebel CRM 21.4 Update and later, workspace articles previously referred to as **`main`**, are now renamed to **`MAIN`**.

What's New in Siebel REST API Guide, Siebel CRM 21.3 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 21.3 Update

Topic	Description
Accessing a Siebel Business Service with the matchrequestformat Parameter	New topic. Describes how to access a Siebel CRM business service by stating the legacy parameter in the request.
Configuring the worker.properties File for Load Balancing	Obsolete topic. For more information about load balancing, see *Siebel Installation Guide* .

What's New in Siebel REST API Guide, Siebel CRM 21.2 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel REST API Guide, Siebel CRM 21.2 Update

Topic	Description
About Using the Siebel REST API	Modified topic. Changes to repository objects no longer require a restart of the EAI object manager component.
Multiple topics	In Siebel CRM 21.2 Update and later, the applicationcontainer directory has been replaced by two directories, as follows: • applicationcontainer_external (for Siebel Application Interface) • applicationcontainer_internal (for all other Siebel Enterprise components)
Multiple topics	In Siebel CRM 21.2 Update and later, in the Siebel Application Interface installation, Web artifacts for application configurations, which were formerly located in **`applicationcontainer\webapps\siebel`**, now map to **`applicationcontainer_external\siebelwebroot`**. The siebelwebroot directory contains subdirectories such as files, fonts, htmltemplates, images, migration, scripts, and smc.

2 Overview of Using the Siebel REST API

Overview of Using the Siebel REST API

This chapter provides an overview of the Siebel CRM Representational State Transfer (REST) application programming interface (API). It includes the following topics:

About Siebel CRM REST API

REST (Representational State Transfer) is a software architecture style that provides a convenient and consistent approach to requesting and modifying data. In the Siebel CRM REST system, resources are stored on the Siebel Server; a client sends a request using a HTTP verb (such as GET, POST, PUT, or DELETE) so that the Siebel Server performs a particular action (such as querying, inserting, upserting, or deleting a Siebel CRM resource), and the Siebel Server performs the action and sends a response.

REST API requests are used for migration resource discovery and for migrating data from the source environment by exporting the data, transferring the data to the target environment, and importing the data into the target environment. For information about migration resource discovery, see *Siebel Database Upgrade Guide* .

Siebel Approval Manager (SAM) has a plug-n-play architecture using REST APIs. REST APIs are used to feed data from external systems into SAM as well as write data back to those external systems. For more information, see *Siebel Approval Manager Guide* .

The Siebel REST API is provided with the Siebel Application Interface installation. For more information about installing Siebel Application Interface, see *Siebel Installation Guide* . The Siebel REST API is enabled by configuring and deploying

a Siebel Application Interface profile. The Siebel Application Interface profile can be deployed to multiple Siebel Application Interface nodes.

The Siebel REST API exposes Siebel Business Objects, Siebel Business Services, and Siebel Repository Objects. For more information about Siebel Business Objects, Siebel Business Services, and Siebel Repository Objects, see *Configuring Siebel Business Applications* .

The following aspects of Siebel REST API are aligned with general best practices of REST APIs:

- A base URI to access Siebel Server resources, for example:
  `http://Server Name:port/Siebel/v1.0/`
- Support for JSON resource representations.
- Support for XML resource representations.
- Operations on Siebel CRM resources are mapped to semantically similar HTTP methods, such as GET, PUT, POST, and DELETE.
- Hypertext links to Siebel CRM Child Business Components in the case of the Data API resources in the REST API response.
- Support for Outbound to interact with other Cloud applications that communicate through REST.

About REST Response Base URIs

REST Response Base Uniform Resource Identifiers (URI) instruct the REST API how to reach the Siebel web application layer during the inbound rest call. The REST Response Base URI is prefixed with the resource URLs that are returned by REST responses, such as child links, self-links, and so on.

The format of the base URI is as follows: `<AI_HOST>:<AI_HTTPS_PORT>/siebel/v1.0/`

- **<AI_HOST>**. Application interface hostname.
- **<AI_HTTPS_PORT>**. Application interface port number
- **Siebel.** Application name
- **v1.0.** REST version

```
<AI_HOST>:<AI_HTTPS_PORT>/siebel/v1.0/
```

The following request is a full URI to access data resources:

```
https://SampleHostname:8080/siebel/v1.0/data/Account/Account/<acc_id>/Contact/<cont_id>
```

In this example, the base URI is: `<AI_HOST><AI_HTTPS_PORT>/siebel/v1.0/`

> **Note:** If the setup has an internet-facing reverse proxy, or if the setup has a load balancer before the request reaches the application interface node, then you must use the hostname and port number of the internet-facing reverse proxy, or of the load balancer, so that the REST URLs synthesized are valid from outside the setup.

About Siebel CRM REST API Architecture

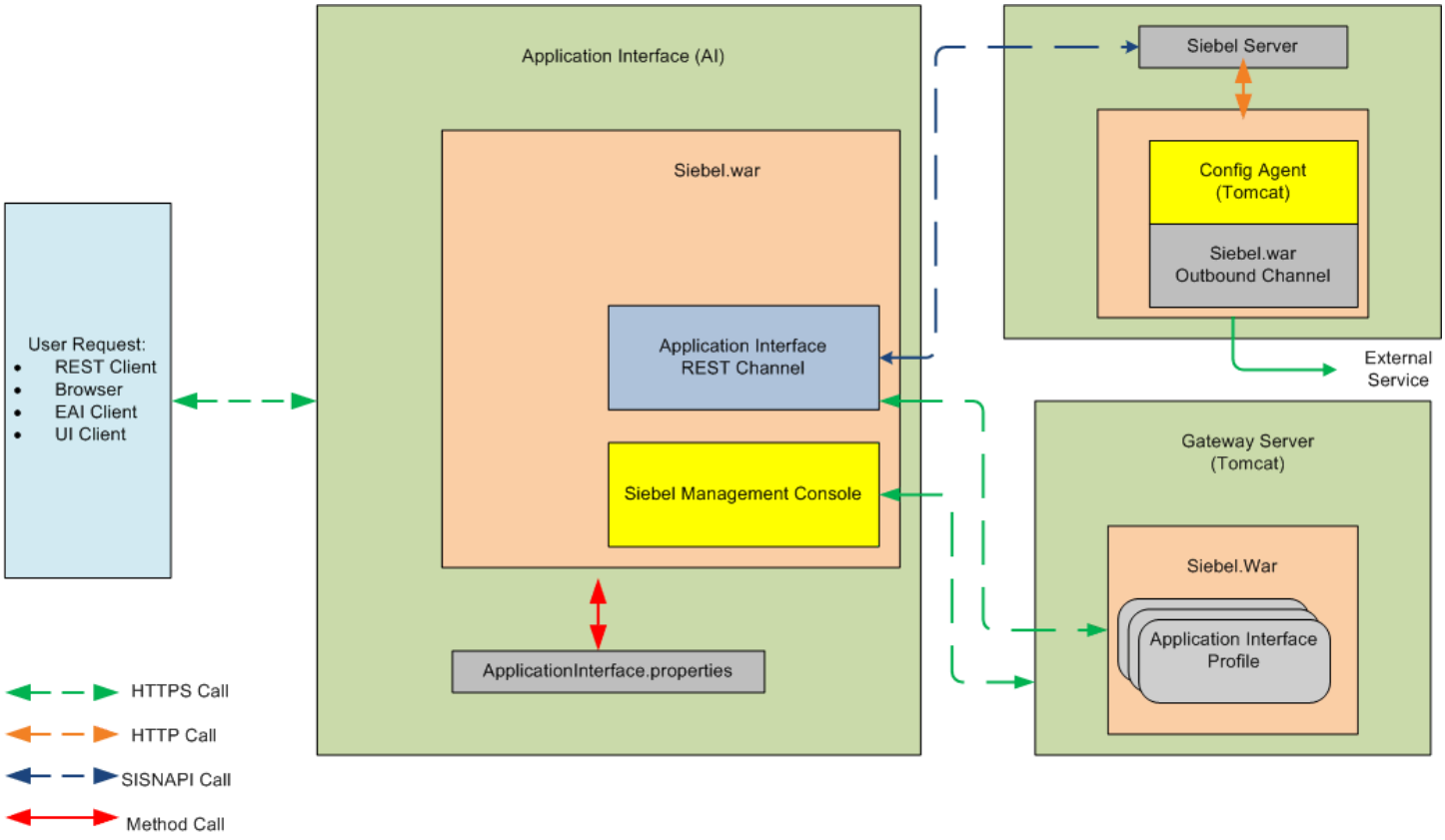
The Siebel CRM REST API services are deployed as a WAR file (siebel-rest.war) on the Siebel Application Interface. The Siebel Application Interface is a Java-based Web component that runs on the Tomcat server. The Siebel Application Interface provides the Web entry point to Siebel Business Application services. The Siebel Application Interface interacts with the Siebel Server and Gateway Server to manage and fulfill Siebel REST requests.

Each Siebel Enterprise can have multiple Siebel Application Interface nodes and can have multiple Siebel Servers but can have only one Gateway Server. Installations of Siebel Enterprise Server and Siebel Application Interface modules include deployment of WAR files into the application container and configuration of application container ports.

The Siebel application configurations are managed by the Siebel Management Console and stored in the Siebel Gateway Server.

For more information about the detailed architecture of the Siebel Application Interface (AI), see *Siebel System Administration Guide* .

The following figure shows the high-level architecture of Siebel REST within the Siebel Application Interface.



- **Requests.** Users send an application request to the Siebel Application Interface. Siebel Application Interface requests include: UI, EAI, or REST. The appropriate application channel accepts the request. Application channels include: UI Channel, EAI Channel, or REST Channel.

- **Siebel Application Interface.** The Siebel Application Interface serves as the Web server for the Siebel Business Applications. The Siebel Application Interface identifies requests for Siebel application data coming from Web clients and flags these requests for routing to a Siebel Server. When information is sent from the Siebel Server back to the Web client, the Siebel Application Interface helps complete the composition of the Web page for forwarding to the client.
- **Siebel Management Console.** The Siebel Management Console is a Web-based application that allows users to configure various Siebel Application components and save the configurations in the Siebel Gateway Server.
- **REST Channel.** The REST channel is one of three application channels that provide an endpoint responsible for receiving and processing user requests. The REST Channel is responsible for serving the REST requests. Other application channels include the UI channel and EAI channel.
- **Siebel Server.** Siebel Server functions as an application server and is composed of server components.
- **Siebel Gateway Server.**This provides the dynamic address registry for Siebel Servers and server components, and also for Siebel Application Interface. The Siebel Application Interface configurations are also stored and managed in the Siebel Gateway Server.
- **Outbound Channel.** The outbound channel forwards requests to external Web services.

For more information, see *Siebel Installation Guide* , *Siebel Deployment Planning Guide* , and *Siebel System Administration Guide* .

About Siebel CRM REST API Requests and Responses

A request can include the following information:

- A request URI. For more information about URI formats, see *About Siebel CRM REST API URI Formats*.
    - The request URI contains the base URI signifying what category of Siebel resources to invoke. Siebel resources include: Business Objects, Repository Objects, and Business Services.
    - The request URI contains the object type of the invoked Siebel CRM resource. For example, Account Business Component under Account Business Object. For more information about the supported Siebel CRM resources, see *About Siebel CRM REST API Supported Resources*.
- The HTTP method that you use to perform a REST API operation (query, insert, upsert, or delete) on the Siebel CRM Server. For more information about supported HTTP methods, see *About Supported HTTP Methods*.
- Header information to define the parameters of the interaction with the Siebel Server and the information and format you want in the response. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.

After the Siebel CRM Server processes the request, the server sends back a response in JSON or XML format, based on the requested content type. The response body contains the results of the REST API call and also additional information based on what was specified in the request.

The response can include the following information:

- A HTTP status code that indicates whether the request was successful or failed. For more information about HTTP status codes, see *About Standard HTTP Status Codes and Error Messages*.
- A list of Siebel proprietary field and value pairs.
- If the request failed, then the response body includes additional information about the error. For additional information about HTTP error status codes, see *About Standard HTTP Status Codes and Error Messages*.

- Hypertext links to Siebel CRM child resources, in the REST API response, such as links to Child Business Components in the case of the Data API. For more information about links, see *About Siebel CRM REST API Response Links*.

About Siebel CRM REST API URI Formats

The Siebel CRM REST API exposes Siebel CRM Business Objects, Business Services, and Repository Objects.

Siebel CRM REST API executes resource requests for each resource category using the following HTTP URI formats:

- Siebel CRM REST API basic URI for Siebel Business Objects:
  `https://Server Name:port/Siebel/v1.0/data/`
- Siebel CRM REST API basic URI for Siebel Business Services:
  `https://Server Name:port/Siebel/v1.0/service/`
- Siebel CRM REST API URI for a Siebel Repository Resource:
  `https://ServerName:port/Siebel/v1.0/workspace/<your workspace name>/`
- Siebel CRM REST API URI to execute Siebel Workflow:
  `https://ServerName:port/Siebel/v1.0/workflow/<your workflow name>/`

The values for these URIs are:

- **Server Name:port:** Indicates the name of the server and port hosting the Siebel REST API services.
- **siebel:** Indicates the product name for the REST API.
- **version:** Indicates the current version number, 1.0, of the REST API.
- **data:** Indicates the requested resource is a Siebel Business Object. For more information about Siebel Business Objects, see *Configuring Siebel Business Applications* .
- **service:** Indicates the request resource is a Siebel Business Service. For more information about Siebel Business Services, see *Configuring Siebel Business Applications* .
- **workspace:** Indicates the request resource is Siebel Repository Data. For more information about workspaces, see *Using Siebel Tools* . For more information about Siebel Repository Objects, see *Configuring Siebel Business Applications* .
- **<your workspace name>:** Indicates the name of your repository workspace.

About URI Parameters

When constructing a URI, there are a number of optional parameters available to manage response results. Some parameters are described in the following table.

Parameter	Description
PageSize	Used only for the GET operation. The PageSize parameter is the integer that tells the Siebel Server how many records to return. If you query all the Siebel contacts whose last name starts with the letter A and you do not want to get too many records (for performance reasons), then you can restrict the number of records returned. You restrict the number of records returned by setting the PageSize parameter to a

Parameter	Description
	reasonable number. The default value is 10. If you do not set a value for this parameter, then query will return only 10 records. All records that match the search criteria are returned. For example, PageSize=20 returns only twenty contact records, even if more exist in the Siebel database. If fewer records exist that match that search criteria, then all records are returned (but no more than twenty). **Note:** It is recommended that you retrieve the lowest number of records required for any one call. The more records that are returned, the larger the message and the slower the response. The maximum number of records cannot exceed 100.
StartRowNum	Used only for the GET operation. The StartRowNum parameter is used when there is a need to start returning records at a specific row. For example, StartRowNum=100 starts at row 100 of the record set. The first number in a record set is zero, therefore, this request starts at record 99 (given you start counting from one for the first record). The default value and returns the records from beginning. This parameter is useful for paging through a record set N records at a time. For example, if there are 100 records in a record set, but you want to retrieve only ten at a time, then enter StartRowNum=0 and PageSize=10 on the first call, then StartRowNum=10 on the next call, then StartRowNum=20 on the next call, and so on.
fields	Used for only GET or Query operations to specify a comma-separated list of property names (fields) that are required in the REST API response. The response contains only the files given in this list irrespective of fields available in the source being queried.
searchspec	Used to include search specifications in the REST API response.
ViewMode	Used as an access control that controls users access to data and application functionality. Values include: • Personal • Sales Rep • Organization • Group • Catalog For more information about access controls, see *About Access Controls for Siebel Business Component REST Requests*.
childlinks	Used for only GET or Query operations to specify a comma-separated list of child Business Components that require links returned in the REST API response. The response returns only links to child objects specified in this parameter value. For more information on the childlinks parameter, see *Querying for a Siebel CRM Repository Resource To Return Only One Child Link*.
uniformresponse	Used for only GET or Query operations to specify a consistent interface to the consumers and enable them to use a single parser for responses of requests of similar type. The response returns one or more records wrapped in an array. It is passed as a query parameter with a case insensitive value for the flag as Y or Yes. For example, URI: `data/Account/Account/?searchspec=([Location] LIKE 'HQ' AND [Account Status]='Active')&uniformresponse=y`

Parameter	Description
	For examples of **uniformresponse**, see *Using Siebel REST API to Access Siebel Business Objects JSON Examples*.
workspace&version	Used to set the developer or integration workspace name and corresponding version in REST or SOAP URI, such as **workspace=dev_sadmin&version=7**, to indicate opening the seventh version of the **dev_sadmin** developer workspace. By using this parameter, you can see your changes to the metadata even before delivering them. An example of metadata change is inactivating an object in developer workspace and previewing this change before delivery in the same workspace or after delivery in integration workspace. In this way, you can switch between metadata of different branches without invalidating sessions. The broad combination of workspace and version yield the following results: • If you indicate both workspace and its version, such as **workspace=dev_sadmin&version=7**, then the specific version of requested workspace displays. You can either inspect developer-bound changes or view integration changes for a specific version, allowing you to navigate newer or older versions to access required metadata. • If you indicate only a workspace name, such as **workspace=dev_sadmin**, then the latest version of the requested workspace displays. • If you do not indicate this parameter then the URI will work with the default integration branch configured for the EAI object manager component used for REST. For examples of this parameter for REST API, see *Using the Developer Workspace Parameter to Preview Changes Without Compiling to the Repository*. For similar examples for SOAP, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .
ExecutionMode	Used with a GET operation where a query can select more than 9900 records. If you use a regular GET REST API to select more than 9900 records, then it will fail and will cause an error. To execute this operation correctly, you must pass the ExecutionMode query parameter. The ExecutionMode parameter can take one of the following values: • **BiDirectional.** This is the default value. • **ForwardOnly.** This value is more efficient than Bidirectional. It is recommended that you use Forward Only in cases where you must process a large number of records in the forward direction only, for example report generation. For operations that are likely to return more than 10000 records, use ForwardOnly to avoid errors. In the following example ForwardOnly is used with the ExecutionMode parameter: **http://ServerName:port/siebel/v1.0/data/Account/Account?PageSize=100&StartRowNum=9950&ExecutionMode=ForwardOnly** **Note:** When using the query parameter ExecutionMode with ForwardOnly, do not use the recordcountneeded parameter with the value set to true. The query recordcountneeded=true works only with the value BiDirectional. For more information on the ExecutionMode parameter, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .
excludeEmptyFieldsInResponse	Used to remove empty fields in REST API responses. As a result, the response body size is reduced. By default, if you do not use this parameter, both empty and complete fields are displayed in the REST response. This applies to data, service, workspace, and workflow REST APIs. The excludeEmptyFieldsInResponse query parameter takes the following values: • **True, yes, Y.** These values remove all empty fields present in the REST response.

Parameter	Description
	• **False, no, N.** These values do not remove any fields from the REST response. This is the default value. In the following example the excludeEmptyFieldsInResponse parameter is set to true: **http://ServerName:port/siebel/v1.0/data/Account/Account?excludeEmptyFieldsInResponse=true**

The syntax for using URI parameters is the parameter name followed by an equal sign (=) with the value of the parameter, and each parameter is separated from other parameters by an ampersand (&). For example, if you want to set the PageSize parameter to 100 and the StartRowNum parameter to 0 (zero), then you enter: Pagesize=100&StartRowNum=0

About Siebel CRM REST API Supported Resources

A REST API resource is a piece of information, such as a data record or a collection of records. Each Siebel CRM REST API resource is identified by a named URI, and it is accessed using standard HTTP methods. For more information about URI formats, see *About Siebel CRM REST API URI Formats*. For more information about standard HTTP methods, see *About Supported HTTP Methods*.

The Siebel CRM REST API supports the following Siebel resources and collections:

- Siebel Business Service Methods that have been configured for Siebel REST API access for users with a given Siebel Responsibility. For more information about configuring Siebel Business Services, see *Configuring Business Service Methods for REST Access*.
- Siebel Repository Object Types and Siebel Repository Object Instances. All repository objects that are supported for access through workspaces are accessible through the Siebel Repository REST API. Before you begin, a workspace has to be created before performing any repository operations and the name of the workspace has be mentioned in the REST API requests. To determine if a repository object is workspace-enabled, see the topic Enabling Workspaces in Siebel Tools and Editing the Repository Objects in *Using Siebel Tools* .

- Siebel Business Component records. The Siebel Business Components under the following Siebel Business Objects are available through the Siebel REST API:
    - Access Group
    - Account
    - Action
    - Asset Management
    - Campaign
    - Catalog
    - Contact
    - Correspondence
    - Employee
    - Entitlement Templates
    - Expense
    - Fund
    - Household
    - Incentive Compensation Plan
    - Internal Product
    - List Mgmt
    - Loyalty Member
    - Loyalty Transaction
    - Offer
    - Opportunity
    - Order Entry
    - Payments
    - Position
    - Price List
    - Project
    - Proposal
    - Quote
    - Service Agreement
    - Service Request
    - Solution
    - Territory Management
    - Time Sheet
    - Usage Pattern Tracking

About Siebel CRM REST API Unsupported Resources

The following repository types and resources are neither accessible nor supported in Siebel REST API.

- Table
- Task
- Dock Object
- Bitmap
- Repository
- Schema
- Server Component type
- Schema Maintenance
- Workflow Policy
- Type
- Project
- EIM Table
- Assignment Criteria
- Assignment Attribute

About Supported HTTP Methods

The following table contains the HTTP methods supported by the Siebel REST API and the corresponding Siebel CRM operation.

HTTP Verb	Siebel Operation	Description
GET	Query	The GET method retrieves a Siebel CRM resource.
POST	Insert	The POST method creates a new Siebel CRM data resource, or executes a business service, or invokes a workflow based on an API type.
PUT	Upsert	The PUT method upserts a Siebel CRM resource.
DELETE	Delete	The DELETE method deletes a Siebel CRM resource.

About Supported HTTP Header Fields

Certain HTTP header fields define the operating parameters of the REST API transaction with Siebel CRM.

The following table contains the HTTP header fields supported by the Siebel REST API.

HTTP Header Field Name	Description	Example
Authorization	The HTTP request header field that indicates the type of authorization. Options include: • Basic, if the Authentication type configured in siebsrvr.properties is Basic or SSO • Bearer, if the Authentication type configured in siebsrvr.properties is OAuth	Authorization: Basic
Content-Type	The HTTP request and response header field that indicates the content type of the message body. Content-Type decides the format of response for all requests. The Siebel REST API supports JSON and XML encoding for the request body. The default value is application/JSON. The Content-Type field is used with POST, PUT, and GET requests. When submitting a POST or PUT request, you typically supply a body with the request. You can indicate the format of the response by setting the HTTP Content-Type header on the request. For GET requests, the content type is used to determine the format of response, either XML or JSON.	Content-Type: application/json

About Standard HTTP Status Codes and Error Messages

Siebel CRM REST API uses standard HTTP status codes, as described in the following table, to indicate the success or failure of API calls. When an error occurs or when a response is successful, the response header contains a HTTP code

and the response body usually contains a message accompanying the HTTP response code with additional information about the error.

HTTP Code	Message	Description
200	OK	The request successfully executed and the response has content.
204	No Content	The request successfully executed, but the content is unavailable. For example, the content was deleted.
401	Unauthorized	The request had invalid authorization credentials.
404	Not Found	The requested resource was not found because of an invalid object name.
405	Not Allowed	The request is not allowed.
406	Not Acceptable	The resource identified by the request is capable of generating only response entities that have content characteristics that are not acceptable according to the accept headers sent in the request.
415	Unsupported Media Type	The data format of the request body, specified in the Content-Type header, is unsupported by the targeted resource.
500	Internal Server Error	The server encountered an unexpected error, preventing it from fulfilling the request.

About Siebel CRM REST API Response Links

A link in the Siebel CRM REST API response contains a location of a resource and metadata about that resource.

The Siebel CRM REST API response can include the following types of links:

- **self link.** The original URL that generated the response.
- **canonical link.** The URL for the same resource as the top-level resource. If you are already viewing the resource as a top-level resource, then this URL is the same as self. If this resource is unavailable as a top-level resource, then this link shows child resource context.
- **parent link.** The URL for the parent resource details. This URL of the parent resource is returned in the response only when retrieving details about a child resource.
- **child link.** The URL for the child resource details. The URL returns the path to retrieve each child collection for this record. The href attribute contains the child type. A response can return several child links.
- **association link**. The URL of a specific resource included in the response. There can be many association links.

Each Siebel CRM REST API response link can include the following types of attributes:

- **rel.** Indicates the relationship of the linked resource to the current resource that contains the list of links. Values include: Self, Parent, Child, and Canonical.
- **href.** Indicates the fully qualified location URL of the linked resource.

About User Authentication

Siebel CRM supports the following mechanisms that the client uses to authenticate user credentials:

- Basic authentication over SSL. This is User ID and Password based authentication. The Base64 encoded value of the User ID and Password must be included in the Authorization header.
- SSO, which can be either:
  - Pre-existing SSO mechanisms used for Siebel Applications or EAI.
  - SAML based SSO mechanisms. This is Identity Provider-Initiated Single Sign-On Authentication. For more information, see *Siebel Security Guide* .
- OAuth user using OAuth 2.0. For more information about OAuth 2.0, see *Siebel Security Guide* .

Authentication parameters are configured in the Siebel Application Interface Profile. For information about REST Inbound Authentication parameters, see *Configuring REST Inbound Authentication Parameters*. For information about configuring the Siebel Application Interface Profile, see *Siebel Installation Guide* .

About Getting the Siebel REST API Specification in the OpenAPI 2.0 Standard Using Describe

Siebel REST APIs use the OpenAPI Specification (formerly the Swagger Specification) to define formats for REST requests and responses to the Siebel REST API servers.

You use the OpenAPI Specification Describe URI parameter to provide additional metadata information in a REST API response and request. It returns a JSON object that contains the attributes, actions, and links defined in the REST resource definition.

This parameter allows you to discover additional metadata on the following:

- Siebel Business Objects. The Siebel Business Objects catalog contains a list of all Business Objects exposed as Base Integration Objects. The following is an example of a Siebel Business Objects URL request with a OpenAPI Specification Describe URI parameter: **host-name/siebel/v1.0/data/describe?openapiversion=2**
- Siebel Business Services. The Siebel Business Services catalog contains list of all Siebel Business Service names, methods defined for Business Services, and links to each Business Service. The following is an example of a Siebel Business Services URL request with a OpenAPI Specification Describe URI parameter: **host-name/siebel/v1.0/service/describe?openapiversion=2**
- Siebel Repository Objects. The Siebel Repository Objects catalog contains lists of all repository types and catalog links to their children. The following is an example of a Siebel Repository Objects URL request with a OpenAPI Specification Describe URI parameter: **host-name/siebel/v1.0/workspace/MAIN/describe?openapiversion=2**

In the response, pagination is implemented for a list of Business Objects, a list of Repository Objects, and a list of Business Services but not for the associated children.

The following table contains the OpenAPI Objects supported by the Siebel REST API.

Field Name	Description
swagger	Required. Specifies the specification version being used.
info	Required. Provides metadata about the API.
schemes object	The transfer protocol of the API. Values MUST be from the list: "http", "https", "ws", "wss". If the schemes object is not included, the default scheme to be used is the one used to access the Swagger definition itself.
securityDefinitions	Security scheme definitions that can be used across the specification.
externalDocs	Additional external documentation.
host	The host (either the name or IP) serving the API. This must be the host only and does not include the scheme nor sub-paths. It may include a port. If the host is not included, the host serving the documentation is to be used (including the port). The host does not support path templating.
basePath	The base path on which the API is served, which is relative to the host. If it is not included, the API is served directly under the host. The value must start with a leading slash (/). The basePath does not support path templating.
definitions	An object to hold data types that are produced and consumed by operations.
tags	A list of tags used by the specification with additional metadata. The order of the tags can be used to reflect on their order by the parsing tools. Not all tags that are used by the Operation Object must be declared. The tags that are not declared may be organized randomly or based on the tools' logic. Each tag name in the list must be unique.
paths	Required. The available paths and operations for the REST API.
security	A declaration of which security schemes are applied for the API as a whole.
parameters	A list of the parameters for the endpoint.
responses	A lists of the responses from the REST API request.

About Getting the Siebel REST API Specification in the OpenAPI 3.0 Standard Using Describe

Siebel REST APIs use the OpenAPI Specification (formerly the Swagger Specification) to define formats for REST requests and responses to the Siebel REST API servers.

You use the OpenAPI Specification Describe URI parameter to provide additional metadata information in a REST API response and request. It returns a JSON object that contains the attributes, actions, and links defined in the REST resource definition.

This parameter allows you to discover additional metadata on the following:

- Siebel Business Objects. The Siebel Business Objects catalog contains a list of all Business Objects exposed as Base Integration Objects. The following is an example of a Siebel Business Objects URL request with a OpenAPI Specification Describe URI parameter: **host-name/siebel/v1.0/data/describe**
- Siebel Business Services. The Siebel Business Services catalog contains list of all Siebel Business Service names, methods defined for Business Services, and links to each Business Service. The following is an example of a Siebel Business Services URL request with a OpenAPI Specification Describe URI parameter: **host-name/siebel/v1.0/service/describe**
- Siebel Repository Objects. The Siebel Repository Objects catalog contains lists of all repository types and catalog links to their children. The following is an example of a Siebel Repository Objects URL request with a OpenAPI Specification Describe URI parameter: **host-name/siebel/v1.0/workspace/MAIN/describe**

In the response, pagination is implemented for a list of Business Objects, a list of Repository Objects, and a list of Business Services but not for the associated children.

The following table contains the OpenAPI Objects supported by the Siebel REST API.

Field Name	Description
swagger	Required. Specifies the specification version being used.
info	Required. Provides metadata about the API.
schemes object	The transfer protocol of the API. Values MUST be from the list: "http", "https", "ws", "wss". If the schemes object is not included, the default scheme to be used is the one used to access the Swagger definition itself.
securityDefinitions	Security scheme definitions that can be used across the specification.
externalDocs	Additional external documentation.
host	The host (either the name or IP) serving the API. This must be the host only and does not include the scheme nor sub-paths. It may include a port. If the host is not included, the host serving the documentation is to be used (including the port). The host does not support path templating.

Field Name	Description
basePath	The base path on which the API is served, which is relative to the host. If it is not included, the API is served directly under the host. The value must start with a leading slash (/). The basePath does not support path templating.
definitions	An object to hold data types that are produced and consumed by operations.
tags	A list of tags used by the specification with additional metadata. The order of the tags can be used to reflect on their order by the parsing tools. Not all tags that are used by the Operation Object must be declared. The tags that are not declared may be organized randomly or based on the tools' logic. Each tag name in the list must be unique.
paths	Required. The available paths and operations for the REST API.
security	A declaration of which security schemes are applied for the API as a whole.
parameters	A list of the parameters for the endpoint.
responses	A lists of the responses from the REST API request.

About Access Controls for Siebel Business Component REST Requests

Access Controls refer to the set of Siebel Business Applications mechanisms that control users access to data and application functionality. For more information about access controls, see *Siebel Security Guide* .

Siebel CRM implements access controls for Business Component data by using the ViewMode query parameter. The Business Component ViewMode query parameter defines the access control for a Business Component in a view and decides which users can access what portion of the data. For more information about View Mode, see *Siebel Security Guide* .

The following fields in the BusComp View Modes list in Siebel Tools determine the allowed visibility for a Business Component.

- **Owner Type.** This field specifies the party type, with one exception (described in the following list), that is used to determine whether a user is associated with a record. This field value specifies the owner of the records in the current view mode. You can use the following owner types:
  - **Person.** The access control can be based on the user's Person record.
  - **Position.** The access control can be based on the position of the user.
  - **Organization.** The access control can be based on the organization of the user, as determined by the organization to which the user's current position belongs.
  - **Group.** The access control can be based on membership in access groups that have access to particular catalogs and categories.
  - **Catalog Category.** Catalog Category is not a party type. Access can be restricted to all of the data in all of the categories across catalogs to which the user has access. This data includes data in public categories and data in private categories to which the user's access groups have access. The user sees a flat, or uncategorized, list of data.
- **Name.** The name typically suggests the View Mode.
  - **Personal.** This name is typically used when Owner type is Person.
  - **Sales Rep.** This name is typically used when Owner type is Position.
  - **Organization.** This name is typically used when Owner type is Organization.
  - **Group.** This name is typically used when Owner type is Group.
  - **Catalog.** This name is typically used when Owner type is Catalog.

You can only use access controls for Siebel REST API GET requests. You can use an access control by adding the ViewMode= parameter to your REST API GET request URL. The URL format is same for both JSON and XML REST requests. The ViewMode query parameter used in the URL is case sensitive. If the ViewMode query parameter is not specified in the URL, by default ViewMode=Sales Rep is used. ViewMode=All is unsupported for REST requests.

You can use the ViewMode query parameter with the following objects:

- **Siebel Business Objects.** The Siebel Business Objects catalog contains a list of all Business Objects exposed as Base Integration Objects. The following is an example of a Siebel Business Objects URL request with a ViewMode query parameter:

  host-name/siebel/v1.0/data/Account/Account/?ViewMode=Sales Rep
- **Siebel Repository Objects.** The Siebel Repository Objects catalog contains lists of all repository types and catalog links to their children. The following is an example of a Siebel Repository Objects URL request with a ViewMode query parameter:

  host-name/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ViewMode=Personal

About Importing Lists Of Values (LOVs) Using REST API

The LOV Import Service provides a mechanism to load LOVs into the Siebel CRM database, either individually or in bulk. This capability is available in both the design repository (DR) environment for development environments and the runtime repository (RR) for test or production environments.

The following defines the LOV Import Service REST API:

- **URI: http://ServerName:port/siebel/v1.0/service/LOV Import Service/Upsert**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{"body":{
 "WorkspaceName":"<Developer Workspace>",
 "ParentWorkspaceName":"<Parent Integration Workspace",
 "SiebelMessage":{
 "MessageId":"",
 "MessageType":"Integration Object",
 "IntObjectName":"List Of Values",
 "IntObjectFormat":"Siebel Hierarchical",
 "ListOfList Of Values":{
 "List Of Values":[
 {
 "Order By":"1",
 "Name":"<Language Independent Code>",
 "Replication Level":"All",
 "Type":"<LOV TYPE>",
 "Translate":"N",
 "Multilingual":"<Y/N>",
 "Language":"<LANG_CD>",
 "Active":"Y",
 "Description":"<Description>",
 "Sub Type":"",
 "Value":"<Language Dependent Code>"
 },…
```

The following table describes the request message fields associated with this API.

Name	Description
WorkspaceName	The workspace where the LOV is created.
ParentWorkspaceName	The parent integration workspace for the WorkspaceName entry where the LOVs are created.
Name	The language-independent code.
Type	The LOV type.
Multilingual	If the LOV_TYPE record is MLOV, then set this entry to Y. For all other LOVs, including all non-LOV_TYPE records, set this entry to N.
Language	Used for only GET or Query operations to specify a comma-separated list of child Business Components that require links returned in the REST API response. The response returns only links to child objects specified in this parameter value. For more information on the childlinks parameter, see *Querying for a Siebel CRM Repository Resource To Return Only One Child Link*.
Description	The three letter language code for the language, for example, ENU or FRA.

Name	Description
Value	The display value for the language-independent code.

The following JSON example shows a basic request payload structure that will create a new LOV TYPE called REST_LOV_TYPE_TEST, under which it also creates two LOVs, LOV #1 and LOV #2:

```
{
 "body": {
 "WorkspaceName": "dev_sadmin_LOVExample",
 "ParentWorkspaceName": "MAIN",
 "SiebelMessage": {
 "MessageId": "",
 "MessageType": "Integration Object",
 "IntObjectName": "List Of Values",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfList Of Values": {
 "List Of Values": [
 {
 "Order By": "1",
 "Name": "REST_LOV_TYPE_TEST",
 "Replication Level": "All",
 "Type": "LOV_TYPE",
 "Translate": "Y",
 "Multilingual": "N",
 "Language": "ENU",
 "Active": "Y",
 "Description": "Creates an LOV_TYPE named REST_LOV_TYPE_TEST",
 "Sub Type": "",
 "Value": " REST_LOV_TYPE_TEST "
 },
 {
 "Order By": "1",
 "Name": "LOV #1",
 "Replication Level": "All",
 "Type": "REST_LOV_TYPE_TEST",
 "Translate": "Y",
 "Multilingual": "N",
 "Language": "ENU",
 "Active": "Y",
 "Description": "Creates LOV named "LOV #1" of LOV TYPE "REST_LOV_TYPE_TEST"",
 "Sub Type": "",
 "Value": "LOV #1"
 },
 {
 "Order By": "2",
 "Name": "LOV #2",
 "Replication Level": "All",
 "Type": "REST_LOV_TYPE_TEST",
 "Translate": "Y",
 "Multilingual": "N",
 "Language": "ENU",
 "Active": "Y",
 "Description": "Creates LOV named "LOV#2" of LOV TYPE "REST_LOV_TYPE_TEST"",
 "Sub Type": "",
 "Value": "LOV #2"
 }
 ]
 }
 }
}
```

```
}
```

The following considerations apply:

- If you are loading a new LOV_TYPE, you must create the LOV_TYPE record first, then create the detail records afterwards.
- To load more than one LOV at a time, create a new List Of Values section in the request payload for each LOV.
- LOVs are automatically created in all integration workspaces and main workspaces, which is consistent with how LOVs are created in the user interface or EIM.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "ErrorContextSearchSpec": "",
 "OMErrorCode": "",
 "ErrorSymbol": "",
 "OMErrorSymbol": "",
 "ErrorCode": "0x0",
 "ErrorContextIntComp": "",
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "List Of Values",
 "MessageType": "Integration Object",
 "List Of Values": [
 {
 "Description": "Creates an LOV_TYPE named REST_LOV_TYPE_TEST",
 "Type": "LOV_TYPE",
 "Language": "ENU",
 "Id": "9SIA-D8H17",
 "Active": "Y",
 "Name": "REST_LOV_TYPE_TEST",
 "Translate": "Y",
 "Order By": "1",
 "Multilingual": "N",
 "Value": " REST_LOV_TYPE_TEST ",
 "Replication Level": "All",
 "Sub Type": ""
 },
 {
 "Description": "Creates LOV named ï¿½LOV #1ï¿½ of LOV TYPE ï¿½REST_LOV_TYPE_TESTï¿½",
 "Type": "REST_LOV_TYPE_TEST",
 "Language": "ENU",
 "Id": "9SIA-D8H1F",
 "Active": "Y",
 "Name": "LOV #1",
 "Translate": "Y",
 "Order By": "1",
 "Multilingual": "N",
 "Value": "LOV #1",
 "Replication Level": "All",
 "Sub Type": ""
 },
 {
 "Description": "Creates LOV named ï¿½LOV#2ï¿½ of LOV TYPE ï¿½REST_LOV_TYPE_TESTï¿½",
 "Type": "REST_LOV_TYPE_TEST",
 "Language": "ENU",
 "Id": "9SIA-D8H1N",
```

```
        "Active": "Y",
        "Name": "LOV #2",
        "Translate": "Y",
        "Order By": "2",
        "Multilingual": "N",
        "Value": "LOV #2",
        "Replication Level": "All",
        "Sub Type": ""
        }
        ]
        }
      }
```

The following table describes the response details for common failed requests:

Name	Siebel Error	Solution
404	{"ERROR":"Business Service 'LOV Import Service' does not exist.(SBL-EAI-04299)"}	Run the RepositoryUpgrade utility or manually create the LOV Import Service as described in the Monthly Update Guide.
403	{"ERROR":"Access to Resource LOV Import Service of type Business Service is denied(SBL-DAT-00825)"}	Grant the user access to the LOV Import Service / UpsertLOV business service and method,
Other	This refers to any other non-200 status code responses.	To diagnose other non-200 responses, see *About Standard HTTP Status Codes and Error Messages*.

Related Books

For more information about how LOVs work in workspaces in Siebel Web Tools or Siebel Tools, see *Using Siebel Tools* .

For more information about working with LOVs in the Siebel application user interface and about rolling back to a prior version of the runtime repository, see *Siebel Applications Administration Guide* .

For information about importing LOVs using EIM, see *Siebel Enterprise Integration Manager Administration Guide* .

For more information about performing repository migrations, see *Siebel Database Upgrade Guide* .

3 Getting Started with the Siebel REST API

Getting Started with the Siebel REST API

This chapter provides an overview of how to get started with using the Siebel CRM REST API. It contains the following topics:

- *About Setting Up the Siebel CRM REST API*
- *Using Siebel Management Console to Configure a Siebel Application Interface Profile*
- *Configuring Business Service Methods for REST Access*
- *Configuring Integration Objects for REST API Data Access*

About Setting Up the Siebel CRM REST API

The Siebel REST API is implemented as part of the Siebel Application Interface installation. You must create and deploy a Siebel Application Interface profile before using REST related components on the Siebel server. For more information about Siebel Application Interface, see *Siebel Installation Guide* .

Before you begin using the Siebel CRM REST API, you must perform the tasks described in this topic. Many of these tasks are described in *Siebel Deployment Planning Guide* .

1. Review all documented hardware and software requirements.

   For more information, see the Certifications tab on My Oracle Support.
2. The latest version of Siebel Enterprise Server software must be installed.

   For more information about performing a new Siebel Enterprise Server software installation, see *Siebel Installation Guide*.
3. Install Siebel Application Interface.

   For more information about installing Siebel Application Interface, see *Siebel Installation Guide* .
4. (Optional) Configure preprocessing and postprocessing steps for inbound REST API. For more information, see *Configuring Preprocessing and Postprocessing Steps in Inbound REST API*.
5. Configure a Siebel Application Interface profile.

   For more information about configuring a Siebel Application Interface profile, see *Siebel Installation Guide* .

   For information about Siebel REST API configuration parameters that must be configured when you create the Siebel Application Interface Profile, see *Using Siebel Management Console to Configure a Siebel Application Interface Profile*.
6. Deploy your Siebel Application Interface profile.

   For more information about deploying a Siebel Application Interface profile, see *Siebel Installation Guide* .

7. Determine your sizing requirements.

   Based on your sizing requirements, install and configure your application interface nodes. It is recommended that you perform load balancing to distribute complex tasks among multiple Java Web Containers. For more information about load balancing, see *Siebel Installation Guide* .
8. Configure Business Service methods for REST access. For more information about configuring Business Service methods, see *Configuring Business Service Methods for REST Access*.
9. Configure integration objects for REST API data access. For more information about configuring integration objects, see *Configuring Integration Objects for REST API Data Access*.

Configuring Preprocessing and Postprocessing Steps in Inbound REST API

Sometimes resources, such as business objects, business services and workflows, that are accessed using REST API may change state variables, such as profile attributes, and session global values as part of a custom implementation. As a result, this can affect the functionality of subsequent business service or business object APIs that depend on these state variables in the same session.

State variables are specific to each implementation, and because the REST API framework does not identify any change to state variables, the REST API framework does not undo state changes automatically. To ensure consistent and correct results for all APIs, there is a need to reverse or reset the state changes implemented by these resources in REST API.

Preprocessing and postprocessing steps can be used for all inbound data, service, workspace and workflow REST APIs. They can assist you when you need to perform setup and cleanup operations before and after invoking the REST API, therefore not affecting subsequent REST API calls on the same session. Preprocessing steps are called before the Object Manager instance processing starts, the REST API is executed, and then the postprocessing steps are called after the Object Manager instance is completed.

Note: You cannot configure preprocessing and postprocessing steps for OpenAPI specification API calls.

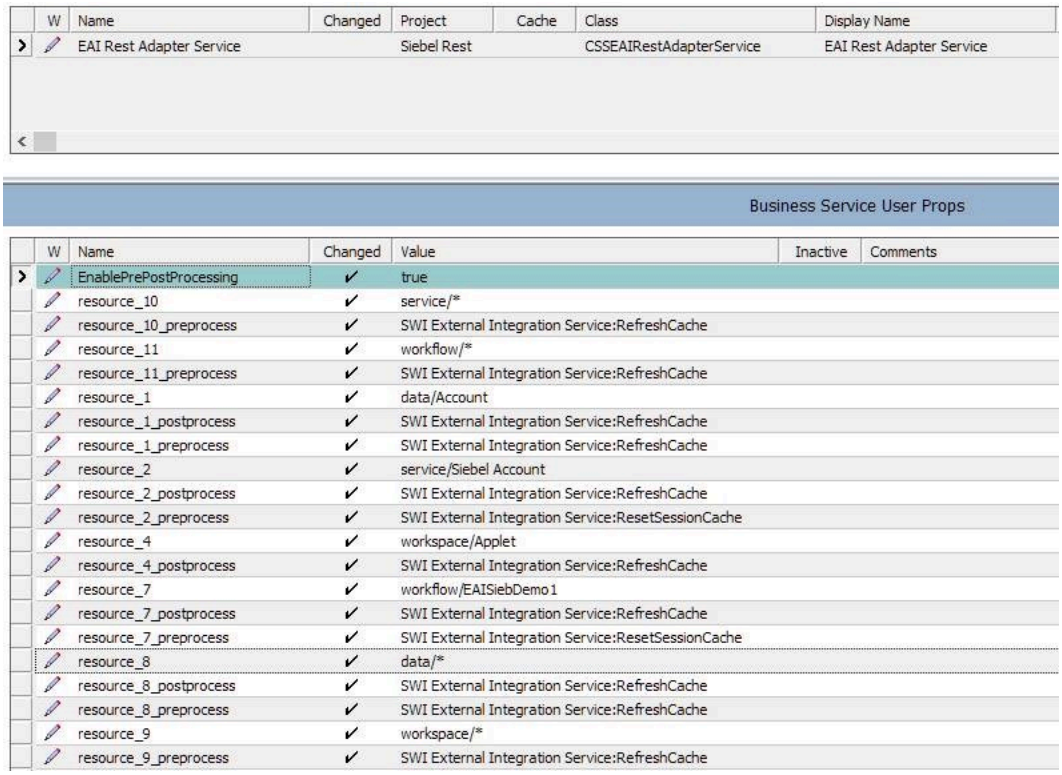
In order to enable preprocessing and postprocessing steps, you must define the following business service user property on the EAI Rest Adapter Service business service in Web Tools, and deliver the workspace with business service user property changes:

- **Business service user property name.** EnablePrePostProcessing
- **Associated business service.** EAI Rest Adapter Service
- **User property value.**True.

  Note: For more information about defining business service user properties, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

Preprocessing and postprocessing steps for each resource require preprocessing and postprocessing functions in the form of a business service and business method. In the EAI Rest Adapter Service in Siebel Tools, you must map the URL that identifies the target resource with the business service and business method that is responsible for each preprocessing and postprocessing function.

The following figure shows the Business Service User Props pane and sample resource mapping user property keys and values:



W	Name	Changed	Project	Cache	Class	Display Name
	EAI Rest Adapter Service		Siebel Rest		CSSEAIRestAdapterService	EAI Rest Adapter Service

Business Service User Props

W	Name	Changed	Value	Inactive	Comments
	EnablePrePostProcessing	✔	true		
	resource_10	✔	service/*		
	resource_10_preprocess	✔	SWI External Integration Service:RefreshCache		
	resource_11	✔	workflow/*		
	resource_11_preprocess	✔	SWI External Integration Service:RefreshCache		
	resource_1	✔	data/Account		
	resource_1_postprocess	✔	SWI External Integration Service:RefreshCache		
	resource_1_preprocess	✔	SWI External Integration Service:RefreshCache		
	resource_2	✔	service/Siebel Account		
	resource_2_postprocess	✔	SWI External Integration Service:RefreshCache		
	resource_2_preprocess	✔	SWI External Integration Service:ResetSessionCache		
	resource_4	✔	workspace/Applet		
	resource_4_postprocess	✔	SWI External Integration Service:RefreshCache		
	resource_7	✔	workflow/EAISiebDemo1		
	resource_7_postprocess	✔	SWI External Integration Service:RefreshCache		
	resource_7_preprocess	✔	SWI External Integration Service:ResetSessionCache		
	resource_8	✔	data/*		
	resource_8_postprocess	✔	SWI External Integration Service:RefreshCache		
	resource_8_preprocess	✔	SWI External Integration Service:RefreshCache		
	resource_9	✔	workspace/*		
	resource_9_preprocess	✔	SWI External Integration Service:RefreshCache		

You can define wildcard specification for the URL mapping resource. For example, if you add an asterisk to a data URL resource (`data/*`), then preprocessing and postprocessing applies to all data APIs. In the event where you list a specific resource and a wildcard, then REST API considers the specific resource. You can only use wildcard to generalize an API type.

The following table contains sample resource URLs and their corresponding user properties and values to configure preprocessing and postprocessing steps.

> **Note:** The maximum number of characters for the user property key is 75, and the user property value is restricted to 250 characters.

Requested URL	User Property Key	User Property Value
data/Account/Account/.....	resource_1	data/Account
	resource_1_preprocess	Business_Service1:Method
	resource_1_postprocess	Business_Service2:Method
service/Siebel Account/.....	resource_2	service/Siebel Account

Requested URL	User Property Key	User Property Value
	resource_2_preprocess	Business_Service3:Method
	resource_2_postprocess	Business_Service4:Method
workspace/Applet/....	resource_3	workspace/Applet
	resource_3_preprocess	Business_Service1:Method
	resource 3 preprocess	Business_Service2:Method
workflow/EAISiebDemo1	resource_4	workflow/EAISiebDemo1
	resource_4_preprocess	Business_Service1:Method
	resource_4_preprocess	Business_Service2:Method
data/...	resource_5	data/*
	resource_5_preprocess	Business_Service4:Method
	resource_5_postprocess	Business_Service5:Method

Using Siebel Management Console to Configure a Siebel Application Interface Profile

After the Siebel Application Interface is installed, you must use the Siebel Management Console to create a Siebel Application Interface profile before using Siebel REST API services. For more information about creating a Siebel Application Interface Profile, see *Siebel Installation Guide* .

This topic includes the Siebel REST configuration parameters that you must configure when you create the Siebel Application Interface Profile. This topic includes the following:

- *Configuring REST Inbound Authentication Parameters*
- *Configuring REST Inbound Default Parameters*
- *Configuring REST Resource Parameters*

Configuring REST Inbound Authentication Parameters

You can configure resource parameters by giving the parameters alternative query names.

The following table contains the REST inbound authentication parameters that you configure when you create a Siebel Application Interface Profile. For more information about configuring a Siebel Application Interface profile, see *Siebel Installation Guide* .

Siebel Management Console Parameter	Section	Description
Allow anonymous inbound REST requests	Authentication, REST Inbound Authentication	Specify whether you want to allow inbound REST requests without authentication. Do one of the following: • Select this check box to allow anonymous inbound REST requests. This option is enabled by default. • Deselect this check box if you don't want to allow users to access the REST APIs anonymously. In this case, the anonymous REST requests will return a status code of 401 in the response.
Anonymous User Name	Authentication, REST Inbound Authentication	Specify an anonymous user to use for anonymous REST inbound requests.
Anonymous User Password	Authentication, REST Inbound Authentication	Specify a password for the anonymous user for REST inbound requests.
Authentication Type	Authentication, REST Inbound Authentication	Specify an authentication type that the Siebel Application Interface nodes accept for REST inbound authentication. You can select one of the following options: • Basic Authentication • Single Sign-On • OAuth
Trust Token	Authentication, REST Inbound Authentication	This option is available if you select the Single Sign-On or OAuth option. Specify the trust token to use for REST inbound authentication
Authentication URL	Authentication, REST Inbound Authentication	This option is available if you select the OAuth option. Specify the URL to use for REST inbound authentication.
User Specification	Authentication, REST Inbound Authentication	This option is available if you select the Single Sign-On option. Specify the user specification to use for REST inbound authentication.
Session Timeout (seconds)	Authentication, REST Inbound Authentication	Specify the session timeout, in seconds, to use for REST inbound authentication. This is the timeout in

Siebel Management Console Parameter	Section	Description
		which a connection remains open for further requests from same user.
Secure Channel	Authentication, REST Inbound Authentication.	This option applies only for the OAuth authentication type. Do one of the following: • Select this check box only when you have already imported the Authentication URLs CA certificate into the Application Interface truststore. • Deselect this check box when the Authentication URLs CA certificate is not available in the Application Interface truststore. In this case, the Application Interface trusts all certificates while calling the Authentication URL over HTTPS.

Configuring REST Inbound Default Parameters

The following table contains the REST inbound default parameters that you configure when you create a Siebel Application Interface Profile. For more information about configuring a Siebel Application Interface profile, see *Siebel Installation Guide* .

Siebel Management Console Parameter	Section	Description
Object Manager	REST Inbound Defaults	Select the Object Manager component to use for REST inbound communications, such as EAI Object Manager.
REST Response Base URL	REST Inbound Defaults	Specify the base URL used to generate URLs for resources in REST responses.
Maximum Possible Connections	REST Inbound Defaults	Specify the maximum number of possible connections available to serve REST requests. The default value is 20. This parameter determines the maximum number of REST requests an AI node can handle at a given time. The EAI Object Manager on the Siebel server serves REST requests, therefore you must ensure that this value doesn't exceed the MaxTasks parameter for the EAI Object Manager. Also, you must allocate some EAI Object Manager tasks for classic EAI or SOAP requests. Consider this example: the MaxTasks parameter on EAIObjmgr_enu is

Siebel Management Console Parameter	Section	Description
		1000. This means that the EAI Object Manager has the capacity to serve 1000 parallel requests. However, if the administrator wants to load the EAI Object Manager only to 80% capacity, then the resulting number of available tasks is 800. If the administrator decides to use 75% of the EAI Object Manager capacity for REST, and the remaining 25% for classic EAI soap requests, then the value of this parameter must be set to 600, that is, 75% of 800. Furthermore, if the administrator wants to distribute the REST load between multiple AI nodes, using any load balancer, then this parameter must also be divided among the available AI nodes. For example, if the setup has four AI nodes, then the Maximum Possible Connections parameter value must be 150, that is, 600 divided by 4.

Configuring REST Resource Parameters

The following table contains the REST resource parameters that you configure when you create a Siebel Application Interface Profile. For more information about configuring a Siebel Application Interface profile, see *Siebel Installation Guide* .

Siebel Management Console Parameter	Section	Description
Method Name	REST Inbound Defaults / REST Resource Parameter List, Query.	Specify the method name to use for queries.
Name	REST Inbound Defaults / REST Resource Parameter List, Query, Parameter List.	Specify the name for each query parameter.
Alias	REST Inbound Defaults / REST Resource Parameter List, Query, Parameter List	Specify the alias for each query parameter in the REST resource URIs. Query parameters include: Limit, Pagesize, and Offset.

4 Accessing Repository Resources

Using the Siebel REST API to Access Repository Resources

This chapter describes the Siebel REST API requests and responses for REST API calls to access Siebel CRM repository resources. It includes the following topics:

- *About Using the Siebel REST API*
- *Using Siebel REST API to Access Siebel Repository Resources JSON Examples*
- *Using Siebel REST API to Access Siebel Repository Data XML Examples*

About Using the Siebel REST API

Each topic in this chapter provides both JSON and XML examples that demonstrate how to use the Siebel REST API calls to interact with Siebel Server resources.

> **Note:** In Siebel CRM 21.2 Update and later, any changes to repository objects (such as a business service, business component or an integration object), do not require a restart of the object manager component used by REST. These changes are available immediately in the REST APIs once the changed workspace is delivered or migrated.

The example REST API calls use the following format:

- An example request, with the following information:
  - **URI.** The location of the Siebel REST API resource on the Siebel Server. For more information about Siebel REST API URL format, see *About Siebel CRM REST API URI Formats*.
  - **HTTP Method.** The HTTP method used to call the Siebel REST API to interact with the Siebel Server. For more information about supported HTTP Methods, *About Supported HTTP Methods*.
  - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is sent by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
  - **Request Body.** The payload for the Siebel REST API request.
- An example response, with the following information:
  - **HTTP Code.** The HTTP status code returned to indicate whether the request was successful or if there was an error. For more information about supported HTTP codes, *About Standard HTTP Status Codes and Error Messages*.
  - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is returned by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
  - **Response Body.** The Siebel REST API response.

Note: Because of the length of REST responses, some REST responses have been omitted.

Using Siebel REST API to Access Siebel Repository Resources JSON Examples

You can use the Siebel REST API to access Siebel CRM Repository Resources. Users can perform QUERY, INSERT, UPDATE, and DELETE operations on the Siebel CRM Repository Resources (such as applets or fields) using REST API requests over HTTP as described in this section.

This topic includes the following information:

Querying for a Siebel CRM Repository Resource

You can query for a Siebel CRM Repository Resource by sending a HTTP GET request to the Repository Resource's URI.

The following request returns the properties of the WriteRecord control configured in the SIS Account List Applet applet:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet / Control/WriteRecord**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource with a Search Specification

You can query for a particular Siebel CRM Repository Resource by stating the SearchSpec parameter in the request.

The following request queries for Business Components matching the criteria specified in the parameters:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Business Component?SearchSpec = [Name] LIKE 'B*'**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource to Return a Subset of Fields

You can query for a Siebel CRM Repository Resource to have a subset of fields in the response. When you query for a repository object, the response lists all the fields and links to child objects. If you specify a subset of fields in the request, then this filters the response to contain only the parameters listed in the query parameter.

The following request returns values for the Name, ProjectName, and Comments fields in SIS Account List Applet:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?fields=Name,ProjectName,Comments**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource To Return Only One Child Link

You can query for a repository object so that the response lists all the fields and links to child objects. You can specify a subset of child links in the request so that the response only contains those child links.

The following request queries for a Siebel CRM Repository Resource to list only the link for List child object under SIS Account List parent object:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ChildLinks=List**
- **HTTP Method:** GET

- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource That Returns a Subset of Child Links

You can submit a HTTP GET request to query for a Repository Resource to have a subset of child links in the response.

The following request queries for the SIS Account List Applet to retrieve child links for only Lists and Chart child objects:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ChildLinks=List,Chart**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic

Querying for a Siebel CRM Repository Resource to Return No Child Links

You can query for a Siebel CRM Repository Resource so that the response has no child links.

The following request queries for a SIS Account List Applet to return no child links:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ChildLinks=None**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic

Querying for a Siebel CRM Repository Resource Using the ViewMode Access Control Parameter

You can control the access to view and modify the Repository Resource by using the ViewMode parameter. This parameter controls operations based on Person, Position or Organization.

The following request queries for a SIS Account List Applet, with the ViewMode parameter set to Personal. This allows access to view the record only to the person who has created the record in the specified workspace.

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ViewMode="Personal"**
- **HTTP Method:** GET

- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying an Applet Using the recordcountneeded Parameter

There are times when you need to know the total number of records a query has retrieved, and not just a subset of the total records. You can query for the total record count by using the recordcountneeded parameter, and setting it to true. If you set this parameter to false, then no record count is returned. If neither true nor false is specified, then the default value for recordcountneeded is false.

When using recordcountneeded, the following considerations apply:

- This parameter applies only to GET URLs of /data and /workspace APIs.
- The total record count returns as a response header named: Total-Record-Count
- Total record count gives count records to which you have access.
- It gives a record count of the last business component specified in the request URL.
- The best practice, to optimize performance, is to only use this parameter when a record count is required.

In the following request, the recordcountneeded parameter is set to true, and the response has a header named Total-Record-Count which indicates that there are 20224 Applets in workspace that match the query.

- **URI: http://ServerName:port/siebel/v1.0/workspace/dev_sadmin_test/Applet?recordcountneeded=true**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Total-Record-Count:** 20224 (Response Header)

Querying for Controls in an Applet Using the recordcountneeded Parameter

There are times when you need to know the total number of records a query has retrieved, and not just a subset of the total records. You can query for the total record count by using the recordcountneeded parameter, and setting it to true. If you set this parameter to false, then no record count is returned. If neither true nor false is specified, then the default value for recordcountneeded is false.

When using recordcountneeded, the following considerations apply:

- This parameter applies only to GET URLs of /data and /workspace APIs.
- The total record count returns as a response header named: Total-Record-Count
- Total record count gives count records to which you have access.
- It gives a record count of the last business component specified in the request URL.
- The best practice, to optimize performance, is to only use this parameter when a record count is required.

In the following request, the recordcountneeded parameter is set to true, and the response has a header named Total-Record-Count which indicates that there are 28 controls associated with the SIS Account List applet.

- **URI: http://ServerName:port/siebel/v1.0/workspace/dev_sadmin_test/Applet/SIS Account List Applet/Control?recordcountneeded=true**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Total-Record-Count:** 28 (Response Header)

Querying for a Control's User Properties in an Applet Using the recordcountneeded Parameter

There are times when you need to know the total number of records a query has retrieved, and not just a subset of the total records. You can query for the total record count by using the recordcountneeded parameter, and setting it to true. If you set this parameter to false, then no record count is returned. If neither true nor false is specified, then the default value for recordcountneeded is false.

When using recordcountneeded, the following considerations apply:

- This parameter applies only to GET URLs of /data and /workspace APIs.
- The total record count returns as a response header named: Total-Record-Count
- Total record count gives count records to which you have access.
- It gives a record count of the last business component specified in the request URL.
- The best practice, to optimize performance, is to only use this parameter when a record count is required.

In the following request, the recordcountneeded parameter is set to true, and the response has a header named Total-Record-Count which indicates that there are 2 user properties associated with a control named QueryAssistant in the SIS Account List applet.

- **URI: http://ServerName:port/siebel/v1.0/workspace/dev_sadmin_test/Applet/SIS Account List Applet/Control/QueryAssistant/Control User Prop?recordcountneeded=true**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Total-Record-Count:** 2 (Response Header)

Querying for a Siebel CRM Repository Resource with a Sort Specification

You can query for a particular Siebel CRM Repository Resource, and sort its response in either an ascending or descending order by stating the sortspec parameter in the request. This provides you with a REST Get API response that is already sorted in a particular order, without the need to code any additional sorting scripts.

The sortspec parameter contains a comma separated list of field:sortorder pairs where spaces before and after the equals sign (=) and colon (:) are ignored. For example:

```
sortspec=First Name:desc,Last Name:asc
```

The keywords for ascending and descending order are `asc` and `desc` respectively. In the following example, the query first sorts the records in ascending order of Citizenship, and then in ascending order of First Name.

```
http://ServerName:port/siebel/v1.0/data/Account/1-1/Contact? sortspec=Citizenship:asc,First Name:asc
```

When using the `sortspec` parameter, the following considerations apply:

- If you do not specify the sortspec parameter, or if nothing is specified after the equals sign (=) in the sort specification, then the response is not sorted.
- When neither an ascending nor descending sort order is specified, the response is returned in ascending order by default.
- This parameter applies only to GET URLs of /data and /workspace APIs.
- You can only use the sortspec parameter on a collection query. If you use it in a single record query, then the sortspec parameter has no effect.
- You can sort at any level of object (child object, grandchild object, and so on), however, you can apply sortspec to the last object, or the leaf object, in the URI.
- To avoid an impact on API performance, only use the sortspec parameter when sorting is necessary.

In the following request, the sortspec parameter is set to `Name:asc`, so the returned response is sorted by the field Name in ascending order.

- **Request URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/Applet?Fields=Name&ChildLinks=None&sortspec=Name:asc&PageSize=3&StartRowNum=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response Body:**

```
"items": [
{
"Name": "A Account Objective Form Applet",
"Link": [
….. ]
},
{
"Name": "A Account Planning Header Applet",
"Link": [
….. },
{
"Name": "A Portfolio Account Detail Applet",
"Link": [
….. }
],
"Link": [
….
```

```
}
```

In the following request, the sortspec parameter is set to `Name:desc`, so the returned response is sorted by the field Name in descending order.

- **Request URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/Applet?Fields=Name&ChildLinks=None&sortspec=Name:desc&PageSize=3**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response Body:**

```
{
"items": [
{
"Name": "new FINCORP Call Report Activity Form Applet (More)",
"Link": [
…. ]
},
{
"Name": "f",
"Link": [
…. ]
},
{
"Name": "eWireless Page Tab Layout Applet",
"Link": [
…. ]
}
],
"Link": [
…. ]
}
```

Using the PageSize Parameter for Parent/Child/Grandchild Records

Inbound REST queries can set the limit for the number of records returned from a query for every level of the hierarchy. To do this, add the `PageSize` parameter with an integer as its argument at the desired level in the request.

Example 1

In this request the `PageSize` parameter is set at the parent level (since there is only one level in this request). This limits the response to 2 Account records.

- **URI: https://ServerName:port/siebel/v1.0/data/Account/Account?Fields=Id&Childlinks=None&StartRowNum=0&PageSize=2**
- **HTTP Method:** GET

- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body:** None

Example 2

In this request for Contacts for a particular Account record, the `PageSize` parameter is set at the child level to 2 for the Contacts.

- **URI: https://ServerName:port/siebel/v1.0/data/Account/Account/1-34Z/Contact?Fields=First Name, Last Name&Childlinks=None&StartRowNum=0&PageSize=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body:** None

Example 3

This request gets the Actions for Contacts that belong to an Account. Here, we are restricting the response to two Activity records for a particular Contact. This parameter is at the grandchild level with page size for grandchild.

- **URI: https://ServerName:porthttps://<ServerAddress>:<port>/siebel/v1.0/data/Account/Account/1-34Z/Contact/1LS-AF98/Action?StartRowNum=0&PageSize=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body:** None

Inserting a Siebel CRM Repository Resource

You can insert a Siebel CRM Repository Resource by sending a HTTP POST request to the Repository Resource's URI.

The following query request inserts a new applet:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet_1**
- **HTTP Method**: POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "Name": "SIS Account List Applet_1",
 "ProjectName":"Siebel Rest",
 "UpgradeBehavior":"Preserve",
 "Comments":"SIS Account List Applet: Added by Rest"
}
```

Upserting a Siebel CRM Repository Resource

You can insert or update a Siebel CRM Repository Resource by sending a HTTP PUT request to the resource's URI.

The following request inserts or, if it already exists, updates, a child record, that is, a WriteRecord control, to SIS Account List Applet_1:

- **URL: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet_1/Control/WriteRecord**
- **HTTP Method:** PUT
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "Name": "WriteRecord",
 "ProjectName":"Siebel Rest",
 "UpgradeBehavior":"Preserve",
 "Comments":"SIS Account List Applet: Added by Rest"
}
```

Deleting a Siebel CRM Repository Resource

You can delete a Siebel CRM Repository Resource by sending a HTTP DELETE request to the resource's URI.

The following request deletes the SIS Account List Applet_1 applet:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet_1**
- **HTTP Method:** DELETE
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Using the Describe Parameter to Return a Catalog of Repository Objects

You can query for a catalog of top level repository objects in a given workspace by sending a HTTP GET request with the describe parameter appended.

The following query requests a catalog of all top level repository objects, paths to retrieve definitions or catalog of each object, various responses to the request with HTTP codes, and so on in OpenAPI 2.0 format.

- **URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/describe?openapiversion=2**
- **HTTP Method:** GET
- **Content-Type:** application/json

- **Authorization**: Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code**: 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "swagger": "2.0",
 "info":
{
},
 "schemes":
 [
 "http",
 "https"
 ],
 "securityDefinitions":
 {
 "Basic Auth":
 {
 "type": "basic"
 }
 },
 "externalDocs":
 {
 "description": "OpenAPI",
 "url": "https://openapis.org"
 },
 "host": "host:port number",
 "basePath": "/siebel/v1.0",
 "tags": [
 {
 "name": "workspace/MAIN/Applet/describe",
 "description": "Cataloging of Applet",
 "externalDocs":
 {
 "description": "Find Out More",
 "url": ""
 }
 },
 {
 "name": "workspace/MAIN/Application/describe",
 "description": "Cataloging of Application",
 "externalDocs":
 {
 "description": "Find Out More",
 "url": ""
 }
 }
 ],
 "paths":
 {
 "/workspace/MAIN/Applet/describe":
 {
 "get":
 {
 "tags": [
 "workspace/MAIN/Applet/describe"
 ],
 "summary": "",
 "description": "",
 "operationId": "workspace/MAIN/Applet/describe",
```

```
"produces": [
"application/xml",
"application/json"
],
"responses": {
"200": {"description": "Successful Operation"},
"204": {"description": "No Resource Found"},
"404": {"description": "There is no data for the requested resource"},
"500": {"description": "Internal Server Error"}
},
"parameters": [],
"security": [
{
"Basic Auth": [],
"OAuth 2.0": []
}
]
}
}
}
```


The following query requests a catalog of all top level repository objects, paths to retrieve definitions or catalog of each object, various responses to the request with HTTP codes, and so on in OpenAPI 3.0 format.

- **URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization**: Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code**: 200
- **Content-Type:** application/json
- **Response body:**


```
{
"openapi": "3.0.1",
"info": {
"version": "1.0"
},
"externalDocs": {
"description": "OpenAPI",
"url": "https://openapis.org"
},
"servers": [
{
"url": "https://host:port number/siebel/v1.0"
}
],
"tags": [
{
"name": "workspace/MAIN/Applet/describe"
}],
"paths": {
"/workspace/MAIN/Applet/describe": {
"get": {
"tags": [
"workspace/MAIN/Applet/describe"
],
"operationId": "workspace/MAIN/Applet/describe",
"parameters": [
```

```
{
"name": "openapiversion",
"in": "query",
"schema": {
"type": "string",
"default": "3.0"
}
}
],
"responses": {
"200": {
"description": "Successful Operation",
"content": {}
},
"204": {
"description": "No Resource Found",
"content": {}
},
"401": {
"description": "Unauthorized",
"content": {}
},
"404": {
"description": "There is no data for the requested resource",
"content": {}
},
"500": {
"description": "Internal Server Error",
"content": {}
}
},
"security": [
{
"basicAuth": [],
"oAuth2.0": []
}
]
}
}
"components": {
"securitySchemes": {
"basicAuth": {
"type": "http",
"scheme": "basic"
},
"oAuth2.0": {
"type": "oauth2",
"flows": {
"implicit": {
"authorizationUrl": "http://openAPI.io/",
"scopes": {
"write:": "modify",
"read:": "read only"
}
}
}
}
}
},
"x-original-swagger-version": "2.0"
}
```

Querying for Repository Resource Metadata

You can query for the description of any repository object by appending the describe parameter to the object name. The response lists all the fields, paths for child objects, various responses with HTTP codes, and so on.

The following request returns applet repository object metadata using the describe parameter:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/Applet/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "swagger": "2.0",
 "info": {},
 "schemes": [],
 "securityDefinitions": {},
 "externalDocs": {},
 "host": "host:port number",
 "basePath": "/siebel/v1.0",
 "definitions": {},
 "tags": [],
 "paths": {
 "/workspace/MAIN/Applet/{key}/Applet Browser Script/describe": {
 "get": {
 "tags": [
 "workspace/MAIN/Applet/{key}/Applet Browser Script/describe"
 ],
 "summary": "",
 "description": "",
 "operationId": "workspace/MAIN/Applet/{key}/Applet Browser Script/describe",
 "produces": [
 "application/xml",
 "application/json"
 ],
 "responses": {
 "200": {"description": "Successful Operation"},
 "204": {"description": "No Resource Found"},
 "404": {"description": "There is no data for the requested resource"},
 "500": {"description": "Internal Server Error"}
 },
 "parameters": [
 {
 "name": "key",
 "in": "path",
 "description": "",
 "required": true,
 "default": "key",
 "type": "string"
 }
 ],
```

```
"security": [
{
"Basic Auth": [],
"OAuth 2.0": []
}
]
}
}
}
}
```


Querying for Child Repository Object MetaData

You can query for child repository objects metadata, such as required fields, properties of each field, catalog URLs for grandchildren, response format, and so on, by appending the describe parameter to the object name.

The following request queries for metadata belonging to the child of the applet repository object called Control.

- **URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/Applet/{key}/Control/describe**
- **HTTP Method**: GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"swagger": "2.0",
"info": {},
"schemes": [],
"securityDefinitions": {},
"externalDocs": {},
"host": "host:port number",
"basePath": "/siebel/v1.0",
"definitions": {
"workspace_MAIN_Applet_{key}_Control_": {
"type": "object",
"required": [
"Name",
"Type",
"Show Popup",
"HTML Row Sensitive",
"Parent Id",
"HTML Default Control"
],
"properties": {
"ActiveX Bind Property": {
"maxLength": 75,
"x-siebel-precision": "0",
"type": "string",
"x-siebel-datatype": "DTYPE_TEXT",
"x-siebel-scale": "0",
"title": "ActiveX Bind Property"
},
```

```
 "tags": [],
 "paths":{}
}
```

Querying Records Using the ExecutionMode Parameter

If you use a regular GET REST API where a query selects more than 9900 records, then it will fail and will cause an error. To execute this operation correctly, you must pass the ExecutionMode=ForwardOnly query parameter. For more information on the values you can use with the ExecutionMode query parameter, see *About URI Parameters*.

The following details are for a request query for a business component that returns 100 records starting from record number 9950, if this record number exists.

- **URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/Business Component?PageSize=100&StartRowNum=9950&ExecutionMode=ForwardOnly**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true, and the response body returns no fields with empty values:

- **URI:**

  http://ServerName:port/siebel/v1.0/workspace/MAIN/Applet?excludeEmptyFieldsInResponse=true
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Using Siebel REST API to Access Siebel Repository Data XML Examples

You can use the Siebel REST API to access Siebel CRM Repository Resources. Users can perform QUERY, INSERT, UPDATE, and DELETE operations on the Siebel CRM Repository Resources (such as account or contacts) using REST API requests over HTTP as described in this section. To view the response in XML format, the Content-Type header should be application/xml. The response will be in XML format wrapped in a `<response>` tag.

This topic includes the following information:

Querying for a Siebel CRM Repository Resource

You can query for a Siebel CRM Repository Resource by sending a HTTP GET request to the Repository Resource's URI.

The following request returns properties of WriteRecord control configured in the SIS Account List applet:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet/Control/WriteRecord**
- **HTTP Method**: GET
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource with a Search Specification

You can query for a specific Siebel CRM Repository Resource by specifying the search specification in the query.

The following request queries for Business Components matching the specified search criteria:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Business Component?SearchSpec = [Name] LIKE 'B*'**

- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Object to Return a Subset of Fields

You can query for a Siebel CRM Repository Object to have a subset of fields in the response. When you query for a repository object, the response lists all the fields and links to child objects. If you specify a subset of fields in the request, then this filters the response to contain only the fields listed in the query parameter.

The following request returns values for Name, ProjectName and Comments fields in SIS Account List Applet:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?fields=Name,Project Name,Comments**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Inserting a Siebel CRM Repository Resource

You can insert a Siebel CRM Repository Resource by sending a HTTP POST request to the Repository Resource's URI.

The following request inserts a new applet called SIS Account List Applet_1 in MyWorkspace:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet_1**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<request>
<Name>SIS Account List Applet_1</Name>
<Project_spcName>Siebel Rest</Project_spcName>
<Upgrade_spcBehavior>Preserve</Upgrade_spcBehavior>
<Comments>SIS Account List Applet: Added by Rest</Comments>
</request>
```

Upserting a Siebel CRM Repository Resource

You can insert or update a Siebel CRM Repository Resource by sending a HTTP PUT request to the Repository Resource's URI.

The following request inserts WriteRecord control (child record) to SIS Account List Applet_1:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet_1/Control/WriteRecord**
- **HTTP Method:** PUT
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
 <request>
 <Name>WriteRecord</Name>
 </request>
```

Deleting a Siebel CRM Repository Resource

You can delete a Siebel CRM Repository Resource by sending a HTTP DELETE request to the Repository Resource's URI.

The following request deletes the applet called SIS Account List Applet_1:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet_1**
- **HTTP Method:** DELETE
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource To Return Only One Child Link

You can query for a repository object so that the response lists all the fields and links to child objects. You can specify a subset of child links in the request so that response only contains those child links.

The following request queries for a Siebel CRM Repository Resource that lists only the link for List child object under the SIS Account List parent object:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ChildLinks=List**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource To Return Subset of Child Links

You can submit a HTTP GET request to query for a Repository Resource to have a subset of child links in the response.

The following request queries for the SIS Account List Applet to obtain child links of only List and Chart:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ChildLinks=List,Chart**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Repository Resource To Return No Child Links

You can query for a Siebel CRM Repository Resource and choose to have no child links in the response.

The following request queries for SIS Account List Applet which returns no child links:

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ChildLinks=None**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Querying for Siebel CRM Repository Resources by Using the ViewMode Access Control Parameter

You can control read and write access of the repository object by using the ViewMode parameter, which controls operations based on Person, Position or Organization.

The following request queries for the SIS Account List Applet, using the parameter ViewMode=Personal. This allows read access only to the person who created the record in the specified workspace.

- **URI: http://ServerName:port/siebel/v1.0/workspace/MyWorkspace/Applet/SIS Account List Applet?ViewMode=Personal**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

- **Request body:** None

Querying Records Using the ExecutionMode Parameter

If you use a regular GET REST API where a query selects more than 9900 records, then it will fail and will cause an error. To execute this operation correctly, you must pass the ExecutionMode=ForwardOnly query parameter. For more information on the values you can use with the ExecutionMode query parameter, see *About URI Parameters*.

The following details are for a request query for a business component that returns 100 records starting from record number 9950, if this record number exists.

- **URI: http://ServerName:port/siebel/v1.0/workspace/MAIN/Business Component?PageSize=100&StartRowNum=9950&ExecutionMode=ForwardOnly**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true, and the response body returns no fields with empty values:

- **URI:**

  http://ServerName:port/siebel/v1.0/workspace/MAIN/Applet?excludeEmptyFieldsInResponse=true
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

5 Accessing Business Objects

Using the Siebel REST API to Access Business Objects

This chapter describes the Siebel REST API requests and responses for REST API calls to access Siebel CRM business objects. It includes the following topics:

Configuring Integration Objects for REST API Data Access

An additional Business Object can be exposed through the Siebel REST API by creating an Integration Object for the Business Object, Base <Business Object Name>. Each Integration Component in the Integration Object should have an Integration Component Key, REST ROWID User Key:1, with one Integration Component Key Field Id defined. Without the Integration Component Key, the REST API upsert requests will not work on the Integration Component. For example:

If BO = "Service Request" then the IO name must be "Base Service Request"

For more information about integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

Supporting RESTful Access to Siebel Business Objects Dynamically

You can use the Dynamic IO feature to enable RESTful access of Business Objects without the need to create Base Integration Objects (Base IO).

Traditionally, to allow any Business Object access via REST, you had to create a new Base IO using Web Tools and migrate the same to production, which involved down-time and effort.

The Dynamic IO feature has:

- A simple process of exposing Business Object as REST API

- No overhead of creating Base Integration Objects
- No overhead of migrating Base Integration Objects from Development to Production
- No down-time, as there is no need to restart AI tomcat as there is no migration involved
- Improved access control via access configuration screen for both Base IO and Dynamic IO based
- Support for automated generation of REST user keys in tools while creating Base Integration Objects

Note: You can only specify a single Integration Object for a given Business Object when using the Administration Web Services -> Inbound REST API Access screen. Even though the Applet (REST Inbound Data Access Service List Applet) will allow you to specify more than one record, only one will be used.

This topic covers:

- *Enabling/Disabling Dynamic IO*
- *Access Control UI*
- *Dynamic IO Flow*
- *Dynamic IO Validation*
- *Dynamic IO Usage Considerations*
- *XSD Naming Conventions*
- *Purging of Old XSD*

Enabling/Disabling Dynamic IO

The Dynamic IO feature is controlled via the component parameter `EnableDynamicIOInREST` which is available in the EAI object manager. This feature is enabled by default, and the value of `EnableDynamicIOInREST` is set to 'true'.

- To disable the Dynamic IO feature, set the value of this parameter to 'false'.

  Example server manager command:

  ```
  change param EnableDynamicIOInREST=false for comp EAIObjmgr_enu
  ```

- To enable the Dynamic IO feature, set the value of this parameter to 'true'.

  Example server manager command:

  ```
  change param EnableDynamicIOInREST=true for comp EAIObjmgr_enu
  ```

The parameter change will take effect for all the new tasks/sessions.

Access Control UI

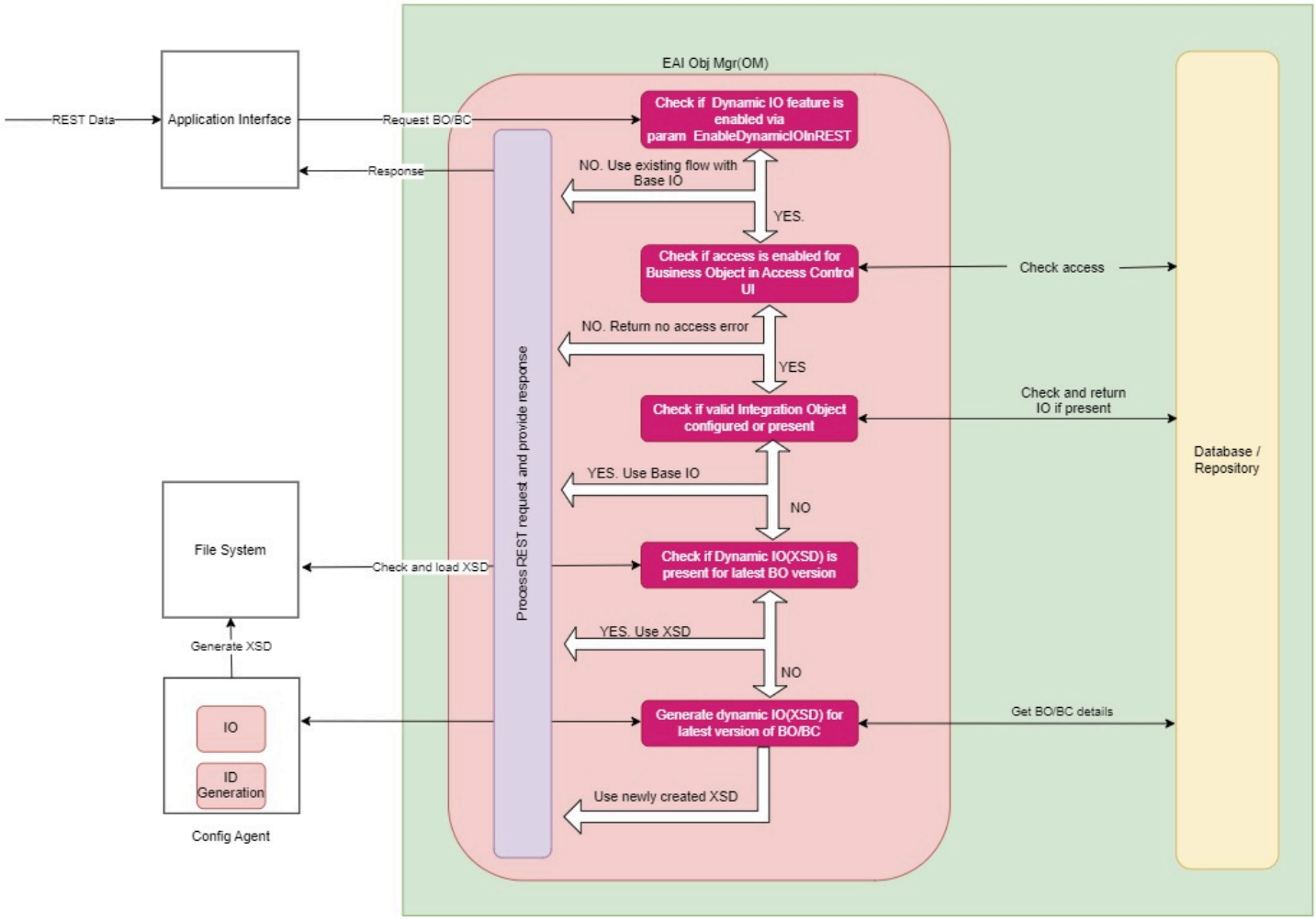
To see the Access Control UI, navigate to Sitemap > Administration Web Services > Inbound REST API Access.

- Once the Dynamic IO feature is enabled at component level, REST data API uses this model for access control.
- To enable access to a Business Object, the admin user should add the corresponding Business Object via the picklist and enable the flag for "Grant Access".
- All existing Base Integration Objects in the system would be migrated to the access control page automatically.

- If the Business Object enabled is not mapped to any Integration Object or if no Integration Object exists in system with Base <Business Object> pattern, then the system tries to expose the Business Object via a Dynamic Integration Object (Dynamic IO is created automatically as an XSD file).
- Open API specification for the data catalog API i.e., data/describe would list out all Business Objects enabled in the access control page.
- Open API specification for the data/Business Object/describe API would list out all possible root Business Components under Business Object if corresponding Business Object has REST access.
- With this access control model, the user can choose to use an Integration Object that does not have "Base" prefix in its name. For example, Business Object "Action" can use either Base Action or Action as Integration Object if available and configured in the access control page.

Dynamic IO Flow

This diagram shows the Dynamic IO flow for REST/data requests.



When there is a REST/data request:

1. The application checks if the Dynamic IO feature is enabled. If it is enabled, it proceeds to the next step, else it uses the existing Base IO flow.
2. It checks if access is provided to Business Object in the access control page. If the access is enabled, it proceeds to the next step, else it returns an error response.
3. If Base IO is present and active, then Base IO model is used to serve request.

4. If Base IO is not available or is inactive, then Dynamic IO/XSD model is used.

   It checks if Dynamic IO (XSD) is available for latest version of requested Business Object.

   - If present, Dynamic IO (XSD) is used, and the request is served.
   - If XSD is not available or available for an older version of BO, then then a new XSD is generated for the latest version of the Business Object being accessed.

     The newly generated Dynamic IO is stored in Siebel server's <Siebel Server>/xsd directory as a file with .xsd extension.

Dynamic IO Validation

When Dynamic IO (XSD) is created, the following validation rules are applied:

- Matching BO/BC as per request needs to be present in repository.
- Active Business Object Component under Business Object should have the corresponding active Business Component.
- Link associated with active Business Object Component should be present and valid in the repository.

Dynamic IO Usage Considerations

- Dynamic IO follows the Business Components and fields in repository.
- Any BC/field customization needed should be made and delivered to runtime repository prior to dynamic IO (XSD) generation.
- Configure the View Mode property in the root Business Component that is being exposed via the dynamic Integration Object REST feature. All child Business Components in the hierarchy inherit the root Business Component's configured View Mode. Empty View Mode information in the root Business Component can lead to data from that Business Component becoming accessible to any user (including anonymous users) without restrictions. For more information, refer to the "Viewing Business Component View Modes" topic of Siebel Security Guide.

  > **Note:** Changing the View Mode of a Business Component may affect the data the Business Component displays in Views in the application. Thorough testing should be done to ensure that your application's functionality has not been affected by this change.

- Base IO model should be followed if any customization is needed, i.e., if custom keys are needed or if there are field/BC changes that cannot be applied to Business Component.
- The first request for a Business Object will take relatively more time when Dynamic IO(XSD) is created in a given Siebel server. Subsequent requests will be faster as the Dynamic IO is re-used.

XSD Naming Conventions

- XSD is generated and stored in the xsd folder of respective Siebel server from which the request is served.
- It has the following naming convention:

  (Root BO)_(Root BC)_(workspacename)_(workspaceversion)_(BO version)_(BC version)_(major version)_(minor version).xsd
- For main workspace requests, workspace version will always be 0.

Purging of Old XSD

- When the requested BO/BC definition is modified and a new task is started, then these new repository changes are considered and the appropriate versioned XSD is generated.
- For MAIN workspace requests old versioned XSD will be deleted prior to generation of new XSD.

About Using the Siebel REST API

Each topic in this chapter provides both JSON and XML examples that demonstrate how to use the Siebel REST API calls to interact with Siebel Server resources.

Note: In Siebel CRM 21.2 Update and later, any changes to repository objects (such as a business service, business component or an integration object), do not require a restart of the object manager component used by REST. These changes are available immediately in the REST APIs once the changed workspace is delivered or migrated.

The example REST API calls use the following format:

- An example request, with the following information:
  - **URI.** The location of the Siebel REST API resource on the Siebel Server. For more information about Siebel REST API URL format, see *About Siebel CRM REST API URI Formats*.
  - **HTTP Method.** The HTTP method used to call the Siebel REST API to interact with the Siebel Server. For more information about supported HTTP Methods, *About Supported HTTP Methods*.
  - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is sent by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
  - **Request Body.** The payload for the Siebel REST API request.
- An example response, with the following information:
  - **HTTP Code.** The HTTP status code returned to indicate whether the request was successful or if there was an error. For more information about supported HTTP codes, *About Standard HTTP Status Codes and Error Messages*.
  - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is returned by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
  - **Response Body.** The Siebel REST API response.

Note: Because of the length of REST responses, some REST responses have been omitted.

Using Siebel REST API to Access Siebel Business Objects JSON Examples

You can use the Siebel REST API to access Siebel CRM Business Objects. Users can perform QUERY, INSERT, UPDATE, and DELETE operations on the Siebel Business Objects using REST API requests over HTTP as described in this section.

This topic includes the following information:

- *Using the Describe Parameter to Return OpenAPI Description of Multiple Object Operations*
- *Using the Uniformresponse Parameter*
- *Querying for All Contacts in an Account Without Using Uniformresponse to Return a Single Record*
- *Querying for All Contacts in an Account Using Uniformresponse to Return Multiple Records*
- *Querying for All Contacts in an Account Without Using Uniformresponse to Return Multiple Records*
- *Querying for a Single Account Record Without Using Uniformresponse*
- *Querying for a Single Account Record Using Uniformresponse*
- *Using the Developer Workspace Parameter to Preview Changes Without Compiling to the Repository*

Querying for a Siebel CRM Business Component

You can query for a Siebel CRM Business Component by sending a HTTP GET request to the resource's URI.

The following request queries for an Account Business Component record on the Siebel CRM Server, which has the ID: 1LS-9XKU:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/1LS-9XKU**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Business Component with a Search Specification

You can query for a Siebel CRM Business Component by sending a HTTP GET request with a search specification in the resource's URI.

The following request retrieves specific contacts associated with the account, ID 1-32HG, using the searchspec parameter appended to the request URI:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/1-32HG/Contact/?searchspec=([First Name] LIKE 'J*' AND [Last Name] LIKE 'A*')**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic

Querying for a Siebel CRM Business Component Record to Return a Subset of Fields

When you query for a Business Component, the response lists all the fields and links to child components. If you specify a subset of fields in the request, then this filters the response to contain only the fields listed in the query parameter.

The following query fetches values for only the Name, Location and Account Status fields of the Account record with the ID: ID 1-32HG:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/1-32HG?fields=Name, Location, Account Status**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic

Querying a Root Business Component Using the recordcountneeded Parameter

There are times when you need to know the total number of records a query has retrieved, and not just a subset of the total records. You can query for the total record count by using the recordcountneeded parameter, and setting it to true.

If you set the this parameter to false, then no record count is returned. If neither true nor false is specified, then the default value for recordcountneeded is false.

When using recordcountneeded, the following considerations apply:

- This parameter applies only to GET URLs of /data and /workspace APIs.
- The total record count returns as a response header named: Total-Record-Count
- Total record count gives count records to which you have access.
- It gives a record count of the last business component specified in the request URL.
- The best practice, to optimize performance, is to only use this parameter when a record count is required.

In the following request, the recordcountneeded parameter is set to true, and the response has a header named Total-Record-Count which indicates that there are 500 records in the Account business component that match the query.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?recordcountneeded=true**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Total-Record-Count:** 500 (Response Header)

Querying A Root Business Component Using the recordneeded and Viewmode Parameters

The default value in the ViewMode parameter is Sales Rep, and any ViewMode queries fetch only the records which are accessible to the Sales Rep role. However, if you need to know the total number of records in the organization of the user accessing the records, you can change the value of the ViewMode parameter to Organization, in conjunction with recordcountneeded parameter. If you set this parameter to false, then no record count is returned. If neither true nor false is specified, then the default value for recordcountneeded is false.

When using recordcountneeded, the following considerations apply:

- This parameter applies only to GET URLs of /data and /workspace APIs.
- The total record count returns as a response header named: Total-Record-Count
- Total record count gives count records to which you have access.
- It gives a record count of the last business component specified in the request URL.
- The best practice, to optimize performance, is to only use this parameter when a record count is required.

In the following request, the recordcountneeded parameter is set to true, and the ViewMode parameter is set to Organization. The response has a header named Total-Record-Count which indicates that there are 5707 records in the Account business component that match the query.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?recordcountneeded=true&ViewMode=Organization**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Total-Record-Count:** 5707 (Response Header)

Querying Child Business Components Using the recordcountneeded Parameter

There are times when you need to know the total number of records a query has retrieved, and not just a subset of the total records. You can query for the total record count by using the recordcountneeded parameter, and setting it to true. If you set this parameter to false, then no record count is returned. If neither true nor false is specified, then the default value for recordcountneeded is false.

When using recordcountneeded, the following considerations apply:

- This parameter applies only to GET URLs of /data and /workspace APIs.
- The total record count returns as a response header named: Total-Record-Count
- Total record count gives count records to which you have access.
- It gives a record count of the last business component specified in the request URL.
- The best practice, to optimize performance, is to only use this parameter when a record count is required.

In the following request, the recordcountneeded parameter is set to true, and the response has a header named Total-Record-Count which indicates that there are 30 contacts associated with this account.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-2WRXH/Contact?recordcountneeded=true**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Total-Record-Count:** 30 (Response Header)

Querying Grandchildren Business Components Using the recordcountneeded Parameter

There are times when you need to know the total number of records a query has retrieved, and not just a subset of the total records. You can query for the total record count by using the recordcountneeded parameter, and setting it to true. If you set this parameter to false, then no record count is returned. If neither true nor false is specified, then the default value for recordcountneeded is false.

When using recordcountneeded, the following considerations apply:

- This parameter applies only to GET URLs of /data and /workspace APIs.
- The total record count returns as a response header named: Total-Record-Count
- Total record count gives count records to which you have access.
- It gives a record count of the last business component specified in the request URL.
- The best practice, to optimize performance, is to only use this parameter when a record count is required.

In the following request, the recordcountneeded parameter is set to true, and the response has a header named Total-Record-Count which indicates that there are 5 Household records associated with a contact in a specific account.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-2WRXH/Contact/0CR-1MF5Z6/Household?recordcountneeded=true**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Total-Record-Count:** 5 (Response Header)

Querying for a Siebel CRM Business Component with a Sort Specification

You can query for a particular Siebel CRM Business Component, and sort its response in either an ascending or descending order by stating the sortspec parameter in the request. This provides you with a REST Get API response that is already sorted in a particular order, without the need to code any additional sorting scripts.

The sortspec parameter contains a comma separated list of field:sortorder pairs where spaces before and after the equals sign (=) and colon (:) are ignored. For example:

```
sortspec=First Name:desc,Last Name:asc
```

The keywords for ascending and descending order are `asc` and `desc` respectively. In the following example, the query first sorts the records in ascending order of Citizenship, and then in ascending order of First Name.

```
http://ServerName:port/siebel/v1.0/data/Account/1-1/Contact? sortspec=Citizenship:asc,First Name:asc
```

The sortspec parameter is only applied to the leaf node, or the last child object in the URI. In the previous example, this is the Contact business component.

When using the sortspecparameter, the following considerations apply:

- If you do not specify the sortspec parameter, or if nothing is specified after the equals sign (=) in the sort specification, then the response is not sorted.
- When neither an ascending nor descending sort order is specified, the response is sorted in ascending order by default.
- This parameter applies only to GET URLs of /data and /workspace APIs.
- You can only use the sortspec parameter on a collection query. If you use it in a single record query, then the sortspec parameter has no effect.
- You can sort at any level of object (child object, grandchild object, and so on), however, you can apply sortspec to the last object, or the leaf object, in the URI.
- To avoid an impact on API performance, only use the sortspec parameter when sorting is necessary.

In the following request, the sortspec parameter is set to `First Name:desc`, so the returned response is sorted by the field First Name in descending order.

- **Request URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Contact?sortspec = First Name : desc & Fields=First Name**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response Body:**

```
 "items": [
 {
 "Id": "24-287Y",
"First Name": "Roger",
 "Link": [
 …. ]
 },
 {
 Id": "1-57LT",
"First Name": "John",
 "Link": [
 …. },
 {
 "Id": "1-A9NW",
"First Name": "Henry",
 "Link": [
 …. }
 ],
 "Link": [
 ….
}
```

In the following request, the sortspec parameter is set to `First Name:asc`, so the returned response is sorted by the field First Name in ascending order.

- **Request URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Contact?sortspec = First Name : asc & Fields=First Name**

- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response Body:**

```
 {
 "items": [
 {
 "Id": "1-AJ3J",
"First Name": "Felix",
 "Link": [
 …. ]
 },
 {
 "Id": "1-A9NW",
"First Name": "Henry",
 "Link": [
 …. ]
 },
 {
 "Id": "1-57LT",
"First Name": "John",
 "Link": [
 …. ]
 }
 ],
 "Link": [
 …. ]
}
```

Inserting a Siebel CRM Business Component

You can insert a Business Component record by sending a HTTP POST request over REST.

The following request inserts a new Account record:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "Name": "AccountExample",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Location": "HQ-Distribution",
 "Description": "AccountData",
 "Primary Organization Id": "1-1DG",
}
```

Inserting Multiple Siebel CRM Business Components

You can insert multiple Business Component records by sending a single HTTP POST request to the resource's URI. The maximum number of records you can insert is 100.

The following request creates multiple new Account records on the Siebel CRM Server:

- **URL: https://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Name": "AccountExample1",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Location": "HQ-Distribution",
 "Description": "AccountData 1",
 "Primary Organization Id": "1-1DG"
 },
 {
 "Name": "AccountExample2",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Location": "HQ-Distribution",
 "Description": "AccountData 2",
 "Primary Organization Id": "1-1DG",
 }
]
```

- **Response body with response code: 200 OK:**

```
[
 {
 "Name": "AccountExample1",
 "Id": "88-1V80SZ",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "AccountData 1",
 "Link": {
 "rel": "self",
 "href": "https://HostName:port/siebel/v1.0/data/Account/Account/88-1V80SZ",
 "name": "Account"
 }
 },
 {
 "Name": "AccountExample2",
 "Id": "88-1V80T2",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "AccountData 2",
 "Link": {
 "rel": "self",
 "href": "https://HostName:port/siebel/v1.0/data/Account/Account/88-1V80T2",
 "name": "Account"
 }
```

```
 }
]
```

Inserting Multiple Siebel CRM Business Components When a Record Already Exists

When you insert multiple Business Component records, it is possible that a record with the same user key combination already exists. When this happens, the entire request is rejected, none of the records are inserted, and the error message in the following example is displayed.

The following details describe a request that is trying to create two new Account records. However, one of these Account record already has the same user key as an existing record on the Siebel CRM Server.

- **URL: https://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Name": "AccountExample1",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Location": "HQ-Distribution",
 "Description": "AccountData 1",
 "Primary Organization Id": "1-1DG"
 },
 {
 "Name": "AccountExample3",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Location": "HQ-Distribution",
 "Description": "AccountData 3",
 "Primary Organization Id": "1-1DG"
 },
 {
 "Name": "AccountExample4",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Location": "HQ-Distribution",
 "Description": "AccountData 4",
 "Primary Organization Id": "1-1DG"
 }
]
```

- **Response body with a status code: 409 (Conflict)**

```
{
 "ERROR": "Insert operation on integration component 'Account' failed because a matching record in
 business component 'Account' with search specification '[Location] = \"HQ-Distribution\" AND [Name] =
 \"AccountExample1\" AND [Primary Organization] = \"Millennium Institutional Finance Services IF ENU\"'
 was found.(SBL-EAI-04383)"
}
```

Inserting a Siebel CRM Child Business Component

You can insert a Siebel CRM child Business Object by sending a HTTP PUT request to the resource's URI.

The following request inserts a Contact child record into an existing Account Business Component:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/88-431RF/Contact**
- **HTTP Method:** PUT
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body**:

```
{
 "Employee Number":"1231",
 "Employer Name":"BXM",
 "Bill To First Name":"MAYANew",
 "Bill To Last Name": "ABRAHAMNew",
 "Primary Organization Id":"0-R9NH",
 "Account Integration Id":"",
 "Job Title":"",
 "Person UId":"0CR-1MF5Z611",
 "Primary Organization":"Default Organization",
 "Personal Contact":"N"
}
```

Inserting Multiple Siebel CRM Child Business Components

You can insert multiple Siebel CRM child Business Components by sending a HTTP PUT request to the resource's URI.

The following request inserts multiple Contacts under an existing Account record:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/1-1/Contact**
- **HTTP Method:** PUT
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "Contact":[
 {
 "Primary Organization Id":"1-1DG",
 "Primary Organization":"Default Organization",
 "First Name":"Ken",
 "Last Name": "Bass",
 "Id":"12345"
 },
 {
 "Primary Organization Id":"1-1DG",
 "Primary Organization":"Default Organization",
 "First Name":"test",
 "Last Name": "test1",
 "Id":"123456"
 }
 ]
```

```
}
```

Upserting a Siebel CRM Business Component

You can insert or update a Siebel CRM Business Component by sending a HTTP PUT request to the resource's URI.

The following request updates the Description field of an existing Account Business Component record:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/88-431RF**
- **HTTP Method:** PUT
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "Name": "AccountExample",
 "Primary Organization":
 "Millennium Institutional Finance Services IF ENU",
 "Location": "HQ-Distribution",
 "Description": "AccountDataUpdate",
 "Primary Organization Id": "1-1DG"
}
```

Upserting Multiple Siebel CRM Business Components

You can insert or update multiple Business Component records by sending a single HTTP PUT request to the resource's URI. The maximum number of records you can insert or update is 100. You use the ID field as the unique key to identify the record. This field, and the other field values to be updated, are mandatory. If the record with this ID field exists in the database, then the fields are updated. Otherwise, the record is inserted if all the user key fields are present in the request body.

The following request updates the existing Account record on the Siebel CRM Server:

- **URL: https://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** PUT
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Name": "AccountExample1",
 "Id": "88-1V80SZ",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 1"
 },
 {
 "Name": "AccountExample2",
 "Id": "88-1V80T2",
 "Location": "HQ-Distribution",
```

```
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 2"
 }
]
```

- **Response body with response code: 200 OK:**

```
[
 {
 "Name": "AccountExample1",
 "Id": "88-1V80SZ",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 1",
 "Link": {
 "rel": "self",
 "href": "https://HostName:port/siebel/v1.0/data/Account/Account/88-1V80SZ",
 "name": "Account"
 }
 },
 {
 "Name": "AccountExample2",
 "Id": "88-1V80T2",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 2",
 "Link": {
 "rel": "self",
 "href": "https://HostName:port/siebel/v1.0/data/Account/Account/88-1V80T2",
 "name": "Account"
 }
 }
]
```

Upserting a Siebel CRM Child Business Component

You can insert or update a Siebel CRM child Business Object by sending a HTTP PUT request to the resource's URI.

The following request inserts a new Opportunity record associated to an Account with the ID 88-431RF:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/88-431RF/Opportunity**
- **HTTP Method:** PUT
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "Id":"123456",
 "Name":"NewOpp",
 "Currency Code": "AUD",
 "Primary Organization":"Default Organization"
}
```

Deleting a Siebel CRM Business Component

You can delete a Siebel CRM Business Component record by sending a HTTP DELETE request to the resource's URI.

The following request deletes an Account record on the Siebel CRM Server:

- **URL: http://ServerName:port/siebel/v1.0/data/Account/Account/88-43CGR**
- **HTTP Method:** DELETE
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Deleting Multiple Siebel CRM Business Components

You can delete multiple Business Component records by sending a single HTTP DELETE request to the resource's URI. The maximum number of records you can delete is 100.

The following request deletes existing Accounts on the Siebel CRM Server using the ID field as the identifier:

- **URL: https://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** DELETE
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request Body: 200 OK:**

```
[
 {
 "Id": "88-1V80SZ"
 },
 {
 "Id": "88-1V80T2"
 }
]
```

- **Response body with status code: 200 OK:** None

Deleting Multiple Siebel CRM Business Components When A Record is Deleted

You can delete multiple Business Component records by sending a single HTTP DELETE request to the resource's URI. The maximum number of records you can delete is 100. When you delete multiple Business Component records, it is possible that a record ID is invalid, or that a record has already been deleted and no longer exists in the database. When this happens, the entire request is rejected, none of the records are deleted, and the error message in the following example is displayed.

The following request deletes multiple Account records where a record ID is invalid, or the record associated with the ID no longer exists in the database:

- **URL: https://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** DELETE
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Id": "88-1V80SZ"
 },
 {
 "Id": "88-1V80T2"
 },
 {
 "Id": "88-1V80L3"
 }
]
```

- **Response body with status code: 400 Bad Request:**

```
{
 "ERROR": "No rows retrieved corresponding to the business component 'Account'(SBL-EAI-04378)"
}
```

Delete Multiple Siebel CRM Business Components with Identifier in the URL and in the Request Body

You can delete multiple Business Component records by sending a single HTTP DELETE request to the resource's URI. The maximum number of records you can delete is 100. If the URL and request body provides the record ID, then Siebel REST ignores the ID in the URL and uses the ID provided in the request body. In this case, Siebel REST only deletes the records with IDs specified in the request body.

The following request deletes multiple Account records on the Siebel CRM Server by passing the IDs specified in the URL and in the request body.

> **Note:** In this example, the ID provided in the URL is ignored, and the records for which IDs are provided in the request body are deleted.

:

- **URL: https://ServerName:port/siebel/v1.0/data/Account/Account/88-1V80SZ**
- **HTTP Method:** DELETE
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
```

```
 "Id": "88-1V80T2"
 },
 {
 "Id": "88-1V80L3"
 }
]
```

- **Response body with status code: 200 OK:** None

Querying for a Siebel CRM Business Component To Return a Subset of Child Links

You query for a Business Component so that the response lists all the fields and links to child components. If you specify a subset of child links in the request, then this filters the response to contain only the child links listed in the query parameter.

The following request queries for an Account Business Component, and returns a link for a related UT Account Partner child Business Component:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?ChildLinks=UT Account Partner**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Business Component To Return Child Links for Multiple Child Business Components

You can query for a Siebel CRM Business Component that retrieves links for multiple child Business Component records related with a parent's Business Component.

The following request queries for an Account Business Component, and returns child links for the following related child Business Components: CUT Address for Account/Contact, FINS Security - Account External Holdings and FINS cBanking Facility:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?ChildLinks=CUT Address for Account/Contact,FINS Security - Account External Holdings,FINS cBanking Facility**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Child Business Component To Return Its Child Links

You can query for a Siebel CRM child Business Component that returns its child links, that is, links to grandchildren, by sending a HTTP GET request to the resource's URI.

The following request queries for a Siebel CRM child Business Component, and returns child links only for List Mgmt Lists:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-3CFLJ/Contact?ChildLinks=List Mgmt Lists**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Querying for a Siebel CRM Business Component by Specifying the ViewMode Parameter

You can control read and write access to a Siebel CRM Business Component by specifying theViewMode parameter in the HTTP GET request. It controls access to operations based on Person, Position, or Organization.

The following request queries for an Account, using the ViewMode="Sales Rep" access control. This gives access to the user to view only those Accounts belonging to a specific sales team:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/?ViewMode="Sales Rep"**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic

Note: If you do not specify the ViewMode parameter, then all accounts belonging to the SalesRep role are returned by default.

Querying for a Siebel CRM Business Component To Return No Child Links

You can query for a Siebel CRM Business Component to retrieve a record without any child links in the response by specifying the parameter ChildLinks=None in the HTTP GET request.

The following request specifies the ChildLinks query parameter so that the response contains no child links:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-3CFLJ/Contact?ChildLinks=None**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Using the Describe Parameter to Return a REST Resource's Metadata

You can retrieve metadata description in OpenAPI format (formerly known as Swagger) for any REST resource by simply appending the describe parameter to the URL and sending a GET request.

The following request uses the describe parameter to return REST resource metadata:

- **URI: <REST_RESOURCE_URL>/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Using the Describe Parameter to Return the Siebel Base Business Object Catalog

You use the OpenAPI describe parameter by appending it to a HTTP GET request to the resource's URI. This retrieves the catalog of Business Objects exposed over REST, their descriptions, the paths to retrieve the catalog of individual objects, request formats, possible responses with HTTP codes, and so on.

The following request uses the describe parameter to return the catalog of Business Objects exposed over REST in OpenAPI 2.0 format:

- **URI: http://ServerName:port/siebel/v1.0/data/describe?openapiversion=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "swagger":"2.0",
 "info":{
 "description":"Siebel REST API",
 "version":"1.0",
 "title":"Siebel REST API",
 "contact":{
 "email":"sample@sample.com"
 }
 },
 "schemes":[
 "http",
 "https"
 ],
 "securityDefinitions":{
 "Basic Auth":{
 "type":"basic"
 },
```

```
"OAuth 2.0":{
"type":"oauth2",
"authorizationUrl":"http://openAPI.io/",
"flow":"implicit",
"scopes":{
"write:":"modify",
"read:":"read only"
}
}
},
"externalDocs":{
"description":"OpenAPI",
"url":"https://openapis.org"
},
"host": "host:port number",
"basePath":"/siebel/v1.0",
"tags":[
{
"name":"data/Account/describe",
"description":"Catalogging of Account",
"externalDocs":{
"description":"Find Out More",
"url":""
}
},
{
"name":"data/Service Request/describe",
"description":"Catalogging of Service Request",
"externalDocs":{
"description":"Find Out More",
"url":""
}
},
"paths":{
"/data/Account/describe":{
"get":{
"tags":[
"data/Account/describe"
],
"summary":"",
"description":"",
"operationId":"data/Account/describe",
"produces":[
"application/xml",
"application/json"
],
"responses":{
"200":{"description":"Successful Operation"},
"204":{ "description":"No Resource Found"},
"404":{"description":"There is no data for the requested resource"},
"500":{"description":"Internal Server Error"}
},
"parameters":[

],
"security":[
{
"Basic Auth":[],
"OAuth 2.0":[]
}
]
}
},
"/data/Service Request/describe":{
"get":{
"tags":[
```

```
"data/Service Request/describe"
],
"summary":"",
"description":"",
"operationId":"data/Service Request/describe",
"produces":[
"application/xml",
"application/json"
],
"responses":{
"200":{"description":"Successful Operation"},
"204":{"description":"No Resource Found"},
"404":{"description":"There is no data for the requested resource"},
"500":{"description":"Internal Server Error"}
},
"parameters":[],
"security":[
{
"Basic Auth":[],
"OAuth 2.0":[]
}
]
},
"post":{
"tags":[
"data_Account_Account_"
],
"summary":"",
"description":"",
"operationId":"data_Account_Account_/post",
"produces":[
"application/xml",
"application/json"
],
"consumes":[
"application/xml",
"application/json"
],
"responses":{
"200":{"description":"Successful Operation"},
"204":{"description":"No Resource Found"},
"404":{ "description":"There is no data for the requested resource"},
"500":{ "description":"Internal Server Error"}
},
"parameters":[
{
"in":"body",
"name":"body",
"description":"",
"required":true,
"schema":{
"$ref":"#/definitions/data_Account_Account_"
}
}
],
"security":[
{
"Basic Auth":[],
"OAuth 2.0":[]
}
]
}
}
}
```

```
}
```

The following request uses the describe parameter to return the catalog of Business Objects exposed over REST in OpenAPI 3.0 format:

- **URI: http://ServerName:port/siebel/v1.0/data/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "openapi": "3.0.1",
 "info": {
 "title": "SblDt",
 "description": "OpenAPI specification for SiebelData - API : data",
 "contact": {
 "email": "sample@sample.com"
 },
 "version": "1.0"
 },
 "externalDocs": {
 "description": "OpenAPI",
 "url": "https://openapis.org"
 },
 "servers": [{
 "url": "https://hostname:port/siebel/v1.0"
 }],
 "tags": [{
 "name": "data/Account/describe"
 }],
 "paths": {
 "/data/Action/describe": {
 "get": {
 "tags": [
 "data/Action/describe"
 ],
 "operationId": "data/Action/describe",
 "parameters": [{
 "name": "openapiversion",
 "in": "query",
 "schema": {
 "type": "string",
 "default": "3.0"
 }
 }],
 "responses": {
 "200": {
 "description": "Successful Operation",
 "content": {}
 },
 "204": {
 "description": "No Resource Found",
 "content": {}
 },
 "401": {
```

```
"description": "Unauthorized",
"content": {}
},
"404": {
"description": "There is no data for the requested resource",
"content": {}
},
"500": {
"description": "Internal Server Error",
"content": {}
}
},
"security": [{
"basicAuth": [],
"oAuth2.0": []
}]
}
}
},
"components": {
"securitySchemes": {
"basicAuth": {
"type": "http",
"scheme": "basic"
},
"oAuth2.0": {
"type": "oauth2",
"flows": {
"implicit": {
"authorizationUrl": "http://openAPI.io/",
"scopes": {
"write:": "modify",
"read:": "read only"
}
}
}
}
}
},
"x-original-swagger-version": "3.0"
}
```


Using the Describe Parameter to Return Business Component Metadata

The OpenAPI resource, describe, is used to retrieve Business Component metadata. The result contains a catalog of fields configured in the Business Component, and it also lists the operations which can be performed on the Business Component, request and response structure, path to catalog of all child Business Components, and so on.

The following request retrieves the metadata description of an Account Business Component using the describe parameter in OpenAPI 2.0 format:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/describe?openapiversion=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:**
- **Response body:**


```
{
"definitions":{
"data_Account_Account_":{
"type":"object",
"required":
"properties":
}
},
"tags":[
{
"name":"data/Account/Account/",
"description":"Operations available on data/Account/Account/",
"externalDocs":{
"description":"Find Out More",
"url":""
}
},
"paths":{
"/data/Account/Account/":{
"get":{
"tags":[
"data_Account_Account_"
],
"summary":"",
"description":"",
"operationId":"data_Account_Account_/get",
"produces":[
"application/xml",
"application/json"
],
"responses":{
"200":{"description":"Successful Operation"},
"204":{"description":"No Resource Found"},
"404":{"description":"There is no data for the requested resource"},
"500":{"description":"Internal Server Error"}
},
"parameters":[
{
"name":"key",
"in":"path",
"description":"",
"required":true,
"type":"string"
}
],
"security":[
{
"Basic Auth":[],
"OAuth 2.0":[]
}
]
}
}
}
}
```

The following request retrieves the metadata description of an Account Business Component using the describe parameter in OpenAPI 3.0 format:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:**
- **Response body:**


```
{
 "openapi": "3.0.1",
 "info": {
 "version": "1.0"
 },
 "externalDocs": {
 "description": "OpenAPI",
 "url": "https://openapis.org"
 },
 "servers": [{
 "url": "https://host:port number/siebel/v1.0"
 }],
 "tags": [{
 "name": "data/Account/Account/"
 }],
 "paths": {
 "/data/Account/Account/": {
 "get": {
 "tags": [
 "data/Account/Account/"
 ],
 "operationId": "data_Account_Account__get",
 "parameters": [{
 "name": "childlinks",
 "in": "query",
 "schema": {
 "type": "string"
 }
 }],
 "responses": {
 "200": {
 "description": "Successful Operation",
 "content": {
 "application/json": {
 "schema": {
 "$ref": "#/components/schemas/data_Account_Account_get_all_response"
 }
 },
 "application/xml": {
 "schema": {
 "$ref": "#/components/schemas/data_Account_Account_get_all_response"
 }
 }
 }
 },
 "204": {
 "description": "No Resource Found",
```

```
"content": {}
},
"401": {
"description": "Unauthorized",
"content": {}
},
"404": {
"description": "There is no data for the requested resource",
"content": {}
},
"500": {
"description": "Internal Server Error",
"content": {}
}
},
"security": [{
"basicAuth": [],
"oAuth2.0": []
}]
}
}
},
"components": {
"schemas": {
"data_Account_Account_": {
"required": [],
"type": "object"
}
}
},
"securitySchemes": {
"basicAuth": {
"type": "http",
"scheme": "basic"
},
"oAuth2.0": {
"type": "oauth2",
"flows": {
"implicit": {
"authorizationUrl": "http://openAPI.io/",
"scopes": {
"write:": "modify",
"read:": "read only"
}
}
}
}
},
"x-original-swagger-version": "2.0"
}
```


Using the Describe Parameter to Return Child Business Component Metadata

You use the OpenAPI describe parameter to retrieve metadata description for a child Business Component by appending it to a HTTP GET request to the resource's URI. It lists all the fields, path to catalog of grandchildren, and request and response structure with HTTP codes.

The following request uses the describe parameter to return child Business Component metadata, that is, the Contact child Business Component Contact, under the Account parent Business Component in OpenAPI 2.0 format:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-6/Contact/describe?openapiversion=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"definitions":{
 "data_Account_Account_{key}_Contact_":{
 "type":"object",
 "required":[],
 "properties":[],
 "tags":[
 {
 "name":"data/Account/Account/{key}/Contact/",
 "description":"Operations available on data/Account/Account/{key}/Contact/",
 "externalDocs":{
 "description":"Find Out More",
 "url":""
 }
 },
 "paths":{
 "/data/Account/Account/{key}/Contact/":{
 "get":{
 "tags":[
 "data_Account_Account_{key}_Contact_"
 ],
 "summary":"",
 "description":"",
 "operationId":"data_Account_Account_{key}_Contact_/get",
 "produces":[
 "application/xml",
 "application/json"
 ],
 "responses":{
 "200":{"description":"Successful Operation"},
 "204":{"description":"No Resource Found"},
 "404":{ "description":"There is no data for the requested resource"},
 "500":{"description":"Internal Server Error"}
 },
 "parameters":[
 {
 "name":"key",
 "in":"path",
 "description":"",
 "required":true,
 "type":"string"
 }
 ],
 "security":[
 {
 "Basic Auth":[],
```

```
"OAuth 2.0":[]
}
]
},
"post":{
"tags":[
"data_Account_Account_{key}_Contact_"
],
"summary":"",
"description":"",
"operationId":"data_Account_Account_{key}_Contact_/post",
"produces":[
"application/xml",
"application/json"
],
"consumes":[
"application/xml",
"application/json"
],
"responses":{
"200":{"description":"Successful Operation"},
"204":{"description":"No Resource Found"},
"404":{"description":"There is no data for the requested resource"},
"500":{"description":"Internal Server Error"}
},
"parameters":[
{
"in":"body",
"name":"body",
"description":"",
"required":true,
"schema":{
"$ref":"#/definitions/data_Account_Account_{key}_Contact_"
}
}
],
"security":[
{
"Basic Auth":[],
"OAuth 2.0":[]
}
]
}
}
}

}
```


The following request uses the describe parameter to return child Business Component metadata, that is, the Contact child Business Component Contact, under the Account parent Business Component in OpenAPI 3.0 format:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-6/Contact/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json

- **Response body:**


```
{
 "paths": {
 "/data/Account/Account/{account_key}/Contact/": {
 "get": {
 "tags": [
 "data/Account/Account/{account_key}/Contact/"
 ],
 "operationId": "data_Account_Account__account_key__Contact__get",
 "parameters": [{
 "name": "account_key",
 "in": "path",
 "required": true,
 "schema": {
 "type": "string"
 }
 }],
 "responses": {
 "200": {
 "description": "Successful Operation",
 "content": {
 "application/json": {
 "schema": {
 "$ref": "#/components/schemas/data_Account_Account__account_key__Contact_get_all_response"
 }
 },
 "application/xml": {
 "schema": {
 "$ref": "#/components/schemas/data_Account_Account__account_key__Contact_get_all_response"
 }
 }
 }
 },
 "204": {
 "description": "No Resource Found",
 "content": {}
 },
 "401": {
 "description": "Unauthorized",
 "content": {}
 },
 "404": {
 "description": "There is no data for the requested resource",
 "content": {}
 },
 "500": {
 "description": "Internal Server Error",
 "content": {}
 }
 },
 "security": [{
 "basicAuth": [],
 "oAuth2.0": []
 }]
 },
 "put": {
 "tags": [
 "data/Account/Account/{account_key}/Contact/"
 ],
 "operationId": "data_Account_Account__account_key__Contact__put",
 "parameters": [{
 "name": "account_key",
 "in": "path",
 "required": true,
 "schema": {
```

```
"type": "string"
}
},
{
"name": "excludeEmptyFieldsInResponse",
"in": "query",
"schema": {
"type": "string",
"default": "false"
}
}
],
"requestBody": {
"content": {
"application/json": {
"schema": {
"$ref": "#/components/schemas/data_Account_Account__account_key__Contact_"
}
},
"application/xml": {
"schema": {
"$ref": "#/components/schemas/data_Account_Account__account_key__Contact_"
}
}
},
"required": true
},
"responses": {
"200": {
"description": "Successful Operation",
"content": {
"application/json": {
"schema": {
"$ref": "#/components/schemas/data_Account_Account__account_key__Contact_put_post_response"
}
},
"application/xml": {
"schema": {
"$ref": "#/components/schemas/data_Account_Account__account_key__Contact_put_post_response"
}
}
}
},
"204": {
"description": "No Resource Found",
"content": {}
},
"304": {
"description": "Not Modified",
"content": {}
},
"401": {
"description": "Unauthorized",
"content": {}
},
"404": {
"description": "There is no data for the requested resource",
"content": {}
},
"500": {
"description": "Internal Server Error",
"content": {}
}
},
"security": [{
"basicAuth": [],
```

```
"oAuth2.0": []
}],
"x-codegen-request-body-name": "body"
}
}
},
"components": {
"schemas": {
"data_Account_Account__account_key__Contact_": {
"required": [],
"type": "object",
"properties": ""
}
}
}
}
```


Using the Describe Parameter to Return OpenAPI Description of Multiple Object Operations

The OpenAPI resource, describe, is used to retrieve Business Component metadata. You can retrieve information on multiple object operations Siebel CRM business component API Doc by sending a single HTTP GET request to the resource's URI.

The following request retrieves the details of multiple object operations for Account APIs SWAGGER documentation.

Note: In this example, the query parameter describemultiobject=y indicates that the user is requesting an OpenAPI description for multiple object REST API.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/describe?describemultiobject=y**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Response body:** This returns a description in OpenAPI (SWAGGER) format.

Using the Uniformresponse Parameter

When the response of a GET REST request has a single record, it also has a single JSON object. This JSON object is identifiable because it starts with curly brackets. However, if the response has multiple records, then response records are enclosed in a json array, which starts with square brackets.

If you want your response enclosed in a json array, regardless of the number of records, then you must use the uniformresponse parameter and you must set the value to either yes or y.

Querying for All Contacts in an Account Without Using Uniformresponse to Return a Single Record

You can query for a Siebel CRM Business Component by sending a HTTPS GET request to the resource's URI, and by specifying that the response is either an array, or a list of items. You can do this by using the uniformresponse parameter which has the following values:

- yes or y. This returns the response as an array enclosed in square brackets.
- no or n. This returns the response as list of items enclosed in curly brackets.

The following request queries for all Contact records in a given Account record from the Siebel CRM Server, without using the `uniformresponse` parameter. There is only one Contact associated to Account, this returns a single record in the response:

- **URI:** `GET https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Contact?fields=First Name, Last Name&childlinks=Account,Position,Organization`
- **HTTP Method:** GET
- **Content Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content Type:**
- **Response body:**

```
{
 "Id": "0CR-1MF5Z6",
 "First Name": "JOHN",
 "Last Name": "SMITH",
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
 {
 "rel": "canonical",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
 {
 "rel": "parent",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Position",
 "name": "Position"
 },
```

```
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Organization",
 "name": "Organization"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Account",
 "name": "Account"
 }
 ]
}
```

Note: The response in this example is not in an array because the response contains only one record.

Querying for All Contacts in an Account Using Uniformresponse to Return Multiple Records

You can query for a Siebel CRM component to return its responses in an array by sending a HTTPS GET request to the resource's URI, and by setting the uniformresponse parameter to yes.

The following request queries for all Contact records in a given Account record from the Siebel CRM Server by using the uniformresponse parameter:

- **URI:** `GET https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Contact?fields=First Name, Last Name&childlinks=Account,Position,Organization&uniformresponse=yes`
- **HTTP Method:** GET
- **Content Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content Type:**
- **Response body:**

```
{
 "items": [
 {
 "Id": "0CR-1MF5Z6",
 "First Name": "JOHN",
 "Last Name": "SMITH",
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
 {
```

```
 "rel": "canonical",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
 {
 "rel": "parent",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Position",
 "name": "Position"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Organization",
 "name": "Organization"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Account",
 "name": "Account"
 }
 ]
 }
 ],
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Contact?
&uniformresponse=y&childlinks=Account,Position,Organization&fields=First Name, Last Name",
 "name": "Contact"
 }
 ]
}
```


> **Note:** The response in this example is in an array even though the array has single element. This is because the request contains the uniformresponse parameter.

Querying for All Contacts in an Account Without Using Uniformresponse to Return Multiple Records

You can query for a Siebel CRM component by sending a HTTPS GET request to the resource's URI.

The following request queries for all Contact records in a given Account record from the Siebel CRM Server, without using the uniformresponse parameter, and receives multiple records in the response:

- **URI:** `GET https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Contact?fields=First Name, Last Name&childlinks=Account,Position,Organization`
- **HTTP Method:** GET
- **Content Type:** application/json

- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content Type:**
- **Response body:**


```
{
 "items": [
 {
 "Id": "0CR-1MF5Z6",
 "First Name": "JOHN",
 "Last Name": "SMITH",
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
 {
 "rel": "canonical",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
 {
 "rel": "parent",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Position",
 "name": "Position"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Organization",
 "name": "Organization"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Account",
 "name": "Account"
 }
 ]
 }
 {
 "Id": "0V-18PLXQ",
 "First Name": "WILLIAM",
 "Last Name": "BROWN",
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
```

```
 {
 "rel": "canonical",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6",
 "name": "Contact"
 },
 {
 "rel": "parent",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Position",
 "name": "Position"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Organization",
 "name": "Organization"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/
Contact/0CR-1MF5Z6/Account",
 "name": "Account"
 }
 ]
 }
 ],
 "Link": {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Contact?
&childlinks=Account,Position,Organization&fields=First Name, Last Name",
 "name": "Contact"
 }
 }
```

Note: The response in this example is in an array because it has multiple records.

Querying for a Single Account Record Without Using Uniformresponse

You can query for a single Siebel CRM Business Component by sending a HTTPS GET request to the resource's URI.

The following request queries for a single Account record from the Siebel CRM Server without using the uniformresponse parameter:

- **URI:** `GET https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND?fields=Name,Location,Id&childlinks=Contact,Position`
- **HTTP Method:** GET
- **Content Type:** application/json
- **Authorization:** Basic

- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content Type:**
- **Response body:**

```
{
 "Name": "3Com",
 "Id": "88-26CND",
 "Location": "Headquarters",
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "canonical",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Contact",
 "name": "Contact"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Position",
 "name": "Position"
 }
 ]
 }
```

> **Note:** The response in this example is not in an array because it has only one record, and the request does not contain the uniformresponse parameter.

Querying for a Single Account Record Using Uniformresponse

You can query for a single Siebel CRM Business Component by sending a HTTPS GET request to the resource's URI.

The following request queries for a single Account record from the Siebel CRM Server by using the uniformresponse parameter:

- **URI:** `GET https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND?fields=Name,Location,Id&childlinks=Contact,Position,Organization&uniformresponse=y`
- **HTTP Method:** GET
- **Content Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content Type:**
- **Response body:**

```
{
 "items": [
 {
 "Name": "3Com",
 "Id": "88-26CND",
 "Location": "Headquarters",
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "canonical",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Contact",
 "name": "Contact"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account/88-26CND/Position",
 "name": "Position"
 },
 ]
 }
]
 "Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Account/Account?
uniformresponse=y&childlinks=Contact,Position,Organization&fields=Name,Location,Id",
 "name": "Account"
 }
 ]
}
```

> **Note:** The response in this example is in an array even though the array has single element. This is because the request contains the uniformresponse parameter.

Using the Developer Workspace Parameter to Preview Changes Without Compiling to the Repository

You can query active Siebel CRM objects by sending a HTTPS GET request to the resource's URI. You can also preview changes to a specific object in a developer branch before you deliver them to the Main workspace. You can achieve this with the help of the developer workspace parameter, workspace&version. For more information about the developer workspace parameter, see *About URI Parameters*.

You must add a responsibility to Siebel Runtime Metadata Publisher Service in the Business Service Access List View for Workspace Inspect. Workspace Inspect internally uses this business service, and the responsibility is needed for Workspace Inspect to work correctly. For more information about associating a Business Service with a responsibility, see *Configuring Business Service Methods for REST Access* and *Siebel Security Guide* .

The following request fetches Quote Item details for a quote with the ID <Quote ID>:

- **URI**: `https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote Id>/QuoteItem`
- **HTTP Method**: GET
- **Content Type**: application/json
- **Authorization:** : Basic
- **Request body**: None

Here are the response details for a successful request, displaying all the Quote Items for a specific Quote:

- **HTTP Code**: 200
- **Content Type**:
- **Response body**:


```
{
 "To Node": "",
 "Parent Product Id": "",
 .
 .
 .
 "Root Quote Item Id": "<Quote Item Id>",
 .
 .
 .
 "Service Account": "",
"Link": [
 {
 "rel": "self",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
 Id>",
 "name": "Quote Item"},
 {
 "rel": "canonical",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
 Id>",
 "name": "Quote Item"
 },
 {
 "rel": "parent",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>",
 "name": "Quote"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
 Id>/Contact",
 "name": "Contact"
 },
 {
 "rel": "child",
 "href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
 Id>/HTIM Audit Trail Item 2",
 "name": "HTIM Audit Trail Item 2"
 },
 {
 "rel": "child",
```

```
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote - Requested Schedule Lines.Line Number (Sequence)",
"name": "Quote - Requested Schedule Lines.Line Number (Sequence)"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/PUB Evidence Quote Item",
"name": "PUB Evidence Quote Item"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Approval History - Quote Item",
"name": "Approval History - Quote Item"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/LOY Quote Point Types",
"name": "LOY Quote Point Types"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Auto Quote Vehicles",
"name": "Auto Quote Vehicles"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote - SAP 4x Schedule Line Item",
"name": "Quote - SAP 4x Schedule Line Item"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote - Requested Schedule Lines",
"name": "Quote - Requested Schedule Lines"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote - Promised Schedule Lines.Line Number (Sequence)",
"name": "Quote - Promised Schedule Lines.Line Number (Sequence)"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote Item XA",
"name": "Quote Item XA"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Price List",
"name": "Price List"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote - Promised Schedule Lines",
"name": "Quote - Promised Schedule Lines"
},
{
"rel": "child",
```

```
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote Item Promotion Deal",
"name": "Quote Item Promotion Deal"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/FS Quote Line Item Asset",
"name": "FS Quote Line Item Asset"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Action",
"name": "Action"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote Item Promotion Deal",
"name": "Quote Item Promotion Deal"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Audit Trail Item 3",
"name": "Audit Trail Item 3"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Activity Plan",
"name": "Activity Plan"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Product Line",
"name": "Product Line"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote Item XA",
"name": "Quote Item XA"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Quote - SAP 4x Schedule Line Item",
"name": "Quote - SAP 4x Schedule Line Item"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/Validation Message - Quote Item",
"name": "Validation Message - Quote Item"
},
{
"rel": "child",
"href": "https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item/<Quote Item
Id>/FS Quote Line Item Asset",
"name": "FS Quote Line Item Asset"
}
]
```

```
}
```

You can modify the object definition and then preview the changes before delivering the workspace. In the following example, Quote Item is no longer a child component of the base Quote integration object (IO). If you pass the workspace name and version number as query parameters in the request, this change to the structure of the IO can be seen. The response fetches an error due to inactivation of the Quote Item object.

For more information about how to inactivate an object in Web Tools, see *Using Siebel Tools* .

URI: `https://<host_name>:<port_number>/siebel/v1.0/data/Quote/Quote/<Quote ID>/Quote Item?workspace=<workspace_name>&version=<ver_num>`

```
{
 "ERROR": "Incorrect Business Component name 'Quote Item' or the Business Component is not mapped to any
 Integration Component in the Integration Object 'Base Quote'.(SBL-EAI-50257)"
}
```

Querying Records Using the ExecutionMode Parameter

If you use a regular GET REST API where a query selects more than 9900 records, then it will fail and will cause an error. To execute this operation correctly, you must pass the ExecutionMode=ForwardOnly query parameter. For more information on the values you can use with the ExecutionMode query parameter, see *About URI Parameters*.

The following details are for a request query for an Account that returns 100 records starting from record number 9950, if this record number exists.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?PageSize=100&StartRowNum=9950&ExecutionMode=ForwardOnly**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true, and the response body returns no fields with empty values:

- **URI:**

  http://ServerName:port/siebel/v1.0/data/Account/Account?excludeEmptyFieldsInResponse=true
- **HTTP Method:** POST
- **Content-Type:** application/json

- **Authorization:** Basic
- **Request body:** None

Using Siebel REST API to Access Siebel CRM Business Objects XML Examples

You can use the Siebel REST API to access Siebel CRM Business Objects. Users can perform QUERY, INSERT, UPDATE, and DELETE operations on the Siebel Business Components using REST API requests over HTTP as described in this section. To view the response in XML format, the Content-Type header should be application/xml. The response will be in XML format wrapped in a `<response>` tag.

> **Note:** Siebel REST API is designed to perform operations on Business Objects and Business Components with fields, and not Business Objects or Business Components with XML tags. If you have integration components with XML tags for field names, then you must use SOAP requests instead of REST API requests.

This topic includes the following information:

Querying for a Siebel CRM Business Component

You can query for a Siebel CRM Business Object resource by sending a HTTP GET request to the Repository Resource's URI.

The following request queries for an Account Business Component record with the ID, 1LS-9XKU:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1LS-9XKU**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Querying for a Siebel CRM Business Component with a Search Specification

You can query for a Siebel CRM Business Component resource by sending a HTTP GET request with a search specification in the resource's URI.

The following request retrieves Contact records whose first name starts with J, last name starts with A, and that are associated with the Account record, ID 1-32HG, using the searchspec parameter:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-32HG/Contact/?searchspec=([First Name] LIKE 'J*' AND [Last Name] LIKE 'A*')**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Querying for a Siebel CRM Business Component Record to Return Specific Subset of Fields

You can query for a Siebel CRM Business Component to return a specific subset of fields by sending a HTTP GET request with the fields query parameter.

The following request fetches values only for the Name, Location and Account Status fields of an Account record, with ID 1-32HG:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-32HG?fields=Name, Location, Account Status**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Inserting a Siebel CRM Parent Business Component

You can insert a parent or root Business Component record by sending a HTTP POST request to the resource's URI.

The following request inserts an Account record:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
 <request>
 <Name>AccountData</Name>
```

```
<Primary_spcOrganization>Millennium Institutional Finance Services IF ENU</
Primary_spcOrganization>
<Location>HQ-Distribution</Location>
<Description>AccountData</Description>
<Primary_spcOrganization_spcId>1-1DG</Primary_spcOrganization_spcId>
</request>
```

Upserting a Siebel CRM Parent Business Component

You can insert or update a Siebel CRM parent Business Component resource by sending a HTTP PUT request to the Repository Resource's URI.

The following request updates the value of the Description field of an existing Account record:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** PUT
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
 <request>
 <Name>AccountData</Name>
 <Primary_spcOrganization>Millennium Institutional Finance Services IF ENU</
 Primary_spcOrganization>
 <Location>HQ-Distribution</Location>
 <Description>AccountDataUpdate</Description>
 <Primary_spcOrganization_spcId>1-1DG</Primary_spcOrganization_spcId>
 </request>
```

Deleting a Siebel CRM Parent Business Component

You can delete a Siebel CRM parent Business Component record by sending a HTTP DELETE request to the Repository Resource's URI.

The following request deletes the Account record on the Siebel CRM Server:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-43CGR**
- **HTTP Method:** DELETE
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<request>
 <Id>88-43CGR</Id>
</request>
```

Querying for a Siebel CRM Business Component To Return a Subset of Child Links

You can query for a Business Component so that the response lists all the fields and links to child components. If you specify a subset of child links in the request, then this filters the response to contain only the child links listed in the query parameter.

The following request queries for an Account Business Component to return links for related UT Account Partner child business components:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?ChildLinks=UT Account Partner**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Querying for a Siebel CRM Business Component To Return Child Links for Multiple Child Business Components

You can query for a Business Component to retrieve links to multiple child Business Component records related to a parent's Business Component.

The following request queries for the Account Business Component to return child links for the following related child Business Components: CUT Address for Account/Contact, FINS Security - Account External Holdings, FINS cBanking Facility:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?ChildLinks=CUT Address for Account/Contact,FINS Security - Account External Holdings,FINS cBanking Facility**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Querying for a Siebel CRM Child Business Component To Return Its Child Links

You can query for a Siebel CRM child Business Component to return its child links (that is, links to grand children) by sending a HTTP GET request to the resource's URI.

The following request queries for the Contact child Business Component to return child links only for the List Mgmt Lists child Business Component:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-3CFLJ/Contact?ChildLinks=List Mgmt Lists**
- **HTTP Method:** GET

- **Content-Type:** application/xml
- **Authorization:** Basic

Querying for a Siebel CRM Business Component To Return No Child Links

You can retrieve details of a record without any child links in the response by specifying the query parameter, ChildLinks=None in the HTTP GET request.

The following query specifies the ChildLinks=Noneparameter so that the response does not return any child links:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-3CFLJ/Contact?ChildLinks=None**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Querying for a Siebel CRM Business Component by Using the ViewMode Access Control Parameter

You can control read and write access to a Siebel CRM Business Component by specifying the ViewMode parameter in the HTTP GET request.

The following request queries for an Account Business Component with a ViewMode="Sales Rep" access control:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/?ViewMode=Sales Rep**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

Querying Records Using the ExecutionMode Parameter

If you use a regular GET REST API where a query selects more than 9900 records, then it will fail and will cause an error. To execute this operation correctly, you must pass the ExecutionMode=ForwardOnly query parameter. For more information on the values you can use with the ExecutionMode query parameter, see *About URI Parameters*.

The following details are for a request query for an Account that returns 100 records starting from record number 9950, if this record number exists.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account?PageSize=100&StartRowNum=9950&ExecutionMode=ForwardOnly**
- **HTTP Method:** GET
- **Content-Type:** application/xml
- **Authorization:** Basic

- **Request body:** None

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true, and the response body returns no fields with empty values:

- **URI:**
  http://ServerName:port/siebel/v1.0/data/Account/Account?excludeEmptyFieldsInResponse=true
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None

Using Siebel REST API to Perform CRUD Operations on Hierarchies of Siebel Object Records

You can use the Siebel REST API to create, update, and delete entire hierarchies (i.e. parent, child and grandchild) of Siebel object records in a single request. These APIs allow you to alter records in Siebel hierarchies with fewer requests, which results in better performance and efficiency.

For example, you can update an Account, its child Contacts, and its grandchild Households in a single request.

This topic includes the following information:

- *Fetching a Complete Hierarchy (Parent, Child, Grandchild)*
- *Inserting a Hierarchy*
- *Multi Insert for Hierarchy*
- *Updating a Hierarchy*
- *Multi Update for Hierarchy*
- *Querying for a Specific Child*

Fetching a Complete Hierarchy (Parent, Child, Grandchild)

Use this API to fetch a complete hierarchy, using the query parameter `?getChildren=ALL`

QueryParams: getallchildren

Value: ALL

URI: `https://host:port/siebel/v1.0/data/Account/Account?getChildren=ALL`

HTTP Method : GET

Content-Type: application/json

Authorization: Basic

Partial Response Body :


```
{
 "YTD Sales": "",
 "Contact Id": "0CR-1MF5Z6",
 "Total Potential Volume Currency Code": "USD",
 "Primary Contact Id": "0CR-1MF5Z6",

 ........
 "Contact": [{
 "Income Expenses Ownership Value": "",
 "Primary Personal Address Indicator": "",
 "DeDup Key Modification Date": "",
 "Since Date": "",
 "My Position Id": "0-5220",
 "AssetLiab Ownership Value": "",
 "ID Type": ""
 },
 {
 "Last Name": "ABRAHAM",
 "Provider Flag": "N",
 .......
 "Primary Assignment Manual Flag": "N",
 "Id": "0CR-1MF5Z6",
 "Login Name": "MAYABRAH",
 "Since Date": "",
 "My Position Id": "0-5220",
 "AssetLiab Ownership Value": "",
 "ID Type": ""
 }
 ],
 "Person": [{
 "Party Name": " SINGH, DARWIN",
 "Account Id": "88-26CND",
 "Primary Position Id": "0-5220",
 "Joined Account Id": "88-26CND",
 "Party UId": "88-34IB1D",
 "Account Primary Ship To Address Id": "",
 "Employee Row Id": "88-34IB1D",
 "Primary Personal Payment Profile Id": "",
 "Work Phone #": "",
 "New Responsibility": "Siebel Administrator",
 "Employee Flag": "Y",
 "Job Title": "",
 "Start Date": "01/30/2023",
 "User Type": "",
 "Party Type Code": "Person"

 },
 {
 "Party Name": "ABRAHAM, MAYA",
 "Account Primary Ship To Person Id": "",
 "Account Id": "88-26CND",
 "Primary Position Id": "0-5220",
 "Login Domain": "",
```

```
 "Thumbnail Source Path": "",
 "Status": "",
 "Image Source Path": "",
 "Id": "0CR-1MF5Z6",
 "Personal Contact": "N",
 "Account Price List Id": "",
 "Joined Account Id": "88-26CND"

 }
 ],
 "Link": [{
 "rel": "self",
 "href": "https://host:port/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 },
 {
 "rel": "canonical",
 "href": "https://host:port/siebel/v1.0/data/Account/Account/88-26CND",
 "name": "Account"
 }
 ]
}
```


Inserting a Hierarchy

Use this API to insert a complete parent, child hierarchy in a single call.

In this example, both Account and child contacts are inserted in a single API call.

URI: `https://host:port/siebel/v1.0/data/Account/Account`

HTTP Method : POST

Content-Type: application/json

Authorization: Basic

Request Body:


```
{
"Name": "John",
"Primary Organization": "ABC",
"Location": "Albany",
"Primary Organization Id": "1SIA-7SY3",
 "Contact":
 {
 "First Name": "Michael",
 "Last Name": "Stoke",
 "Primary Organization": "ABC",
 "Personal Contact": "5",
 "Person UId": "98765"
 }
}
```


Response Body:


```
{
 "items": {
 "Name": "John",
 "Id": "88-34R1YK",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
```

```
"Primary Organization": "ABC",
"Contact": {
"Id": "88-34R1YN",
"Link": {
"rel": "self",
"href": "https://host:port/siebel/v1.0/data/Account/Account/88-34R1YK/Contact/88-34R1YN",
"name": "Contact"
}
},
"Link": {
"rel": "self",
"href": "https://host:port/siebel/v1.0/data/Account/Account/88-34R1YK",
"name": "Account"
}
}
}
```

Multi Insert for Hierarchy

Use this API to insert an array of objects and its children.

URI: `https://host:port/siebel/v1.0/data/Account/Account`

HTTP Method : POST

Content-Type: application/json

Authorization: Basic

Request Body:

```
[
{
"Name": "Rob Stark",
"Primary Organization": "ABC",
"Location": "Albany",
"Primary Organization Id": "1SIA-7SY3",

"Contact": {
"First Name": "Mike",
"Last Name": "Logan",
"Primary Organization": "ABC",
"Personal Contact": "6",
"Person UId": "7867665",

"Household": {
"Household Number": "12",
"Primary Organization": "ABC"
}
}
},

{
"Name": "Edward Jacob",
"Location": "Albany",
"Primary Organization Id": "1SIA-7SY3",
"Primary Organization": "ABC",

"Contact": {
"First Name": "Ray",
"Last Name": "Paul",
"Primary Organization": "ABC",
```

```
 "Personal Contact": "7",
 "Person UId": "783675",

 "Household": {
 "Household Number": "15",
 "Primary Organization": "ABC"
 }
 }
 }
]
```

Response Body:

```
[{
 "Name": "Rob Stark",
 "Id": "88-34R224",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
 "Primary Organization": "ABC",
 "Contact": [{
 "Id": "88-34R227",
 "Contact_Households": [{
 "Id": "88-34R22B"
 }]
 }]
 },
 {
 "Name": "Edward Jacob",
 "Id": "88-34R22H",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
 "Primary Organization": "ABC",
 "Contact": [{
 "Id": "88-34R22K",
 "Contact_Households": [{
 "Id": "88-34R22O"
 }]
 }]
 }
]
```

Updating a Hierarchy

Use this API to update a hierarchy of objects and its children.

URI: `https://host:port/siebel/v1.0/data/Account/Account`

HTTP Method : PUT

Content-Type: application/json

Authorization: Basic

Request Body:

```
{
"Name": "Rob Stark",
"Id": "88-34R1ZI",
"Primary Organization": "ABC",
"Location": "Albany",
"Primary Organization Id": "1SIA-7SY3",
```

```
 "Contact": {
 "Id": "88-34R1ZL",
 "First Name": "Joseph",
 "Last Name": "Paul",

 "Household": {
 "Id": "88-34R1ZP",
 "Household Number":"24"
 }
 }
}
```

Response Body:

```
{
 "items": {
 "Name": "Rob Stark",
 "Id": "88-34R1ZI",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
 "Primary Organization": "ABC",
 "Contact": {
 "Id": "88-34R1ZL",
 "Contact_Households": {
 "Id": "88-34R1ZP"
 }
 }
 }
}
```

Multi Update for Hierarchy

Use this API to update an array of objects and its children.

URI: `https://host:port/siebel/v1.0/data/Account/Account`

HTTP Method : PUT

Content-Type: application/json

Authorization: Basic

Request Body:

```
[{
 "Name": "Rob Stark",
 "Id": "88-34R224",
 "Primary Organization": "ABC",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
 "Contact": {
 "Id": "88-34R227",
 "First Name": "Joseph",
 "Last Name": "Paul",
 "Primary Organization": "ABC",
 "Personal Contact": "6",
 "Person UId": "7867665",
 "Household": {
 "Id": "88-34R22B",
 "Household Number": "16",
 "Primary Organization": "ABC"
 }
```

```
 }
 },
 {
 "Name": "Roberts John",
 "Id": "88-34R22H",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
 "Primary Organization": "ABC",
 "Contact": {
 "Id": "88-34R22K",
 "First Name": "Joe",
 "Last Name": "Miler",
 "Primary Organization": "ABC",
 "Personal Contact": "7",
 "Person UId": "783675",
 "Household": {
 "Id": "88-34R22O",
 "Household Number": "10",
 "Primary Organization": "ABC"
 }
 }
 }
]
```

Response Body:

```
[{
 "Name": "Rob Stark",
 "Id": "88-34R224",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
 "Primary Organization": "ABC",
 "Contact": [{
 "Id": "88-34R227",
 "Contact_Households": [{
 "Id": "88-34R22B"
 }]
 }]
 },
 {
 "Name": "Roberts John",
 "Id": "88-34R22H",
 "Location": "Albany",
 "Primary Organization Id": "1SIA-7SY3",
 "Primary Organization": "ABC",
 "Contact": [{
 "Id": "88-34R22K",
 "Contact_Households": [{
 "Id": "88-34R22O"
 }]
 }]
 }
]
```

Querying for a Specific Child

You can query a hierarchy of parent and child objects by providing the pattern in the query parameter named `QueryHierarchy`.

This example queries Account with a given set of fields and its child Contact with a set of fields and a searchspec and sortspec.

URI: `https://host:port/siebel/v1.0/data/Account/Account?QueryHierarchy={"Account": {"fields": "SBA Review,CUT First Name","searchspec":"([CUT First Name] LIKE 'M*' )","Contact": {"fields": "First Name","sortspec": "First Name : asc"}}}`

HTTP Method : GET

Content-Type: application/json

Authorization: Basic

Response Body:


```
{
 "items": [{
 "Id": "88-26CND",
 "CUT First Name": "MAYA",
 "SBA Review": "",
 "Contact": [{
 "Id": "88-34IB1D",
 "First Name": "DARWIN"
 },
 {
 "Id": "0CR-1MF5Z6",
 "First Name": "MAYA"
 }
 ]
 },
 {
 "Id": "1-3EV",
 "CUT First Name": "Michael",
 "SBA Review": "",
 "Contact": {
 "Id": "1-33E-VFFIX",
 "First Name": "Michael"

 }
 },
 {
 "Id": "88-14QA1",
 "CUT First Name": "Mary",
 "SBA Review": "",
 "Contact": {
 "Id": "88-14QAF",
 "First Name": "Mary"
 }
 },
 {
 "Id": "88-34JIR4",
 "CUT First Name": "MULTI NAME3",
 "SBA Review": "",
 "Contact": [{
 "Id": "88-34JIR7",
 "First Name": "MULTI NAME3"
 },
 {
 "Id": "88-34JIRH",
 "First Name": "MULTI NAME5"
 }
 ]

 },
 {
 "Id": "88-34JHYB",
 "CUT First Name": "MULTI NAME3",
 "SBA Review": "",
```

```
"Contact": [{
"Id": "88-34JHYF",
"First Name": "MULTI NAME3"
},
{
"Id": "88-34JHYO",
"First Name": "MULTI NAME5"
}
]
},
{
"Id": "88-2XVT1",
"CUT First Name": "Mobile",
"SBA Review": "",
"Contact": {
"Id": "88-2XVW6",
"First Name": "Mobile"
}
},
{
"Id": "1-32G1",
"CUT First Name": "Mary",
"SBA Review": "",
"Contact": [{
"Id": "1-GBZC",
"First Name": "Christine"
},
{
"Id": "1-4EP",
"First Name": "Leis"
},
{
"Id": "1-47NA",
"First Name": "Mary"
}
]
}
]
}
```

6 Accessing Business Services

Using the Siebel REST API to Access Business Services

This chapter describes the Siebel REST API requests and responses for REST API calls to access Siebel CRM resources. It includes the following topics:

- *About Using the Siebel REST API*
- *Using Siebel REST API to Access Siebel Business Services JSON Examples*
- *Using Siebel REST API to Access Siebel CRM Business Objects XML Examples*

Configuring Business Service Methods for REST Access

You can use the Siebel REST API to access Siebel Business Services. Before you access the Siebel Business Services, you must associate the business with a responsibility to control access to the Business Service and its methods. For more information about associating a Business Service with a responsibility, see *Siebel Security Guide* .

Note: It is recommended that you do not add responsibilities to the EAI Siebel Adapter Business Service as it may lead to denial of general REST (repository or data) and SOAP Web Services access to other users.

The REST Service API has more restrictive access for Siebel Business Services than other Siebel Access Channels. Each Business Service and Business Service Method that is accessed through the REST API needs to be explicitly configured for access.

Note: For the Siebel REST API, if a Business Service Method is not configured for access by a particular responsibility, then it will not be accessible. This is different from SOAP or the User Interface channels where an unconfigured Business Service is accessible to all responsibilities.

To configure Business Service Methods for REST Access

1. Log in as an administrator.
2. Navigate to the Administration - Application screen, then the Business Service Access view, and then the Access By Responsibility view.
3. In the Business Service list, click New to select a Business Service.

   A new record appears in the Business Service list.
4. Click the Select button in the Name field.

   The Business Service dialog box appears.
5. Select the Business Service to which you want to control access, then click OK.

   The selected Business Service appears in the Business Service list view.
6. In the Access By Responsibility list view, click New.

The Add Responsibilities dialog box appears.

7. Select a responsibility to associate with the Business Service and then click OK.
   The selected responsibility appears in the Access By Responsibility list view.
8. In the Business Service Method list, click New to specify the Business Service methods to which the responsibility gains access.
   The Business Service Method dialog box appears. This dialog box displays the list of Business Service methods to which access is currently controlled.
9. If the Business Service method to which you want to allow the responsibility access appears in the Business Service Method dialog box, select it, then click OK.
10. Click the Select button in the Name field.
    The Business Service Method dialog box appears.
11. Select a Business Service method to associate with the responsibility and then click OK.
    The selected Business Service method appears in the Business Service Method list view.
12. From the Broadest Visibility list, select the view mode to associate with the responsibility.
13. Step off the record to save changes.
14. Click Clear Cache.
15. Restart the Siebel Application Interface Apache Tomcat server by executing the following batch scripts:
    For Microsoft Windows:

    ```
    \applicationcontainer_external\bin\shutdown.bat
    \applicationcontainer_external\bin\startup.bat
    ```

    For UNIX:

    ```
    \applicationcontainer_external\bin\shutdown.sh
    \applicationcontainer_external\bin\startup.sh
    ```

    For more information about the Siebel Application Interface Apache Tomcat server, see *Siebel Installation Guide* .

About Using the Siebel REST API

Each topic in this chapter provides both JSON and XML examples that demonstrate how to use the Siebel REST API calls to interact with Siebel Server resources.

> **Note:** In Siebel CRM 21.2 Update and later, any changes to repository objects (such as a business service, business component or an integration object), do not require a restart of the object manager component used by REST. These changes are available immediately in the REST APIs once the changed workspace is delivered or migrated.

The example REST API calls use the following format:

- An example request, with the following information:
  - **URI.** The location of the Siebel REST API resource on the Siebel Server. For more information about Siebel REST API URL format, see *About Siebel CRM REST API URI Formats*.
  - **HTTP Method.** The HTTP method used to call the Siebel REST API to interact with the Siebel Server. For more information about supported HTTP Methods, *About Supported HTTP Methods*.

- **Content-Type.** The part of the HTTP header that indicates the media type of the data that is sent by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
- **Request Body.** The payload for the Siebel REST API request.

- An example response, with the following information:
  - **HTTP Code.** The HTTP status code returned to indicate whether the request was successful or if there was an error. For more information about supported HTTP codes, *About Standard HTTP Status Codes and Error Messages*.
  - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is returned by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
  - **Response Body.** The Siebel REST API response.

Note: Because of the length of REST responses, some REST responses have been omitted.

Using Siebel REST API to Access Siebel Business Services JSON Examples

You can use the Siebel REST API to access Siebel Business Services. Users can call Siebel Business Services using the Siebel REST API HTTP POST request by specifying the Business Service Name, Method Name, and method parameters in the URI or request body.

Before you can access the Siebel Business Services, you must configure the access and responsibility values of the Siebel Business Service. For more information, see *Configuring Business Service Methods for REST Access*.

This topic includes the following information:

- *Accessing a Siebel Business Service with Arguments in the Request Body*
- *Accessing a Siebel Business Service with Arguments in the Request URI*
- *Accessing the QueryByExample Method of the Siebel Account Business Service*
- *Accessing a Siebel Business Service with the matchrequestformat Parameter*
- *Querying for an Account Using the Siebel Business Service QueryById Method*
- *Inserting an Account Using the Siebel Account Business Service*
- *Updating an Account Using a Siebel Account Business Service*
- *Upserting an Account Using a Siebel Account Business Service*
- *Using the Describe Parameter to Return Methods of a Business Service*
- *Using the Describe Parameter to Return a Catalog of Paths for All Available Business Services*

Accessing a Siebel Business Service with Arguments in the Request Body

You can access a Siebel CRM Business Service by sending a HTTP POST request to the resource's URI.

The following request calls the CreateAccount Method of the Account Business Service with the Business Service arguments included in the request body. This request creates an account called REST Test Business Service3 on the Siebel CRM Server:

- **URI: http://ServerName:port/siebel/v1.0/service/Account/CreateAccount**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "Account IO":
 {
 "IntObjectName":"Account IO",
 "IntObjectFormat":"Siebel Hierarchical",
 "ListOfAccount IO":
 {
 "Account":
 {
 "Name":"REST Test Business Service3"
 }
 }
 }
 }
}
```

Accessing a Siebel Business Service with Arguments in the Request URI

You can access a Siebel CRM Business Service by sending a HTTP POST request to the resource's URI.

The following request calls the QueryPage method of the AccountWS Business Service with the PageSize=10, StartRowNum=0, and ViewMode=All parameters included in the request URI. This request returns the first 10 Account records (because PageSize is 10 and StartRowNum is 0) matching the criteria specified in the body in the format of the integration object called Account_EMR. Only simple parameters, for example, strings or numbers, can be passed as query parameters, whereas complex arguments, such as integration objects, must be passed in the request body.

- **URI: http://ServerName:port/siebel/v1.0/service/AccountWS/QueryPage?PageSize=10&StartRowNum=0&ViewMode=All**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "SiebelMessage":
 {
 "MessageId":"",
 "MessageType":"Integration Object",
 "IntObjectName":"Account_EMR",
 "IntObjectFormat":"Siebel Hierarchical",
```

```
 "ListOfAccount_EMR":
 {
 "Account":
 {
 "Name":"A"
 }
 }
 }
 }
}
```

Accessing the QueryByExample Method of the Siebel Account Business Service

You can access a Siebel CRM Business Service by sending a HTTP POST request to the resource's URI.

The following request calls the QueryByExample method of the Siebel Account Business Service. This request queries for and returns an account called 3Com.

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryByExample**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "SiebelMessage":
 {
 "MessageId":"",
 "MessageType":"Integration Object",
 "IntObjectName":"Account Interface",
 "IntObjectFormat":"Siebel Hierarchical",
 "ListOfAccount Interface":
 {
 "Account":
 {
 "Name":"3Com"
 }
 }
 }
 }
}
```

Accessing a Siebel Business Service with the matchrequestformat Parameter

You can access a Siebel CRM business service by sending the matchrequestformat parameter in your request. The matchrequestformat parameter request and response has a SOAP-like structure that uses `Listof` wrappers, where

additional attributes such as pagination, and so on, can be present. The following changes occur in the input and output formats:

- The input request `body` wrapper is removed.
- The output response `Listof` wrapper is added to the integration object name, and to all child integration component names.

The following request calls the QueryByExample method of the Siebel Account Business Service. The matchrequestformat parameter value is set to Y, and the business service method arguments are in the request body. This request queries for accounts matching the account-related criteria entered by the user:

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryByExample?matchrequestformat=y**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "Account Interface",
 "MessageType": "Integration Object",
 "ListOfAccount Interface": {
 "Account": [
 {
 "Account Id": "1-1075",
 "Partner Flag": "Y",
 ......................
 "ListOfBusiness Address": {
 "Business Address": [
 {
 ........................
 "Province": "",
 "IsPrimaryMVG": "Y"
 }
 ]
 },
 "ListOfRelated Sales Rep": {
 ........................
 }
 },
 {
 "Account Id": "1-106W",
 "Account Status": "Active",
 ........................
 "ListOfBusiness Address": {
 "Business Address": [
 {
 ..............
 }
 ]
 }
 }
 ]
 }
 }
}
```

- **Response body:**

The following is the response of the QueryByExample method of the Siebel Account Business Service:

```
{
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "Account Interface",
 "MessageType": "Integration Object",
 "ListOfAccount Interface": {
 "Account": [
 {
 "Account Id": "1-1075",
 "Partner Flag": "Y",
 ........................
 "ListOfBusiness Address": {
 "Business Address": [
 {
 ........................
 "Province": "",
 "IsPrimaryMVG": "Y"
 }
 ]
 },
 "ListOfRelated Sales Rep": {
 ........................
 }
 },
 {
 "Account Id": "1-106W",
 "Account Status": "Active",
 ........................
 "ListOfBusiness Address": {
 "Business Address": [
 {
 ..............
 }
 ]
 }
 }
 ]
 }
 }
}
```

Note: Swagger specification for the same business service can be verified using the matchrequestformat parameter mentioned in Swagger specification URL. For example **http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryByExample/describe?matchrequestformat=y**

Querying for an Account Using the Siebel Business Service QueryById Method

You can query for an Account record using the Siebel CRM Business Service QueryById method by sending a HTTP POST request to the resource's URI.

The following request calls the QueryById method of the Siebel Account Business Service, and the PrimaryRowId parameter is accepted as the query parameter. The following query returns an account with the PrimaryRowId value as 88-459YQ.

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById?PrimaryRowId=88-459YQ**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
}
```

Inserting an Account Using the Siebel Account Business Service

You can create a record in Siebel CRM database by using the Insert method of Siebel CRM Business Service by sending a HTTP POST request to the resource's URI.

The following requests calls the Insert method of Siebel Account Business service to insert a new Account record with its details given in the request.

- **URI: https://ServerName:port/siebel/v1.0/service/Siebel Account/Insert**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "SiebelMessage":
 {
 "MessageId":"",
 "MessageType":"Integration Object",
 "IntObjectName":"Account Interface",
 "IntObjectFormat":"Siebel Hierarchical",
 "ListOfAccount Interface":
 {
 "Account":
 {
 "Account Id":"2345",
 "Name":"REST Test Business Service"
 }
 }
 }
}
}
```

Updating an Account Using a Siebel Account Business Service

You can modify the record by using the Update method of Siebel CRM Business Service by sending a HTTP POST request to the resource's URI.

The following request calls the Update method of Siebel Account Business Service to update the Main Phone Number field value.

- **URI: https://ServerName:port/siebel/siebel/v1.0/service/Siebel Account/Update**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "SiebelMessage":
 {
 "MessageId":"",
 "MessageType":"Integration Object",
 "IntObjectName":"Account Interface",
 "IntObjectFormat":"Siebel Hierarchical",
 "ListOfAccount Interface":
 {
 "Account":
 {
 "Account Id":"1-6",
 "Name":"REST Test Business Service32",
 "Main Phone Number": "2018742315"
 }
 }
 }
 }
}
```

Upserting an Account Using a Siebel Account Business Service

You can upsert a record by using the InsertorUpdate method of Siebel CRM Business Service by sending a HTTP POST request to the resource's URI.

The following details are for a request to call the InsertOrUpdate method of Siebel Account Business service to insert an account with its details given in the request.

- **URI: https://ServerName:port/siebel/v1.0/service/Siebel Account/InsertOrUpdate**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "SiebelMessage":
 {
 "MessageId":"",
 "MessageType":"Integration Object",
 "IntObjectName":"Account Interface",
 "IntObjectFormat":"Siebel Hierarchical",
 "ListOfAccount Interface":
 {
 "Account":
```

```
{
"Account Id":"34567",
"Name":"Rest",
"Main Phone Number": "20187423154"
}
}
}
}
}
```

Using the Describe Parameter to Return Methods of a Business Service

You can use the OpenAPI describe parameter to access catalog of available methods of a Business Service by appending it to a HTTP GET request to the resource's URI.

The following request uses the describe parameter to return methods of Siebel Account Business Service URI in OpenAPI 2.0 format:

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/describe?openapiversion=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
"swagger": "2.0",
"info": {},
"schemes": [],
"securityDefinitions": {},
"externalDocs": {},
"host": "host:port number",
"basePath": "/siebel/v1.0",

"definitions": {},

"tags": [],

"paths": {

"/service/Siebel Account/Delete/describe": {

"get": {

"tags": [

"service/Siebel Account/Delete/describe"

],

"summary": "",
```

```
"description": "",
"operationId": "service/Siebel Account/Delete/describe",
"produces": [
],
"responses": {
},
"parameters": [],
"security": [
]
}
},
"/service/Siebel Account/Insert/describe": {
},
"/service/Siebel Account/InsertOrUpdate/describe": {
},
"/service/Siebel Account/QueryByExample/describe": {
},
"/service/Siebel Account/QueryById/describe": {
},
"/service/Siebel Account/Synchronize/describe": {
},
"/service/Siebel Account/Update/describe": {
}
}
}
```


The following request uses the describe parameter to return methods of Siebel Account Business Service URI in OpenAPI 3.0 format:

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200

- **Content-Type:** application/json
- **Response body:**


```
{
 "openapi": "3.0.1",
 "info": {
 "version": "1.0"
 },
 "externalDocs": {
 "description": "OpenAPI",
 "url": "https://openapis.org"
 },
 "servers": [],
 "tags": [
 ],
 "paths": {
 "/service/Siebel Account/Delete/describe": {
 "get": {
 "tags": [
 "service/Siebel Account/Delete/describe"
 ],
 "operationId": "service/Siebel_Account/Delete/describe",
 "parameters": [{
 "name": "openapiversion",
 "in": "query",
 "schema": {
 "type": "string",
 "default": "3.0"
 }
 }],
 "responses": {
 "200": {
 "description": "Successful Operation",
 "content": {}
 },
 "204": {
 "description": "No Resource Found",
 "content": {}
 },
 "401": {
 "description": "Unauthorized",
 "content": {}
 },
 "403": {
 "description": "Access Forbidden",
 "content": {}
 },
 "404": {
 "description": "There is no data for the requested resource",
 "content": {}
 },
 "500": {
 "description": "Internal Server Error",
 "content": {}
 }
 },
 "security": [{
 "basicAuth": [],
 "oAuth2.0": []
 }]
 }
 }
 },
 "/service/Siebel Account/Insert/describe": {
 },
```

```
 "/service/Siebel Account/InsertOrUpdate/describe": {
 },
 "/service/Siebel Account/QueryByExample/describe": {
 },
 "/service/Siebel Account/QueryById/describe": {
 },
 "/service/Siebel Account/Synchronize/describe": {
 },
 "/service/Siebel Account/Update/describe": {
 }
}
```

Using the Describe Parameter to Return a Catalog of Paths for All Available Business Services

You can use the OpenAPI describe parameter to retrieve path to all available Business Services exposed over REST by appending it to a HTTP GET request.

The following request uses the describe parameter to return a catalog of URLs for all available Business Services from the Siebel CRM Server in OpenAPI 2.0 format:

- **URI: http://ServerName:port/siebel/v1.0/service/describe?openapiversion=2**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
 "swagger": "2.0", //Existing Swagger specification version
 "info": { //Metadata about API
 },
 "schemes": [
 "http",
 "https"
 ],
 "securityDefinitions": { //information about the authorization schemes
 "Basic Auth": {
 "type": "basic"
 }
 },
 "externalDocs": {
 "description": "OpenAPI",
 "url": "https://openapis.org"
 },
 "host": "host:port number",
 "basePath": "/siebel/v1.0",
 "tags": [
 {
 "name": "service/ABO Bulk Request Explode Service/describe",
 "description": "Catalogging of ABO Bulk Request Explode Service",
 "externalDocs": {
```

```
"description": "Find Out More",
"url": ""
}
},

{

"name": "service/nextSet/describe",

"description": "Catalogging of nextSet",

"externalDocs": {

"description": "Find Out More",

"url": ""

}

}

],
"paths": {
"/service/ABO Bulk Request Explode Service/describe": { //links
"get": {
"tags": [
"service/ABO Bulk Request Explode Service/describe"
],
"summary": "",
"description": "",
"operationId": "service/ABO Bulk Request Explode Service/describe",
"produces": [
"application/xml",
"application/json"
],
"responses": {
"200": {"description": "Successful Operation"},
"204": {"description": "No Resource Found"},
"404": {"description": "There is no data for the requested resource"},
"500": {"description": "Internal Server Error"}
},
"parameters": [],
"security": [
{
"Basic Auth": [],
"OAuth 2.0": []
}
]
}
},

"/service/describe?PageSize=10&StartRowNum=10": { //links
"get": {
"tags": [
"service/nextSet/describe"
],
"summary": "",
"description": "",
"operationId": "service/nextSet/describe",
"produces": [
"application/xml",
"application/json"
],
"responses": {
"200": {"description": "Successful Operation"},
"204": {"description": "No Resource Found"},
```

```
"404": {"description": "There is no data for the requested resource"},
"500": {"description": "Internal Server Error"}
},
"parameters": [],
"security": [
{
"Basic Auth": [],
"OAuth 2.0": []
}
]
}
}
}
}
```


The following request uses the describe parameter to return a catalog of URLs for all available Business Services from the Siebel CRM Server in OpenAPI 3.0 format:

- **URI: http://ServerName:port/siebel/v1.0/service/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"openapi": "3.0.1",
"info": {
"version": "1.0"
},
"externalDocs": {
"description": "OpenAPI",
"url": "https://openapis.org"
},
"servers": [{
"url": "http://host:port/siebel/v1.0"
},
{
"url": "http://host:port/siebel/v1.0"
}
],
"tags": [{
"name": "service/04-EQ1Y1/describe"
},
{
"name": "service/ABO Bulk Request Explode Service/describe"
}
],
"paths": {
"/service/04-EQ1Y1/describe": {
"get": {
"tags": [
"service/04-EQ1Y1/describe"
],
"operationId": "service/04-EQ1Y1/describe",
"parameters": [{
"name": "openapiversion",
```

```
"in": "query",
"schema": {
"type": "string",
"default": "3.0"
}
}],
"responses": {
"200": {
"description": "Successful Operation",
"content": {}
},
"204": {
"description": "No Resource Found",
"content": {}
},
"401": {
"description": "Unauthorized",
"content": {}
},
"403": {
"description": "Access Forbidden",
"content": {}
},
"404": {
"description": "There is no data for the requested resource",
"content": {}
},
"500": {
"description": "Internal Server Error",
"content": {}
}
},
"security": [{
"basicAuth": [],
"oAuth2.0": []
}]
}
},
"/service/ABO Bulk Request Explode Service/describe": {
"get": {
"tags": [
"service/ABO Bulk Request Explode Service/describe"
],
"operationId": "service/ABO_Bulk_Request_Explode_Service/describe",
"parameters": [{
"name": "openapiversion",
"in": "query",
"schema": {
"type": "string",
"default": "3.0"
}
}],
"responses": {
"200": {
"description": "Successful Operation",
"content": {}
},
"204": {
"description": "No Resource Found",
"content": {}
},
"401": {
"description": "Unauthorized",
"content": {}
},
"403": {
```

```
"description": "Access Forbidden",
"content": {}
},
"404": {
"description": "There is no data for the requested resource",
"content": {}
},
"500": {
"description": "Internal Server Error",
"content": {}
}
},
"security": [{
"basicAuth": [],
"oAuth2.0": []
}]
}
},

"/service/describe?PageSize=10&StartRowNum=10": {
"get": {
"tags": [
"service/nextSet/describe"
],
"operationId": "service/nextSet/describe",
"parameters": [{
"name": "openapiversion",
"in": "query",
"schema": {
"type": "string",
"default": "3.0"
}
}],
"responses": {
"200": {
"description": "Successful Operation",
"content": {}
},
"204": {
"description": "No Resource Found",
"content": {}
},
"401": {
"description": "Unauthorized",
"content": {}
},
"403": {
"description": "Access Forbidden",
"content": {}
},
"404": {
"description": "There is no data for the requested resource",
"content": {}
},
"500": {
"description": "Internal Server Error",
"content": {}
}
},
"security": [{
"basicAuth": [],
"oAuth2.0": []
}]
}
}
},
```

```
"components": {
"securitySchemes": {
"basicAuth": {
"type": "http",
"scheme": "basic"
},
"oAuth2.0": {
"type": "oauth2",
"flows": {
"implicit": {
"authorizationUrl": "http://openAPI.io/",
"scopes": {
"write:": "modify",
"read:": "read only"
}
}
}
}
}
},
"x-original-swagger-version": "2.0"
}
```


Using the Describe Parameter to Return the OpenAPI Specification of a Business Service Method

You can use the describe parameter to retrieve the OpenAPI specification of a business service method, as shown in the following request:

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById/describe**
- **HTTP Method:** GET
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:** The response body contains the OpenAPI specification of the business service method, all its input and output arguments, and its integration objects, if applicable.

Using the Developer Workspace Parameter to Preview Changes to a Business Service Without Compiling to the Repository

You can preview or inspect changes to a business object in a developer workspace branch before you deliver these changes to the Main workspace. You can achieve this with the help of the developer workspace parameter: workspace&version. For more information about the developer workspace parameter, see *About URI Parameters*.

Note: This feature is useful when developing and testing repository changes.

You must add a responsibility to Siebel Runtime Metadata Publisher Service in the Business Service Access List View for Workspace Inspect. Workspace Inspect uses this business service internally, and this responsibility is needed for Workspace Inspect to work correctly. For more information about associating a Business Service with a responsibility, see *Configuring Business Service Methods for REST Access* and *Siebel Security Guide* .

The following example demonstrates previewing workspace changes without actually delivering the workspace. It uses the Siebel CRM Business Service Siebel Account, and QueryById method.

First, the standard Siebel Business Service method argument SiebelMessage points to the integration object (and output argument): Account Interface. When this method is called through REST API, the received output is: Account Interface IO.

Next, you create a workspace in Web Tools. In this example we use: dev_sadmin_mytestws. You must update the Siebel Business Service method argument SiebelMessage to a different integration object, in this example: Account IO. Version the workspace, creating the workspace: version 1.

Without delivering the workspace, use REST API to call the Siebel CRM Business Service QueryById method, using the following argument:

- workspace=dev_sadmin_mytestws&version=1

The received response is in the form Account IO, as described in the following response example.

- **URI**: `https://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById?uniformresponse=y&workspace=dev_sadmin_mytestws&version=1`
- **HTTP Method**: POST
- **Content Type**: application/json
- **Authorization:** : Basic
- **Request body**:

```
{
"body":{
"PrimaryRowId":"1-63Q9"
}
}
```

Here are the partial response details for a successful request, you'll see that the output integration object is Account IO:

- **HTTP Code**: 200
- **Content Type**: application/json
- **Response body**:

```
{
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "Account IO",
 "MessageType": "Integration Object",
 "Account": [
 {
 "Location": "HQ-Distribution",
 "PO Approved Flag": "",
 "Home Page": "3COM.com",
 "Domestic Ultimate DUNS": "",
 "Assignment Area Code": "",
 "Name": "3Com",
 "PO Auto Approval Limit": "",
```

```
"Parent Account Location": "",
"Type": "Customer",
"Parent Account Name": "",
"Integration Id": "",
"Price List Id": "",
"DUNS Number": "",
"Partner Flag": "N",
"Parent Account Integration Id": "",
"Credit Status Date": "",
"Main Fax Number": "6505551212",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Parent Account Id": "",
"PO Auto Approval Date": "",
"Price List": "",
"VAT registration number": "",
"Account Status": "Active",
"Global Ultimate DUNS": "",
"Skip Credit Check": "N",
"Alias": "",
"Main Phone Number": "6505551212",
"Assignment Country Code": "",
"Expertise": "",
"Credit Auto Approval Limit": "0",
"Currency Code": "USD",
"Parent HQ DUNS": "",
"PO Auto Approval Currency Code": "",
"Price List Integration Id": ""
}
]
}
}
```


Next, in Web Tools, update the Siebel Business Service method argument SiebelMessage with a different integration object. This time with the name: Account_EMR. Version the workspace, creating the workspace: version 2.

Without delivering the workspace, use REST API to call the Siebel CRM Business Service QueryById method using the following parameter:

- workspace=dev_sadmin_mytestws&version=2

The received response is in the form Account_EMR, as described in the following example:

- **URI**: `https://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById?uniformresponse=y&workspace=dev_sadmin_mytestws&version=2`
- **HTTP Method**: POST
- **Content Type**: application/json
- **Authorization:** : Basic

Here are the partial response details for a successful request, where the output integration object is Account_EMR:

- **HTTP Code**: 200
- **Content Type**: application/json
- **Response body**:


```
{
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "Account_EMR",
 "MessageType": "Integration Object",
 "Account": [
 {
 "Account Status": "Active",
```

```
"Parent Account Integration Id": "",
"Parent Account Id": "",
"Alias": "",
"Account Id": "1-63Q9",
"Name": "3Com",
"Location": "HQ-Distribution",
"Currency Code": "USD",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Integration Id": "",
"Parent Account Name": "",
"Parent Account Location": "",
"Related Contact": [
{
"Contact Id": "88-23SUX",
"First Name": "Selfservice",
"Contact Integration Id": "",
"Middle Name": "",
"Last Name": "Merchant",
"Primary Organization": "Comms-Media Default Organization (COM ENU)",
"Person UId": "88-23SUX"

]
}
]
}
}
```


Note: If you omit the `param &version=<version_number>` query, then REST works only on the latest version in that workspace.

Using the Describe Workspace Parameter to Preview the Modified OpenAPI Specification of a Business Service Method Without Compiling to the Repository

You may need to preview or inspect the modified OpenAPI specification of changes to a business service in a developer workspace branch before you deliver these changes to the Main workspace. You can achieve this by using the /describe parameter, and the developer workspace parameter, workspace&version. For more information, see *About URI Parameters*.

Note: This feature is useful when developing and testing repository changes.

You must add a responsibility to Siebel Runtime Metadata Publisher Service in the Business Service Access List View for Workspace Inspect. Workspace Inspect uses this business service internally, and this responsibility is needed for Workspace Inspect to work correctly. For more information about associating a Business Service with a responsibility, see *Configuring Business Service Methods for REST Access* and *Siebel Security Guide* .

The following example uses the Siebel CRM Business Service Siebel Account, and QueryById method.

First, the standard Siebel Business Service method argument SiebelMessage points to the integration object (and output argument): Account Interface. When we generate the OpenAPI specification of this method using the /describe parameter, the specification output is: Account Interface IO.

Next, you create a workspace in Web Tools. In this example we use: dev_sadmin_mytestws. You must update the Siebel Business Service method argument SiebelMessage to a different integration object, in this case: Account IO. Version the workspace, creating the workspace: version 1.

Without delivering the workspace, use REST API to preview the modified OpenAPI specification of this method, using the following parameters:

- /describe
- workspace=dev_sadmin_mytestws&version=1

- **URI**: `https://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById/describe?workspace=dev_sadmin_mytestws&version=1`
- **HTTP Method**: GET
- **Content Type**: application/json
- **Authorization:** : Basic

Here are the response details for a successful request:

- **HTTP Code**: 200
- **Content Type**: application/json
- **Response body**: The response body contains the OpenAPI specification of the method, all its input and output arguments, and integration objects, if applicable. The specification for the business service method output is: Account IO. This is because we updated version 1 of the workspace.

Next, in Web Tools, update the Siebel Business Service method argument SiebelMessage with a different integration object. This time with the name: Account_EMR. Version the workspace, creating the workspace: version 2.

Without delivering the workspace, you can preview the modified OpenAPI specification of this method using REST API and the following parameters:

- /describe
- workspace=dev_sadmin_mytestws&version=2

The received response is in the form Account_EMR, as described in the following example:

- **URI**: `https://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById/describe?uniformresponse=y&workspace=dev_sadmin_mytestws&version=1`
- **HTTP Method**: GET
- **Content Type**: application/json
- **Authorization** :Basic

Here are the partial response details for a successful request, where the output integration object is Account_EMR:

- **HTTP Code**: 200
- **Content Type**: application/json
- **Response body**: The response body contains the OpenAPI specification of the method, all its input and output arguments, and integration objects, if applicable. The specification for the business service method output is Account_EMR. This is because we updated version 2 of the workspace.

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true, and the response body returns no fields with empty values:

- **URI:**

  http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById?excludeEmptyFieldsInResponse=true
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":{
 "PrimaryRowId":"88-26CND",
 "Number of Employees": ""
 }
}
```

- **Response body:** The response body returns Account with only non-empty fields.

Using Siebel REST API to Access Siebel Business Services XML Examples

You can use the Siebel REST API to access Siebel Business Services. Users can call Siebel Business Services using the Siebel REST API HTTP POST request by specifying the Business Service Name, Method Name, and method parameters in the URI. To view the response in XML format, the Content-Type header should be application/xml. The response will be in XML format wrapped in a `<response>` tag.

This topic includes the following information:

- *Using a Siebel CRM Business Service to Insert an Account*
- *Using a Siebel CRM Business Service to Update an Account*
- *Using a Siebel CRM Business Service to Delete an Account*
- *Using a Siebel CRM Business Service to Query for an Account*
- *Accessing the QueryByExample Method of the Siebel Account Business Service*

Using a Siebel CRM Business Service to Insert an Account

You can use a Siebel CRM Business Service resource to insert an Account by sending a HTTP POST request to the Repository Resource's URI.

The following request calls the Insert method of Siebel Account Business service, which inserts a new account with details given in the request.

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/Insert**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<body>
 <SiebelMessage>
 <MessageId></MessageId>
 <MessageType>Integration Object</MessageType>
 <IntObjectName>Account Interface</IntObjectName>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <ListOfAccount_spcInterface>
 <Account>
 <Name>Test Account</Name>
 <Account_spcId>1234</Account_spcId>
 </Account>
 </ListOfAccount_spcInterface>
 </SiebelMessage>
</body>
```

If the Siebel CRM Business Service fails with any error, then Siebel REST API returns the HTTP status code 500, details of the error, and the Siebel error code which is available in the response body. Consider the following two examples:

- If Siebel REST API tries to insert a record without the required fields, then it returns the HTTP error code 500, with the following message in the response body:

```
"ERROR": "No user key can be used for the Integration Component instance 'Account'.(SBL-EAI-04397)"
```

- If Siebel REST tries to insert a record that already exists, then it returns the HTTP error code 500 with the Siebel error message bubbled up, as shown in the following example:

```
"ERROR": "Insert operation on integration component 'Account' failed because
a matching record in Business Component 'Account' with search specification
'[Location] = \"HQ-Distribution\" AND [Name] = \"AccountData1234\" AND
[Primary Organization] = \"Millennium Institutional Finance Services IF
ENU\"' was found.(SBL-EAI-04383)"
```

Using a Siebel CRM Business Service to Delete an Account

You can use a Siebel CRM Business Service to delete an Account record by sending a HTTP POST request to the resource's URI.

The following request deletes an Account record on the Siebel CRM Server using the delete method of Siebel Account Business Service:

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/Delete**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<body>
 <SiebelMessage>
 <MessageId></MessageId>
 <MessageType>Integration Object</MessageType>
 <IntObjectName>Account Interface</IntObjectName>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <ListOfAccount_spcInterface>
 <Account>
 <Account_spcId>1-34Z</Account_spcId>
 </Account>
 </ListOfAccount_spcInterface>
 </SiebelMessage>
</body>
```

Using a Siebel CRM Business Service to Query an Account

You can use a Siebel CRM Business Service to query for an Account record by sending a HTTP POST request to the resource's URI.

The following request queries for an Account record with an ID=1-6 using the QueryById method of the Siebel Account Business Service:

- **URI: http://ServerName:port/siebel/v1.0/v1.0/service/Siebel Account/QueryById?PrimaryRowId=1-6**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<xml>
</xml>
```

Accessing the QueryByExample Method of the Siebel Account Business Service

You can use a Siebel CRM Business Service to query for an Account record by sending a HTTP POST request to the resource's URI.

The following request uses the QueryByExample method of the Siebel Account Business Service to query for an Account with the name 3Com:

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryByExample**
- **HTTP Method**: POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<body>
 <SiebelMessage>
 <MessageId></MessageId>
 <MessageType>Integration Object</MessageType>
 <IntObjectName>Account Interface</IntObjectName>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <ListOfAccount_spcInterface>
 <Account>
 <Name>3Com</Name>
 </Account>
 </ListOfAccount_spcInterface>
 </SiebelMessage>
</body>
```

Using a Siebel CRM Business Service to Update an Account

You can use a Siebel CRM Business Service resource to update an Account record by sending a HTTP POST request to the Repository Resource's URI.

The following request updates the Main Phone field for an existing Account record using the update method of the Siebel Account Business Service:

- **URI: http://ServerName:port/siebel/v1.0/service/Siebel Account/Update**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<body>
 <SiebelMessage>
 <MessageId></MessageId>
 <MessageType>Integration Object</MessageType>
 <IntObjectName>Account Interface</IntObjectName>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <ListOfAccount_spcInterface>
 <Account><Account_spcId>1-34Z</Account_spcId>
 <Main_spcPhone_spcNumber>2018742315</Main_spcPhone_spcNumber>
 </Account>
 </ListOfAccount_spcInterface>
 </SiebelMessage>
</body>
```

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true:

- **URI:**

  http://ServerName:port/siebel/v1.0/service/Siebel Account/QueryById?excludeEmptyFieldsInResponse=true
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<body>
 <PrimaryRowId>88-26CND</PrimaryRowId>
 <Number_of_Employees></Number_of_Employees>
</body>
```

- **Response body:** The response body returns Account with only non-empty fields.

7 Accessing Workflows

Using the Siebel REST API to Access Workflows

This chapter describes how to execute workflows from the Siebel REST API interface. You can invoke all active, completed workflows that are not of type interactive flow and long running flow from Siebel REST API.

For more information on HTTP methods supported by the Siebel REST API and the corresponding Siebel CRM operation, see *About Supported HTTP Methods*.

When accessing workflows using Siebel REST API, the following considerations apply:

- All executable workflows are listed in the Workflow Catalog API.
- You can use JSON or XML data as input.
- Workflow execution in Siebel REST API is access-controlled. To provide access, see *Configuring Siebel Workflows*.
- The API description can be obtained in OpenAPI format.

Note: Do not add responsibilities to the Workflow Process Manager business service. This can prevent general workflow REST APIs from executing correctly.

It includes the following topics:

- *Configuring Siebel Workflows*
- *About Using the Siebel REST API*
- *Using Siebel REST API to Access Workflows JSON Examples*
- *Using Siebel REST API to Access Workflows XML Example*

Configuring Siebel Workflows

Before you can execute workflows from Siebel REST API, you use Siebel Application to add responsibilities to specific workflows to allow access from Siebel REST API.

To add responsibilities to Siebel workflows

1. Log in as an administrator.
2. Navigate to the Administration - Application screen and then the Business Process Access view.
3. Add the required workflow to the Business Process list.
4. Add the required responsibility to the Responsibility list.

   Note: For more information about responsibilities and access control, see *Siebel Security Guide* .
5. Save your changes.
6. Click Clear Cache in both the Business Process and Responsibilities applets.

About Using the Siebel REST API

Each topic in this chapter provides both JSON and XML examples that demonstrate how to use the Siebel REST API calls to interact with Siebel Server resources.

Note: In Siebel CRM 21.2 Update and later, any changes to repository objects (such as a business service, business component or an integration object), do not require a restart of the object manager component used by REST. These changes are available immediately in the REST APIs once the changed workspace is delivered or migrated.

The example REST API calls use the following format:

- An example request, with the following information:
  - **URI.** The location of the Siebel REST API resource on the Siebel Server. For more information about Siebel REST API URL format, see *About Siebel CRM REST API URI Formats*.
  - **HTTP Method.** The HTTP method used to call the Siebel REST API to interact with the Siebel Server. For more information about supported HTTP Methods, *About Supported HTTP Methods*.
  - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is sent by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
  - **Request Body.** The payload for the Siebel REST API request.
- An example response, with the following information:
  - **HTTP Code.** The HTTP status code returned to indicate whether the request was successful or if there was an error. For more information about supported HTTP codes, *About Standard HTTP Status Codes and Error Messages*.
  - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is returned by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
  - **Response Body.** The Siebel REST API response.

Note: Because of the length of REST responses, some REST responses have been omitted.

Using Siebel REST API to Access Workflows JSON Examples

You can use the Siebel REST API to access workflows. You can invoke all active and completed workflows that are not interactive flow and long running flow types from Siebel REST API.

This topic includes information on the following workflow APIs:

- *Cataloging API*
- *Workflow API Description in OpenAPI Format*

- *Workflow Invocation as REST API*

Cataloging API

This example describes how to access a workflow catalog in Siebel REST API. This API lists all the workflows available in Siebel system for invocation. All interactive and long running type workflows are omitted from the list.

- **URI: https://ServerName:port/siebel/v1.0/workflow/describe**
- **HTTP Method:** GET
- **Accept:** application/json
- **Pagination inputs:** The following query parameters can be used for pagination:
  - PageSize
  - StartRowNum
- **Search specification (SearchSpec):**You can specify a search specification to catalog workflows based on a search criterion instead of cataloging all workflows in the repository. The following example returns all workflow names starting with EAI.

```
https://ServerName:port/siebel/v1.0/workflow/describe?SearchSpec=Name LIKE 'EAI*'
```

Workflow API Description in OpenAPI Format

This example describes how to access documentation for a workflow API in OpenAPI format. This API provides all the details of a specific workflow, such as the input and output format, in OpenAPI format.

- **URI: https://ServerName:port/siebel/v1.0/workflow/EAISiebDemo1/describe**
- **HTTP Method:** GET
- **Accept:** application/json

Workflow Invocation as REST API

This API enables you to run a workflow as Siebel REST API.

In this example below, you invoke a workflow named EAISiebDemo1. The inputs and outputs depict a workflow that accepts a Siebel message as input, and then responds with a Siebel message, in this case a EAI Siebel Adapter query.

- **URI: https://ServerName:port/siebel/v1.0/workflow/EAISiebDemo1**
- **HTTP Method:** POST
- **Accept:** application/json
- **Request body:**

```
{
 "Object Id":Opt_Id",
 "Process Instance Id":Opt_Id",
 "Siebel Operation Object Id":Opt_Id"
 "SiebelMessage":{
 "MessageId":"",
 "MessageType":"Integration Object",
 "IntObjectName":"EAI Account",
```

```
 "IntObjectFormat":"Siebel Hierarchical",
 "ListOfEAI Account":{
 "Account": [{
 "Name":"REST Acc7",
 "Location": "HQ-Distribution",
 "Description": "AccountData6",
 "ListOfContact": {
 "Contact": [ {
 "First Name": "Test_12367",
 "Middle Name": "Test_1236",
 "Last Name": "Test_12367"
 } ]
 }
 }]
 }
 }
}
```


- **Response body:**


```
{
 "InOut1": "",
 "Error Code": "",
 "Siebel Operation Object Id": "",
 "Process Instance Id": "88-2ZA5HG",
 "Error Message": "",
 "Object Id": "",
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "88-4JURI",
 "IntObjectName": "EAI Account",
 "MessageType": "Integration Object",
 "ListOfEAI Account": {
 "Account": [
 {
 "Freight Terms": "",
 "Location": "HQ-Distribution",
 "EAI Sync Status Code": "",
 "Region": "",
 "Home Page": "",
 "Name": "REST Acc7",
 "EAI Sync Date": "",
 "Last Manager Review Date": "",
 "Type": "Customer",
 "Freight Terms Info": "",
 "Competitor": "N",
 "Integration Id": "",
 "Price List Id": "",
 "DUNS Number": "",
 "Division": "",
 "GSA Flag": "",
 "Partner Flag": "N",
 "CSN": "88-2Z64JB",
 "Main Fax Number": "",
 "Price List": "",
 "Account Status": "Active",
 "Language Code": "ENU",
 "Alias": "",
 "Main Phone Number": "",
 "Description": "AccountData6",
 "Currency Code": "USD",
 "EAI Sync Error Text": "",
 "Prospect Flag": "N",
 "Managers Review": "",
 "Price List Integration Id": "",
 "ListOfContact": {
```

```
"Contact": [
{
"M/M": "",
"Active Status": "Y",
"Fax Phone #": "",
"Preferred Language Code": "",
"Email Address": "",
"Preferred Communications": "",
"First Name": "Test_12367",
"M/F": "",
"Middle Name": "Test_1236",
"Cellular Phone #": "",
"Job Title": "",
"Home Phone #": "",
"Assistant Phone #": "",
"Work Phone #": "",
"Integration Id": "",
"Last Name": "Test_12367",
"ListOfContact_Organization": {
"Contact_Organization": [
{
"Organization": "Default Organization",
"Organization Integration Id": ""
}
]
}
}
]
},
"ListOfAccount_Position": {
"Account_Position": [
{
"Position": "Siebel Administrator",
"Position Integration Id": ""
}
]
},
"ListOfAccount_Organization": {
"Account_Organization": [
{
"Organization": "Default Organization",
"Organization Id": "0-R9NH",
"Organization Integration Id": ""
}
]
}
}
]
}
}
```


In the example below, you invoke a workflow named eSales - Save Account Shipping Address. However, the inputs and outputs depict a workflow that accepts address details as input, and then responds with an updated CUT address.

- **URL: https://ServerName:port/siebel/v1.0/workflow/eSales - Save Account Shipping Address**
- **HTTP Method:** POST
- **Accept:** application/json
- **Request body:**


```
{
 "Object Id":"89-ZCBD",
 "Address Id": "89-ZCBY",
 "Street Address":"Street address sample1",
```

```
 "State":"CA",
 "City":"SM"
}
```

- **Response body:**

```
{
 "Error Code": "",
 "Address Id": "89-ZCBY",
 "Street Address": "Street address sample1",
 "Siebel Operation Object Id": "89-ZCBY",
 "Process Instance Id": "88-2ZO387",
 "Object Id": "89-ZCBD",
 "Error Message": "",
 "Street Address 2": "",
 "City": "SM",
 "State": "CA",
 "Country": "",
 "Postal Code": ""
}
```

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true:

- **URI:**

  http://ServerName:port/siebel/v1.0/workflow/eSales - Save Account Shipping Address?excludeEmptyFieldsInResponse=true

- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "Object Id": "89-ZCBD",
 "Address Id": "89-ZCBY",
 "Street Address": "Street address sample1",
 "State": "CA",
 "City": "SM",
 "Postal Code": ""
}
```

- **Response body:** The response body returns Account with only non-empty fields.

Using Siebel REST API to Access Workflows XML Example

You can use the Siebel REST API to access workflows. You can invoke all active, completed workflows, that are not of type interactive flow and long running flow, from Siebel REST API. To accept input in xml format and provide the response in XML format, the Content- Type header should be application/xml. The request should be wrapped in `<request>` tag and response will be wrapped in a `<response>` tag.

This topic includes information on the following workflow API:

- *Workflow Invocation as REST API*

Workflow Invocation as REST API

This API enables you to access a workflow in Siebel REST API.

In the example below, you invoke a workflow named EAISiebDemo1. The inputs and outputs depict a workflow that accepts a Siebel message as input, and then responds with a Siebel message, in this case a EAI Siebel Adapter query.

- **URI: https://ServerName:port/siebel/v1.0/workflow/EAISiebDemo1**
- **HTTP Method:** POST
- **Accept:** application/xml
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
 <SiebelMessage>
 <MessageId />
 <MessageType>Integration Object</MessageType>
 <IntObjectName>EAI Account</IntObjectName>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <ListOfEAI_spcAccount>
 <Account>
 <Name>REST Acc7</Name>
 <Location>HQ-Distribution</Location>
 </Account>
 </ListOfEAI_spcAccount>
 </SiebelMessage>
</request>
```

- **Response body:**

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
 <InOut1 />
 <Error_spcCode />
 <Siebel_spcOperation_spcObject_spcId />
 <Process_spcInstance_spcId>88-2ZA5HY</Process_spcInstance_spcId>
 <Error_spcMessage />
 <Object_spcId />
 <SiebelMessage>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <MessageId>88-4JURS</MessageId>
 <IntObjectName>EAI Account</IntObjectName>
 <MessageType>Integration Object</MessageType>
 <ListOfEAI_spcAccount>
```

```
<Account>
<Location>HQ-Distribution</Location>
<EAI_spcSync_spcStatus_spcCode />
<Region />
<Home_spcPage />
<Name>REST Acc7</Name>
<EAI_spcSync_spcDate />
.........................
<Managers_spcReview />
<Price_spcList_spcIntegration_spcId />
<ListOfContact>
<Contact>
<Preferred_spcCommunications />
<First_spcName>Test_123467</First_spcName>
<M_slhF />
<Middle_spcName>Test_12346</Middle_spcName>
<Cellular_spcPhone_spc_pnd />
<Job_spcTitle />
<Home_spcPhone_spc_pnd />
.........................
<ListOfContact_undOrganization>
<Contact_undOrganization>
<Organization>Default Organization</Organization>
<Organization_spcIntegration_spcId />
</Contact_undOrganization>
</ListOfContact_undOrganization>
</Contact>
</ListOfContact>
<ListOfAccount_undPosition>
<Account_undPosition>
<Position>Siebel Administrator</Position>
<Position_spcIntegration_spcId />
</Account_undPosition>
</ListOfAccount_undPosition>
<ListOfAccount_undOrganization>
<Account_undOrganization>
<Organization>Default Organization</Organization>
<Organization_spcId>0-R9NH</Organization_spcId>
<Organization_spcIntegration_spcId />
</Account_undOrganization>
</ListOfAccount_undOrganization>
</Account>
</ListOfEAI_spcAccount>
</SiebelMessage>
</response>
```

In the example below, you also invoke a workflow named eSales - Save Account ShippingAddress. However, the inputs and outputs depict a workflow that accepts address details as input, and then responds with an updated CUT address.

- **URL: https://ServerName:port/siebel/v1.0/workflow/eSales - Save Account Shipping Address**
- **HTTP Method:** POST
- **Accept:** application/xml
- **Request body:**

```
<?xml version="1.0" ?>
<request>
 <Object_spcId>89-ZCBD</Object_spcId>
 <Address_spcId>89-ZCBY</Address_spcId>
 <Street_spcAddress>Street address sample1</Street_spcAddress>
 <State>CA</State>
 <City>SM</City>
</request>
```

- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <Error_spcCode></Error_spcCode>
 <Address_spcId>89-ZCBY</Address_spcId>
 <Street_spcAddress>Street address sample1</Street_spcAddress>
 <Siebel_spcOperation_spcObject_spcId>89-ZCBY</Siebel_spcOperation_spcObject_spcId>
 <Process_spcInstance_spcId>88-2ZO38H</Process_spcInstance_spcId>
 <Object_spcId>89-ZCBD</Object_spcId>
 <Error_spcMessage></Error_spcMessage>
 <Street_spcAddress_spc2></Street_spcAddress_spc2>
 <City>SM</City>
 <State>CA</State>
 <Country></Country>
 <Postal_spcCode></Postal_spcCode>
</response>
```

Removing Empty Fields Using the ExcludeEmptyFieldsInResponse Parameter

You can use the ExcludeEmptyFieldsInResponse = true parameter to remove empty fields in REST API responses and to reduce the response body size. By default, if you do not use this parameter, both empty and non-empty fields are returned in the REST response body. This applies to data, service, workspace, and workflow REST APIs. For more information on the values you can use with the ExcludeEmptyFieldsInResponse parameter, see *About URI Parameters*.

In the following example the excludeEmptyFieldsInResponse parameter is set to true:

- **URI:**

  http://ServerName:port/siebel/v1.0/workflow/eSales - Save Account Shipping Address?excludeEmptyFieldsInResponse=true
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<request>
 <Object_spcId>89-ZCBD</Object_spcId>
 <Address_spcId>89-ZCBY</Address_spcId>
 <Street_spcAddress>abcd</Street_spcAddress>
 <State>CA</State>
 <City>SM</City>
 <Country></Country>
</request>
```

- **Response body:** The response body returns Account with only non-empty fields.

8 Accessing Attachments

Using the Siebel REST API to Access Attachments

This chapter describes how Siebel REST API supports attaching external documents and files to records. For example, attaching a spreadsheet file to an account. You can use inbound REST APIs to develop applications that can attach files to records, allowing you to insert, update, delete, query, and download an attachment.

For more information on HTTP methods supported by the Siebel REST API and the corresponding Siebel CRM operation, see *About Supported HTTP Methods*.

When accessing attachments using Siebel REST API, the following considerations apply:

- In order to insert, update, delete, or query attachments, the integration components and business components must have an attachment integration component or business component.
- The contents of the attached file must be Base64 encoded, and must be added as business component field value in the request JSON or XML body.
- All create, read, update and delete (CRUD) operations are supported for attachment APIs and can be performed on a single or multiple attachment records at a time.
- The maximum number of attachments in a single REST request is restricted to 100.
- The attachment size is restricted by the heap size of the Tomcat and process memory size of the Siebel OM process.

It includes the following topics:

- *About Using the Siebel REST API*
- *Using Siebel REST API to Access Attachments JSON Examples*
- *Using Siebel REST API to Access Attachments XML Examples*
- *Using Siebel REST API Business Services to Access Attachments JSON Examples*
- *Using Siebel REST API Business Services to Access Attachments XML Examples*

About Using the Siebel REST API

Each topic in this chapter provides both JSON and XML examples that demonstrate how to use the Siebel REST API calls to interact with Siebel Server resources.

> **Note:** In Siebel CRM 21.2 Update and later, any changes to repository objects (such as a business service, business component or an integration object), do not require a restart of the object manager component used by REST. These changes are available immediately in the REST APIs once the changed workspace is delivered or migrated.

The example REST API calls use the following format:

- An example request, with the following information:

    - **URI.** The location of the Siebel REST API resource on the Siebel Server. For more information about Siebel REST API URL format, see *About Siebel CRM REST API URI Formats*.
    - **HTTP Method.** The HTTP method used to call the Siebel REST API to interact with the Siebel Server. For more information about supported HTTP Methods, *About Supported HTTP Methods*.
    - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is sent by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
    - **Request Body.** The payload for the Siebel REST API request.
- An example response, with the following information:
    - **HTTP Code.** The HTTP status code returned to indicate whether the request was successful or if there was an error. For more information about supported HTTP codes, *About Standard HTTP Status Codes and Error Messages*.
    - **Content-Type.** The part of the HTTP header that indicates the media type of the data that is returned by the Siebel REST API HTTP methods. For more information about supported HTTP headers, see *About Supported HTTP Header Fields*.
    - **Response Body.** The Siebel REST API response.

Note: Because of the length of REST responses, some REST responses have been omitted.

Using Siebel REST API to Access Attachments JSON Examples

You can use the Siebel REST API to access transactional attachments. Inbound REST APIs can be used to develop applications that can attach files to records, allowing you to insert, update, delete, query, and download an attachment.

This topic includes the following information:

- *Querying the Details of an Attachment*
- *Querying an Attachment to Return Attachment Contents*
- *Querying an Attachment to Return as a File*
- *Querying a Child Attachment*
- *Querying a Child Attachment to Return Attachment Contents*
- *Querying Multiple or All Attachments in an Account*
- *Querying Multiple Attachments to Return Content*
- *Inserting a Top-Level Attachment*
- *Updating a Single Record with Multiple Attachments*
- *Deleting an Attachment from an Account*
- *Inserting Multiple Top-Level Attachments*
- *Updating an Account with an Attachment*
- *Updating Multiple Accounts by Adding an Attachment to Each Account*
- *Updating Multiple Accounts by Adding Multiple Attachments to Each Account*

- *Inserting an Attachment into a Child Record*
- *Deleting Multiple Attachments from a Record*
- *Deleting a Single Attachment from Multiple Records*
- *Deleting Multiple Attachments from Multiple Records*
- *Deleting Multiple Attachments from a Child Record*

Querying the Details of an Attachment

You can retrieve the details of an attachment by using its record ID. The following request queries for the attachment with the ID: 1CE-77YE . This is done through the integration component named `SR Resolution Item` in the integration object `AttachmentDocIO`. The response returns the attachment details, such as Name, ID, Size, and so on. It also returns the `Res Attachment Id` field, the value of which is a link for users to download the file. In the URI below AttachmentDocIO refers to a Siebel Business Object that maps to the AttachmentDocIO Integration Object. SR Resolution Item is a Business Component in the AttachmentDocIO Business Object. It maps to an Integration Component in the AttachmentDocIO Integration Object.

- **URI: http://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution%20Item/1CE-77YE**
- **HTTP Method:** GET
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:**


```
{
 "ResFileDeferFlg": "R",
 "ResFileName": "Air Compressor Valve Repair and Replace",
 "Solution Type": "Resolution Item",
 "ResFileExt": "doc",
 "Internal Publish Flag": "",
 "Sent": "",
 "Res Attachment Id": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution
 Item/1CE-77YE?fields=Res Attachment Id",
 "Name": "1CE-77YE",
 "Created By Name": "ESADMIN",
 "ResFileSrcType": "FILE",
 "Comment": "",
 "ResFileSize": "3382",
 "Id": "1CE-77YE",
 "ResFileDockReqFlg": "N",
 "Updated By Name": "ESADMIN",
 "ResFileDockStatFlg": "E",
 "ResFileAutoUpdFlg": "Y",
 "Description": "",
 "ResFileDate": "10/31/2001 03:19:15",
 "ResFileSrcPath": "",
 "Link": [
 {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/1CE-77YE",
 "name": "SR Resolution Item"
 },
 {
 "rel": "canonical",
 "href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/1CE-77YE",
 "name": "SR Resolution Item"
 },
```

```
 {
 "rel": "child",
 "href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/1CE-77YE/
Catalog Category",
 "name": "Catalog Category"
 }
 ]
}
```

Querying an Attachment to Return Attachment Contents

You can query an attachment to return attachment contents by using the inlineattachment query parameter in the request URI. By setting the inlineattachment parameter to true, it returns the `Res Attachment Id` field. The returned value of this field is the attachment details, returned as a Base64 encoded string. The following is a truncated example of this returned value:

```
"Res Attachment Id": "ACFsH29qUDaf8RRRQAfebtnP516d+2Fn/AIaK8Q89fs3/AKTQ0UUi0f/Z..."
```

- **URI: http://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution %20Item/88-1X701B?inlineattachment=true**
- **HTTP Method:** GET
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
{
 "ResFileDeferFlg": "R",
 "ResFileName": "Att25test",
 "Solution Type": "Resolution Item",
 "ResFileExt": "jpg",
 "Internal Publish Flag": "",
 "Sent": "",
 "Res Attachment Id": "ACFsH29qUDaf8RRRQAfebtnP516d+2Fn/AIaK8Q89fs3/AKTQ0UUi0f/Z...",
 "Name": "88-1X701B",
 "Created By Name": "SADMIN",
 "ResFileSrcType": "FILE",
 "Comment": "",
 "ResFileSize": "184206",
 "Id": "88-1X701B",
 "ResFileDockReqFlg": "N",
 "Updated By Name": "SADMIN",
 "ResFileDockStatFlg": "E",
 "ResFileAutoUpdFlg": "Y",
 "Description": "",
 "ResFileDate": "09/28/2021 02:15:24",
 "ResFileSrcPath": "",
 "Link": [
 {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1X701B",
 "name": "SR Resolution Item"
 },
 {
 "rel": "canonical",
 "href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1X701B",
 "name": "SR Resolution Item"
```

```
 },
 {
 "rel": "child",
 "href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1X701B/
Catalog Category",
 "name": "Catalog Category"
 }
 ]
}
```

Querying an Attachment to Return Contents as a File

You can query an attachment to return attachment contents as a file by using the ?fields=Accnt Attachment Id query parameter in the request URI. This query only returns a single attachment as a file.

- **URI: http://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution %20Item/88-1X701B?fields=Accnt%20Attachment%20Id**
- **HTTP Method:** GET
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:** testfile.txt

Querying a Child Level Attachment

You can query a child attachment to return both the details of the child attachment, and a link to download the child attachment. In this example, the request queries a child attachment through the child integration component `Account Attachment`, which is attached to `Account`. The response contains details of `Account Attachment` and a link to download the attachment with the value `Accnt Attachment Id`. In this example, the first instance of Account in the URI refers to the Account Business Object. The second instance refers to the Account Business Component within that Business Object. The child of the Account Business Component is the Account Attachment Business Component. These all map to the out of the box Base Account Integration Object.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account %20Attachment/88-1X5M7F**
- **HTTP Method:** GET
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
{
 "AccntFileSize": "10720448",
 "AccntFileName": "AccountChildHuge",
 "Comment": "",
 "Account Id": "1-5GZO",
 "Id": "88-1X5M7F",
 "AccntFileDate": "09/16/2021 02:02:39",
 "AccntFileDockStatFlg": "E",
 "AccntFileSrcType": "FILE",
```

```
 "AccntFileAutoUpdFlg": "Y",
 "AccntFileDockReqFlg": "N",
 "AccntFileExt": "txt",
 "AccntFileDeferFlg": "R",
 "AccntFileSrcPath": "",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account
Attachment/88-1X5M7F?fields=Accnt Attachment Id",
 "Link": [
 {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M7F",
 "name": "Account Attachment"
 },
 {
 "rel": "canonical",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M7F",
 "name": "Account Attachment"
 },
 {
 "rel": "parent",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO",
 "name": "Account"
 }
 ]
}
```


Querying a Child Attachment to Return Attachment Contents

You can query a child attachment to return its contents by using the inlineattachment query parameter in the request URI. By setting the inlineattachment parameter to true, it returns the `Res Attachment Id` field. The returned value of this field is the attachment details, returned as a Base64 encoded string.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account %20Attachment/88-1X62NN?inlineattachment=True**
- **HTTP Method:** GET
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:**


```
{
 "AccntFileSize": "109",
 "AccntFileName": "multi_atta_File_acc_1",
 "Comment": "",
 "Account Id": "1-5GZO",
 "Id": "88-1X62NN",
 "AccntFileDate": "09/20/2021 23:28:26",
 "AccntFileDockStatFlg": "E",
 "AccntFileSrcType": "FILE",
 "AccntFileAutoUpdFlg": "Y",
 "AccntFileDockReqFlg": "N",
 "AccntFileExt": "txt",
 "AccntFileDeferFlg": "R",
 "AccntFileSrcPath": "",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw==",
 "Link": [
 {
```

```
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X62NN",
 "name": "Account Attachment"
 },
 {
 "rel": "canonical",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X62NN",
 "name": "Account Attachment"
 },
 {
 "rel": "parent",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO",
 "name": "Account"
 }
 ]
}
```

Querying Multiple or All Attachments in an Account

You can retrieve the details of all attachments associated with a record, in this example, `Account Attachment` is queried in the `Account` record. The response is a JSON array, where each element of the array contains attachment details, such as Name, ID, Size, and so on. It also returns the `Res Attachment Id` field, the value of which is a link for users to download the file.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account%20Attachment/**
- **HTTP Method:** GET
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
{
 "items": [
 {
 "AccntFileSize": "10720448",
 "AccntFileName": "AccountChildHuge",
 "Comment": "",
 "Account Id": "1-5GZO",
 "Id": "88-1X5M7F",
 "AccntFileDate": "09/16/2021 02:02:39",
 "AccntFileDockStatFlg": "E",
 "AccntFileSrcType": "FILE",
 "AccntFileAutoUpdFlg": "Y",
 "AccntFileDockReqFlg": "N",
 "AccntFileExt": "txt",
 "AccntFileDeferFlg": "R",
 "AccntFileSrcPath": "",
 "Res Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account
Attachment/88-1X5M7F?fields=Res Attachment Id",
 "Link": [
 {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M7F",
 "name": "Account Attachment"
 },
 {
 "rel": "canonical",
```

```
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M7F",
"name": "Account Attachment"
},
{
"rel": "parent",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO",
"name": "Account"
}
]
},
{
"AccntFileSize": "105553",
"AccntFileName": "Tesrtttxtttt1",
"Comment": "",
"Account Id": "1-5GZO",
"Id": "88-1X5M75",
"AccntFileDate": "09/15/2021 23:17:31",
"AccntFileDockStatFlg": "E",
"AccntFileSrcType": "FILE",
"AccntFileAutoUpdFlg": "Y",
"AccntFileDockReqFlg": "N",
"AccntFileExt": "txt",
"AccntFileDeferFlg": "R",
"AccntFileSrcPath": "",
"Res Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account
Attachment/88-1X5M75?fields=Res Attachment Id",
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M75",
"name": "Account Attachment"
},
{
"rel": "canonical",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M75",
"name": "Account Attachment"
},
{
"rel": "parent",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO",
"name": "Account"
}
]
}
],
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment",
"name": "Account Attachment"
}
}
```


Querying Multiple Attachments to Return Content

You can query multiple or all attachments associated with an account to return the attachment contents in the response body. In this example, by using the inlineattachment parameter, and setting it to true, it queries all the attachments associated with `Account`. The response is a JSON array in which each element of the array contains details of the

attachment such as Name, ID, Size, and so on. It also returns the `Res Attachment Id` field. The returned value of this field is the attachment contents, returned as a Base64 encoded string.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account%20Attachment/?inlineattachment=True**
- **HTTP Method:** GET
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:**


```
{
 "items": [
 {
 "AccntFileSize": "109",
 "AccntFileName": "multi_atta_File_acc_1",
 "Comment": "",
 "Account Id": "1-5GZO",
 "Id": "88-1X62NN",
 "AccntFileDate": "09/20/2021 23:28:26",
 "AccntFileDockStatFlg": "E",
 "AccntFileSrcType": "FILE",
 "AccntFileAutoUpdFlg": "Y",
 "AccntFileDockReqFlg": "N",
 "AccntFileExt": "txt",
 "AccntFileDeferFlg": "R",
 "AccntFileSrcPath": "",
 "Res Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw==",
 "Link": [
 {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X62NN",
 "name": "Account Attachment"
 },
 {
 "rel": "canonical",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X62NN",
 "name": "Account Attachment"
 },
 {
 "rel": "parent",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO",
 "name": "Account"
 }
 ]
 },
 {
 "AccntFileSize": "109",
 "AccntFileName": "multi_atta_File_acc_3",
 "Comment": "",
 "Account Id": "1-5GZO",
 "Id": "88-1X5M6B",
 "AccntFileDate": "09/20/2021 23:28:26",
 "AccntFileDockStatFlg": "E",
 "AccntFileSrcType": "FILE",
 "AccntFileAutoUpdFlg": "Y",
 "AccntFileDockReqFlg": "N",
 "AccntFileExt": "txt",
 "AccntFileDeferFlg": "R",
 "AccntFileSrcPath": "",
 "Res Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw==",
```

```
 "Link": [
 {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M6B",
 "name": "Account Attachment"
 },
 {
 "rel": "canonical",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M6B",
 "name": "Account Attachment"
 },
 {
 "rel": "parent",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO",
 "name": "Account"
 }
 ]
 }
 ],
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment?
inlineattachment=True",
 "name": "Account Attachment"
 }
```


Inserting a Parent Level Attachment

This example describes how to insert a top-level attachment to a record. In this example, the record is called `SR Resolution Item`. The attachment contents are provided in the `Res Attachment Id` field as a Base64 encoded value.

- **URI: https://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution%20Item**
- **HTTP Method:** POST
- **Accept:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "ResFileName": "Demo_11",
 "ResFileExt": "txt",
 "Res Attachment Id": "5XAHcAMABLAEkAQwBCADcARABRAG8AZwBJAEM...."
}
```

- **Response body:**

```
{
 "items": {
 "ResFileExt": "txt",
 "Res Attachment Id": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution
 Item/88-1X70T3?fields=Res Attachment Id",
 "Id": "88-1X70T3",
 "ResFileName": "Demo_11",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1X70T3",
 "name": "SR Resolution Item"
 }
 }
```

```
}
```

Updating a Single Record with Multiple Attachments

You can update a single account to add multiple attachments using either of the two methods described below.

Updating a Single Record with Multiple Attachments: Specifying the ID in the Request

This example uses single object operations syntax to perform the update, whereby the ID is specified in the URI.

This example describes how to insert multiple attachments to a single record. In this example, an account is updated with three additional files. The request body is an array, and each array element contains attachment details, such as the file name. The attachment contents are provided in the `Accnt Attachment Id` field as a Base64 encoded value. The response body contains the attachment contents, attachment IDs, and a link to download the new attachments.

- **URI: https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account%20Attachment/**
- **HTTP Method:** PUT
- **Content-Type:** application/json
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**

```
{
 "Account Attachment": [
 {
 "AccntFileName": "multi_atta_File_acc_1",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "AccntFileName": "multi_atta_File_acc_2",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "AccntFileName": "multi_atta_File_acc_3",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 }
 ]
}
```

- **Response body:**

```
{
 "items": {
 "Id": "1-5GZO",
 "Account Attachment": [
 {
 "Id": "88-1X5M6B",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account
 Attachment/88-1X5M6B?fields=Accnt Attachment Id",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X5M6B",
 "name": "Account Attachment"
```

```
}
},
{
"Id": "88-1X62NN",
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account
Attachment/88-1X62NN?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X62NN",
"name": "Account Attachment"
}
},
{
"Id": "88-1X62NO",
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account
Attachment/88-1X62NO?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1X62NO",
"name": "Account Attachment"
}
}
],
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO",
"name": "Account"
}
}
}
```


Updating a Record with Multiple Attachments: Specifying the ID in an Array

This example uses multi-object object operations syntax to perform the update, whereby the ID is contained in the array body.

This example describes how to update an account by adding several attachments. The request body has an array with one account, and the child attachment, `Account Attachment`, is an array of attachment objects. The attachment contents are provided in the `Accnt Attachment Id` field as a Base64 encoded value. The response body contains the attachment contents, attachment IDs, and a link to download the new attachments.

- **URL: https://ServerName:port/siebel/v1.0/data/Account/Account/**
- **HTTP Method:** PUT
- **Accept:** application/json
- **Content-Type:** multipart/form-data
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**


```
[
{
"Name": "SingleAccountMultiAttachments",
"Location": "HQ-Distribution",
"Primary Organization Id": "1-1DG",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Description": "Updated AccountData 1 new ",
"Account Attachment": [
{
"AccntFileName": "SingleAccountMultiAttachments_file_1",
"AccntFileExt": "txt",
```

```
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "AccntFileName": "SingleAccountMultiAttachments_file_2",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "AccntFileName": "SingleAccountMultiAttachments_file_3",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 }
 ]
 }
]
```


- **Response body:**


```
{
 "items": [
 {
 "Name": "SingleAccountMultiAttachments",
 "Id": "88-1X5MBT",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 1 new ",
 "Account Attachment": [
 {
 "Id": "88-1X5MBW",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBW?fields=Accnt Attachment Id",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBW",
 "name": "Account Attachment"
 }
 },
 {
 "Id": "88-1X5MBX",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBX?fields=Accnt Attachment Id",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBX",
 "name": "Account Attachment"
 }
 },
 {
 "Id": "88-1X5MBY",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBY?fields=Accnt Attachment Id",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBY",
 "name": "Account Attachment"
 }
 }
 ],
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT",
 "name": "Account"
```

```
 }
 }
 ]
}
```

Deleting a Single Attachment in an Account

You can delete a single attachment attached to an account using either of the two methods described below.

Deleting an Attachment: Specifying the ID in the Request

This example uses single object operations syntax to delete an attachment, which is identified by its ID in the URL.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account%20Attachment/88-1X5EPZ**
- **HTTP Method:** DELETE
- **Accept:** application/json
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
{
 "items": {
 "Id": "1-5GZO",
 "operation": "skipnode"
 }
}
```

Deleting an Attachment: Specifying the ID in an Array

This example uses multi-object object operations syntax to delete an attachment. The attachment ID is contained in the array body.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account**
- **HTTP Method:** DELETE
- **Accept:** application/json
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Id": "88-1X5MBB",
 "Account Attachment": [
 {
 "Id": "88-1X5MBF"
 }
 ]
 }
]
```

- **Response body:**

```
[
 {
 "Id": "88-1YQ5LV",
 "operation": "skipnode"
 }
]
```

Inserting Multiple Parent Level Attachments

You can insert multiple top-level attachments to a record.

The following details describe how to upload several top-level attachments, where the request body is an array, and each element has attachment details, such as the `ResFileName` and `ResFileExt`fields and contents of the file to be attached as Base64 encoded string in the `Res Attachment Id` field. The response body will have the details of all the attachments inserted along with link to download each of the new attachments.

- **URI: https://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution%20Item**
- **HTTP Method:** POST
- **Accept:** application/json
- **Content-Type:** multipart/form-data
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**

```
[{
 "ResFileName": "DemoTest_12",
 "ResFileExt": "txt",
 "Res Attachment Id": "//5XAHcAMABLAEkAQwBCADcARABRAG8AZwBJAEMAQQBnAEkAbAB..."
},
{
 "ResFileName": "DemoTest_13",
 "ResFileExt": "txt",
 "Res Attachment Id": "//5XAHcAMABLAEkAQwBCADcARABRAG8AZwBJAEMAQQBnAEkAbAB..."
},
{
 "ResFileName": "DemoTest_14",
 "ResFileExt": "txt",
 "Res Attachment Id": "//5XAHcAMABLAEkAQwBCADcARABRAG8AZwBJAEMAQQBnAEkAbAB..."
}]
```

- **Response body:**

```
[
{
"ResFileExt": "txt",
"Res Attachment Id": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution
 Item/88-1X70TI?fields=Res Attachment Id",
 "Id": "88-1X70TI",
"ResFileName": "DemoTest_12", "Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1X70TI",
 "name": "SR Resolution Item"
}
```

```
],
},
{
"ResFileExt": "txt",
"Res Attachment Id": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution
 Item/88-1X70TJ?fields=Res Attachment Id",
"Id": "88-1X70TJ",

"ResFileName": "DemoTest_13", "Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1X70TJ",
 "name": "SR Resolution Item"
}
],
},
{
"ResFileExt": "txt",
"Res Attachment Id": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution
 Item/88-1X70TK?fields=Res Attachment Id",
 "Id": "88-1X70TK",
"ResFileName": "DemoTest_14", "Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1X70TK",
 "name": "SR Resolution Item"
}
]
}
]
```


Updating a Single Account by Adding One Attachment

This example describes how to update an account by adding a single attachment. This is done by specifying the contents of the attachment as a Base64 encoded value in the `Accnt Attachment Id` field. The response contains the attachment details, as well as a link to download the new attachment.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/**
- **HTTP Method:** PUT
- **Content-Type:** multipart/form-data
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**


```
[
 {
 "Name": "SingleAccountSingleAttachment",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 1 new ",
 "Account Attachment": [
 {
 "AccntFileName": "SingleAccountSingleAttachment_file",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 }
 ]
 }
```

```
]
```

- **Response body:**

```
{
 "items": [
 {
 "Name": "SingleAccountSingleAttachment",
 "Id": "88-1X5MBT",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 1 new ",
 "Account Attachment": [
 {
 "Id": "88-1X5MBW",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBW?fields=Accnt Attachment Id",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
 Attachment/88-1X5MBW",
 "name": "Account Attachment"
 }
 }
 ],
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT",
 "name": "Account"
 }
 }
 ]
}
```

Updating a Multiple Accounts by Adding One Attachment to Each Account

This example describes how to update multiple accounts by inserting an attachment in each account. This request body is an array of accounts, and each array element contains details of the account where the attachment will be added. Each account has an attachment as a child element, with attachment details, such as name and the attachment content, provided as a Base64 encoded string. The response has an array of account objects and details of the added attachments.

- **URI: https://ServerName:port/siebel/v1.0/data/Account/Account/**
- **HTTP Method:** PUT
- **Accept:** application/json
- **Content-Type:** multipart/form-data
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Name": "MultiAccountSingleAttachmentEach",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
```

```
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Description": "Updated AccountData 1 new ",
"Account Attachment": [
{
"AccntFileName": "MultiAccountSingleAttachmentEach_file",
"AccntFileExt": "txt",
"Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
}
]
},
{
"Name": "MultiAccountExample43",
"Location": "HQ-Distribution",
"Primary Organization Id": "1-1DG",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Description": "Updated AccountData 2 new",
"Account Attachment": [
{
"AccntFileName": "MultiAccountSingleAttachmentEach_file",
"AccntFileExt": "txt",
"Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
}
]
}
]
```


- **Response body:**


```
{
"items": [
{
"Name": "MultiAccountSingleAttachmentEach",
"Id": "88-1X5MBT",
"Location": "HQ-Distribution",
"Primary Organization Id": "1-1DG",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Description": "Updated AccountData 1 new ",
"Account Attachment": [
{
"Id": "88-1X5MBW",
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBW?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBW",
"name": "Account Attachment"
}
}
],
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT",
"name": "Account"
}
},
{
"Name": "SingleAccountMultiChildAtt2",
"Id": "88-1X5MBZ",
"Location": "HQ-Distribution",
"Primary Organization Id": "1-1DG",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Description": "Updated AccountData 1 new ",
"Account Attachment": [
{
"Id": "88-1X5MC2",
```

```
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC2?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC2",
"name": "Account Attachment"
}
}
],
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ",
"name": "Account"
}
}
]
}
```


Updating a Multiple Accounts by Adding Multiple Attachments to Each Account

This example describes how to update multiple accounts by inserting multiple attachments in each account. Multiple accounts are upserted to add multiple attachments to each account. The request body is an array of accounts objects. Each account object contains details of the account where the attachment will be added, and a child array of attachment objects. The response body has an array of account objects, each containing an array of attachment objects.

- **URI: https://ServerName:port/siebel/v1.0/data/Account/Account/**
- **HTTP Method:** PUT
- **Accept:** application/json
- **Content-Type:** multipart/form-data
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**


```
[
{
"Name": "MultiAccountMultiAttachmentsEach_1",
"Location": "HQ-Distribution",
"Primary Organization Id": "1-1DG",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Description": "Updated AccountData 1 new ",
"Account Attachment": [
{
"AccntFileName": "MultiAccountMultiAttachmentsEach_file_1",
"AccntFileExt": "txt",
"Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
},
{
"AccntFileName": "MultiAccountMultiAttachmentsEach_file_2",
"AccntFileExt": "txt",
"Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
},
{
"AccntFileName": "MultiAccountMultiAttachmentsEach_file_3",
"AccntFileExt": "txt",
"Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
```

```
 }
 ]
 },
 {
 "Name": "MultiAccountMultiAttachmentsEach_2",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 2 new",
 "Account Attachment": [
 {
 "AccntFileName": "MultiAccountMultiAttachmentsEach_file_1",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "AccntFileName": "MultiAccountMultiAttachmentsEach_file_2",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "AccntFileName": "MultiAccountMultiAttachmentsEach_file_3",
 "AccntFileExt": "txt",
 "Accnt Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 }
 ]
 }
]
```


- **Response body:**


```
{
 "items": [
 {
 "Name": "MultiAccountMultiAttachmentsEach_1",
 "Id": "88-1X5MBT",
 "Location": "HQ-Distribution",
 "Primary Organization Id": "1-1DG",
 "Primary Organization": "Millennium Institutional Finance Services IF ENU",
 "Description": "Updated AccountData 1 new ",
 "Account Attachment": [
 {
 "Id": "88-1X5MBW",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBW?fields=Accnt Attachment Id",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBW",
 "name": "Account Attachment"
 }
 },
 {
 "Id": "88-1X5MBX",
 "Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBX?fields=Accnt Attachment Id",
 "Link": {
 "rel": "self",
 "href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBX",
 "name": "Account Attachment"
 }
 },
 {
 "Id": "88-1X5MBY",
```

```
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBY?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT/Account
Attachment/88-1X5MBY",
"name": "Account Attachment"
}
}
],
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBT",
"name": "Account"
}
},
{
"Name": "MultiAccountMultiAttachmentsEach_2",
"Id": "88-1X5MBZ",
"Location": "HQ-Distribution",
"Primary Organization Id": "1-1DG",
"Primary Organization": "Millennium Institutional Finance Services IF ENU",
"Description": "Updated AccountData 1 new ",
"Account Attachment": [
{
"Id": "88-1X5MC2",
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC2?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC2",
"name": "Account Attachment"
}
},
{
"Id": "88-1X5MC3",
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC3?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC3",
"name": "Account Attachment"
}
},
{
"Id": "88-1X5MC4",
"Accnt Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC4?fields=Accnt Attachment Id",
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ/Account
Attachment/88-1X5MC4",
"name": "Account Attachment"
}
}
],
"Link": {
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5MBZ",
"name": "Account"
}
}
]
```

```
}
```

Inserting an Attachment into a Child Account Quote

This example describes how to update an account by inserting an attachment into a child record, in this instance `Quote` is a child object of `Account`. You update multiple quote objects under the parent account object to add multiple attachments under each quote. The response body contains details and links of the attachments inserted.

- **URI: https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote**
- **HTTP Method:** PUT
- **Accept:** application/json
- **Content-Type:** multipart/form-data
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Id": "88-1X63EP",
 "Quote Attachment": [
 {
 "QuoteFileName": "SingleAccountMultiAttachments_file_1",
 "QuoteFileExt": "txt",
 "Quote Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "QuoteFileName": "SingleAccountMultiAttachments_file_2",
 "QuoteFileExt": "txt",
 "Quote Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "QuoteFileName": "SingleAccountMultiAttachments_file_3",
 "QuoteFileExt": "txt",
 "Quote Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 }
 ]
 },
 {
 "Id": "88-1X0T5H",
 "Quote Attachment": [
 {
 "QuoteFileName": "SingleAccountMultiAttachments_file_1",
 "QuoteFileExt": "txt",
 "Quote Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "QuoteFileName": "SingleAccountMultiAttachments_file_2",
 "QuoteFileExt": "txt",
 "Quote Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 },
 {
 "QuoteFileName": "SingleAccountMultiAttachments_file_3",
 "QuoteFileExt": "txt",
 "Quote Attachment Id": "RmlsZSBmb3IgYWNjb3VudCBhdHRhY2htZW50IGZvciBmYWNjb3VudCBudW1iZXIgMw=="
 }
 ]
 }
```

```
]
```

- **Response body:**

```
[
{
"Id": "1-5GZO",
"Quote": [
{
"Id": "88-1X0T5H",
"Quote Attachment": [
{

"Quote Attachment Id": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X0T5H/
Quote Attachment/88-1X62MT?fields=Quote Attachment Id",
"Id": "88-1X62MT",
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X0T5H/Quote
 Attachment/88-1X62MT",
"name": "Quote Attachment"
}
]
},
{

"Quote Attachment Id": " https://ServerName:port/siebel/v1.0/data /Account/Account/1-5GZO/
Quote/88-1X0T5H/Quote Attachment/88-1X62MU?fields=Quote Attachment Id ", "Id": "88-1X62MU",
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X0T5H/Quote
 Attachment/88-1X62MU",
"name": "Quote Attachment"
}
],
},
{
"Quote Attachment Id": " https://ServerName:port/siebel/v1.0/Account/Account/1-5GZO/Quote/88-1X0T5H/
Quote Attachment/88-1X62MV?fields=Quote Attachment Id ",


"Id": "88-1X62MV",
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X0T5H/Quote
 Attachment/88-1X62MV",
"name": "Quote Attachment"
}
}
],
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X0T5H", "name":
 "Quote"
}
]
},
{
"Id": "88-1X62N0",
"Quote Attachment": [
{
```

```
"Quote Attachment Id": " https://ServerName:port/siebel/v1.0/data /Account/Account/1-5GZO/
Quote/88-1X62N0/Quote Attachment/88-1X62N3?fields=Quote Attachment Id ",
"Id": "88-1X62N3",
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X62N0/Quote
 Attachment/88-1X62N3",
"name": "Quote Attachment"
}
]
},
{
"Quote Attachment Id": " https://ServerName:port/siebel/v1.0/data /Account/Account/1-5GZO/
Quote/88-1X62N0/Quote Attachment/88-1X62N4?fields=Quote Attachment Id ", "Id": "88-1X62N4",
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X62N0/Quote
 Attachment/88-1X62N4",
"name": "Quote Attachment"
}
]
},
{
"Quote Attachment Id": " https://ServerName:port/siebel/v1.0/data /Account/Account/1-5GZO/
Quote/88-1X62N0/Quote Attachment/88-1X62N5?fields=Quote Attachment Id ", "Id": "88-1X62N5",
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X62N0/Quote
 Attachment/88-1X62N5",
"name": "Quote Attachment"
}
]
}
]
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/88-1X62N0", "name":
 "Quote"
}
]
}
],
"Link": [
{
"rel": "self",
"href": "https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO", "name": "Account"
}
]
}
]
```


Deleting Multiple Attachments on a Single Record

The following request deletes two attachments, specified by their corresponding IDs, attached to one account:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/**
- **HTTP Method:** DELETE

- **Accept:** application/json
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Id": "88-1X5MBB",
 "Account Attachment": [
 {
 "Id": "88-1X5MBF"
 },
 {
 "Id": "88-1X5MBG"
 }
 ]
 }
]
```

- **Response body:** None.

Deleting a Single Attachment from Multiple Records

This request has two different parent account records each with an attachment record to delete. Each attachment is identified by its ID:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/**
- **HTTP Method:** DELETE
- **Accept:** application/json
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Id": "88-1X5MBB",
 "Account Attachment": [
 {
 "Id": "88-1X5MBF"
 }
 ]
 },
 {
 "Id": "88-1X5MBT",
 "Account Attachment": [
 {
 "Id": "88-1X5MBX"
 }
 ]
 }
]
```

- **Response body:** None.

Deleting Multiple Attachments from Multiple Records

The following request deletes two attachments from two records, that is, each record containing two attachments respectively. Each attachment is identified by its ID:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/**
- **HTTP Method:** DELETE
- **Accept:** application/json
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Id": "88-1X5MBB",
 "Account Attachment": [
 {
 "Id": "88-1X5MBF"
 },
 {
 "Id": "88-1X5MBG"
 }
 ]
 },
 {
 "Id": "88-1X5MBT",
 "Account Attachment": [
 {
 "Id": "88-1X5MBX"
 },
 {
 "Id": "88-1X5MBY"
 }
 ]
 }
]
```

- **Response body:** None.

Deleting Multiple Attachments from a Child Record

The following request deletes three attachments from the child record Quote. Each attachment is identified by its ID:

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Quote/**
- **HTTP Method:** DELETE
- **Accept:** application/json
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
[
 {
 "Id": "88-1X0T5H",
```

```
 "Quote Attachment": [
 {
 "Id": "88-1X63EF"
 },
 {
 "Id": "88-1X63EE"
 },
 {
 "Id": "88-1X63ED"
 }
 ]
 }
]
```

- **Response body:**

```
[
 {
 "Id": "1-5GZO",
 "operation": "skipnode",
 "Quote": [
 {
 "Id": "88-1X0T5H"
 }
 ]
 }
]
```

Using Siebel REST API to Access Attachments XML Examples

You can use the Siebel REST API to access transactional attachments. You can use inbound REST APIs to develop applications that can attach files to records, allowing you to insert, update, delete, query, and download an attachment. To view the response in XML format, the Content-Type header should be application/xml. The response will be in XML format wrapped in a `<response>` tag.

This topic includes the following information:

Querying the Details of a Top Level Attachment

You can retrieve the details of an attachment by using its record ID. The following request queries for the attachment with the ID: 88-1YPBGI . This is done through the integration component named `SR Resolution Item` in the integration object `AttachmentDocIO`. The response returns the attachment details, such as Name, ID, Size, and so on. It also returns

the `Res Attachment Id` field, the value of which is a link for users to download the file. In the URI below AttachmentDocIO refers to a Siebel Business Object that maps to the AttachmentDocIO Integration Object. SR Resolution Item is a Business Component in the AttachmentDocIO Business Object. It maps to an Integration Component in the AttachmentDocIO Integration Object.

- **URI: http://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution %20Item/88-1YPBGI**
- **HTTP Method:** GET
- **Accept:** application/xml
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <ResFileDeferFlg>R</ResFileDeferFlg>
 <ResFileName>SatAttXML</ResFileName>
 <Solution_spcType>Resolution Item</Solution_spcType>
 <ResFileExt>jpg</ResFileExt>
 <Internal_spcPublish_spcFlag></Internal_spcPublish_spcFlag>
 <Sent></Sent>
 <Res_spcAttachment_spcId>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution
 Item/88-1YPBGI?fields=Res Attachment Id</Res_spcAttachment_spcId>
 <Name>88-1YPBGI</Name>
 <Created_spcBy_spcName>SADMIN</Created_spcBy_spcName>
 <ResFileSrcType>FILE</ResFileSrcType>
 <Comment></Comment>
 <ResFileSize>140</ResFileSize>
 <Id>88-1YPBGI</Id>
 <ResFileDockReqFlg>N</ResFileDockReqFlg>
 <Updated_spcBy_spcName>SADMIN</Updated_spcBy_spcName>
 <ResFileDockStatFlg>E</ResFileDockStatFlg>
 <ResFileAutoUpdFlg>Y</ResFileAutoUpdFlg>
 <Description></Description>
 <ResFileDate>09/30/2021 02:40:05</ResFileDate>
 <ResFileSrcPath></ResFileSrcPath>
 <Link>
 <rel>self</rel>
 <href>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1YPBGI</href>
 <name>SR Resolution Item</name>
 </Link>
 <Link>
 <rel>canonical</rel>
 <href>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1YPBGI</href>
 <name>SR Resolution Item</name>
 </Link>
 <Link>
 <rel>child</rel>
 <href>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1YPBGI/Catalog
 Category</href>
 <name>Catalog Category</name>
 </Link>
</response>
```

Querying an Attachment to Return Attachment Contents

You can query an attachment to return the attachment contents by using the inlineattachment parameter. By setting the inlineattachment parameter to true, it returns the `Res Attachment Id` field. The returned value of this field is the attachment contents, returned as a Base64 encoded string. The following is a truncated example of this returned value:

```
<Res_spcAttachment_spcId>SW4gdGhlIGV4YW1wbGUgYmVsb3csIHdlIGRlbGV0ZSBhIHNpbm...</Res_spcAttachment_spcId>
```

- **URI: http://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution %20Item/88-1YPFR1?inlineattachment=True**
- **HTTP Method:** GET
- **Accept:** application/xml
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <ResFileDeferFlg>R</ResFileDeferFlg>
 <ResFileName>SatAttXML</ResFileName>
 <Solution_spcType>Resolution Item</Solution_spcType>
 <ResFileExt>jpg</ResFileExt>
 <Internal_spcPublish_spcFlag></Internal_spcPublish_spcFlag>
 <Sent></Sent>
 <Res_spcAttachment_spcId>SW4gdGhlIGV4YW1wbGUgYmVsb3csIHdlIGRlbGV0ZSBhIHNpbm...</
Res_spcAttachment_spcId>
 <Name>88-1YPFR1</Name>
 <Created_spcBy_spcName>SADMIN</Created_spcBy_spcName>
 <ResFileSrcType>FILE</ResFileSrcType>
 <Comment></Comment>
 <ResFileSize>140</ResFileSize>
 <Id>88-1YPFR1</Id>
 <ResFileDockReqFlg>N</ResFileDockReqFlg>
 <Updated_spcBy_spcName>SADMIN</Updated_spcBy_spcName>
 <ResFileDockStatFlg>E</ResFileDockStatFlg>
 <ResFileAutoUpdFlg>Y</ResFileAutoUpdFlg>
 <Description></Description>
 <ResFileDate>09/30/2021 07:36:55</ResFileDate>
 <ResFileSrcPath></ResFileSrcPath>
 <Link>
 <rel>self</rel>
 <href>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1YPFR1</href>
 <name>SR Resolution Item</name>
 </Link>
 <Link>
 <rel>canonical</rel>
 <href>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1YPFR1</href>
 <name>SR Resolution Item</name>
 </Link>
 <Link>
 <rel>child</rel>
 <href>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1YPFR1/Catalog
 Category</href>
 <name>Catalog Category</name>
 </Link>
</response>
```

Querying a Child Level Attachment

You can query a child level attachment to return both the details of the child level attachment, and a link to download the child level attachment. In this example, the request queries a child level attachment through the child integration component `Account Attachment`, which is attached to an account. The response contains details of `Account Attachment` and a link to download the attachment with the value `Accnt Attachment Id`. In this example, the first instance of Account in the URI refers to the Account Business Object. The second instance refers to the Account Business Component within that Business Object. The child of the Account Business Component is the Account Attachment Business Component. These all map to the out of the box Base Account Integration Object.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5SUL/Account%20Attachment/88-1YLSD3**
- **HTTP Method:** GET
- **Accept:** application/xml
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <AccntFileSize>610</AccntFileSize>
 <AccntFileName>Input_Params</AccntFileName>
 <Comment></Comment>
 <Account_spcId>1-5GZO</Account_spcId>
 <Id>88-1YLSD3</Id>
 <AccntFileDate>08/19/2021 01:34:19</AccntFileDate>
 <AccntFileDockStatFlg>E</AccntFileDockStatFlg>
 <AccntFileSrcType>FILE</AccntFileSrcType>
 <AccntFileAutoUpdFlg>Y</AccntFileAutoUpdFlg>
 <AccntFileDockReqFlg>N</AccntFileDockReqFlg>
 <AccntFileExt>xml</AccntFileExt>
 <AccntFileDeferFlg>R</AccntFileDeferFlg>
 <AccntFileSrcPath></AccntFileSrcPath>
 <Accnt_spcAttachment_spcId>https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account
 Attachment/88-1YLSD3?fields=Accnt Attachment Id</Accnt_spcAttachment_spcId>
 <Link>
 <rel>self</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1YLSD3</
href>
 <name>Account Attachment</name>
 </Link>
 <Link>
 <rel>canonical</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1YLSD3</
href>
 <name>Account Attachment</name>
 </Link>
 <Link>
 <rel>parent</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO</href>
 <name>Account</name>
 </Link>
</response>
```

Querying a Child Attachment to Return Attachment Contents

You can query a child attachment to return its contents by using the inlineattachment query parameter in the request URI. By setting the inlineattachment parameter to true, it returns the `Res Attachment Id` field. The returned value of this field is the attachment contents, returned as a Base64 encoded string.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5SUL/Account%20Attachment/88-1YLSD3?inlineattachment=True**
- **HTTP Method:** GET
- **Accept:** application/xml
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <AccntFileSize>610</AccntFileSize>
 <AccntFileName>Input_Params</AccntFileName>
 <Comment></Comment>
 <Account_spcId>1-5GZO</Account_spcId>
 <Id>88-1YLSD3</Id>
 <AccntFileDate>08/19/2021 01:34:19</AccntFileDate>
 <AccntFileDockStatFlg>E</AccntFileDockStatFlg>
 <AccntFileSrcType>FILE</AccntFileSrcType>
 <AccntFileAutoUpdFlg>Y</AccntFileAutoUpdFlg>
 <AccntFileDockReqFlg>N</AccntFileDockReqFlg>
 <AccntFileExt>xml</AccntFileExt>
 <AccntFileDeferFlg>R</AccntFileDeferFlg>
 <AccntFileSrcPath></AccntFileSrcPath>
 <Accnt_spcAttachment_spcId>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZ...</Accnt_spcAttachment_spcId>
 <Link>
 <rel>self</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1YLSD3</
href>
 <name>Account Attachment</name>
 </Link>
 <Link>
 <rel>canonical</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO/Account Attachment/88-1YLSD3</
href>
 <name>Account Attachment</name>
 </Link>
 <Link>
 <rel>parent</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/1-5GZO</href>
 <name>Account</name>
 </Link>
</response>
```

Querying an Attachment to Return Contents as a File

You can query an attachment to return the attachment contents by using the following query parameter in the request URI: param ?fields=Accnt Attachment Id. The contents of the attachment are returned as a file. This query only returns a single attachment as a file.

- **URI: http://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5SUL/Account%20Attachment/88-1YLSD3?fields=Accnt%20Attachment%20Id**
- **HTTP Method:** GET
- **Accept:** application/xml
- **Authorization:** Basic
- **Content-Type:** application/xml
- **Request body:** None
- **Response body:** test.jpeg

Inserting a Top-Level Attachment

This example describes how to insert a top-level attachment to a record. In this example, the record type is `SR Resolution Item`. The attachment contents are provided in the `Res_spcAttachment_spcId` field as a Base64 encoded value.

- **URI: https://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution%20Item**
- **HTTP Method:** POST
- **Accept:** application/xml
- **Content-Type:** application/xml
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<request>
 <ResFileName>SatAttXML</ResFileName>
 <ResFileAutoUpdFlg>Y</ResFileAutoUpdFlg>
 <ResFileDockReqFlg>Y</ResFileDockReqFlg>
 <ResFileExt>jpg</ResFileExt>
 <Res_spcAttachment_spcId>/9j/4AAQSkZJRgABAQEAkACQAAD/2wBDAAIBAQIBAQICAgICAg...</
Res_spcAttachment_spcId>
</request>
```

- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <items>
 <ResFileDockReqFlg>Y</ResFileDockReqFlg>
 <ResFileExt>jpg</ResFileExt>
 <Res_spcAttachment_spcId>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution
 Item/88-1YPBGI?fields=Res Attachment Id</Res_spcAttachment_spcId>
 <Id>88-1YPBGI</Id>
 <ResFileAutoUpdFlg>Y</ResFileAutoUpdFlg>
```

```
 <ResFileName>SatAttXML</ResFileName>
 <Link>
 <rel>self</rel>
 <href>https://ServerName:port/siebel/v1.0/data/SR Resolution Item/SR Resolution Item/88-1YPBGI</href>
 <name>SR Resolution Item</name>
 </Link>
 </items>
</response>
```

Inserting an Attachment into a Child Record

This example describes how to update an account by inserting an attachment `Account Attachment` into a child record. The attachment contents are provided in the `Accnt_spcAttachment_spcId` field as a Base64 encoded value.

- **URI: https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5SUL/Account%20Attachment**
- **HTTP Method:** PUT
- **Accept:** application/xml
- **Content-Type:** application/xml
- **Transfer encoding:** Chunked
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
 <AccntFileName>multi_atta_File_acc_33577</AccntFileName>
 <AccntFileExt>txt</AccntFileExt>
 <Accnt_spcAttachment_spcId> W1RBQkxFXQpTX0NPTlRBQ1QKCltVUERBVEVdCltTR...
 </Accnt_spcAttachment_spcId>
</request>
```

- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <items>
 <Id>88-1X5SUL</Id>
 <Account_spcAttachment>
 <Id>88-1X5SXX</Id>
 <Accnt_spcAttachment_spcId>https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5SUL/Account
 Attachment/88-1X5SXX?fields=Accnt Attachment Id</Accnt_spcAttachment_spcId>
 <Link>
 <rel>self</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5SUL/Account Attachment/88-1X5SXX</
href>
 <name>Account Attachment</name>
 </Link>
 </Account_spcAttachment>
 <Link>
 <rel>self</rel>
 <href>https://ServerName:port/siebel/v1.0/data/Account/Account/88-1X5SUL</href>
 <name>Account</name>
 </Link>
 </items>
</response>
```

Deleting an Attachment from an Account

This example describes how to delete an attachment, which is identified by its ID in the URI.

- **URI: http://ServerName:port/siebel/v1.0/data/AttachmentDocIO/SR%20Resolution%20Item/88-1YPBGI**
- **HTTP Method:** DELETE
- **Accept:** application/xml
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:** None
- **Response body:**

```
200 OK
```

Using Siebel REST API Business Services to Access Attachments JSON Examples

You can use Siebel REST API to invoke business services to access transactional attachments. You can use these to insert and retrieve attachments.

This topic includes the following information:

- *Invoking a Business Service to Insert an Attachment*
- *Invoking a Business Service to Retrieve Attachment Contents*

Invoking a Business Service to Insert an Attachment

In this example, the business service `RetrieveAttachmentService` contains the Insert method, which accepts the `AttachmentDocIO` integration object as input. The `AttachmentDocIO` integration object contains the `SR Resolution Item` integration component, which has a field named `Res Attachment Id`. The attachment contents are provided as Base64 encoded string value to the `Res Attachment Id` field. When you invoke the Insert method, the attachment with the file contents provided in the `Res Attachment Id` field is inserted. The response contains the attachment ID.

- **URI: http://ServerName:port/siebel/v1.0/service/RetrieveAttachmentService/Insert**
- **HTTP Method**: POST
- **Content-Type:** application/json
- **Transfer-Encoding:** chunked
- **Authorization:** Basic
- **Request body:**

```
{
 "body": {
```

```
 "LOVLanguageMode": "LDC",
 "SiebelMessageIn": {
 "IntObjectName": "AttachmentDocIO",
 "ListOfAttachmentDocIO": {
 "SR Resolution Item": {
 "Id": "11111111",
 "ResFileExt": "txt",
 "ResFileName": "ServiceREVUT2_DemoHuge_UN11",
 "Res Attachment Id": "//5XAHcAMABLAEkAQwBCADcARABRAG8AZwBJAEMAQQBnAEkAbABSAHAAZ..."
 }
 }
 }
 }
}
```

- **Response body:**

```
{
 "SiebelMessageOut":{
 "IntObjectFormat":"Siebel Hierarchical",
 "MessageId":"",
 "IntObjectName":"AttachmentDocIO",
 "MessageType":"Integration Object",
 "SR Resolution Item":{
 "Id":"88-305ZQO",
 "Mod Id":"0"
 }
 }
}
```

Invoking a Business Service to Retrieve Attachment Contents

In this example, the business service `RetrieveAttachmentService` contains the QueryPage method, which accepts the `AttachmentDocIO` integration object as input. The `AttachmentDocIO` integration object contains the `SR Resolution Item` integration component, which has a field named `Res Attachment Id`. When you invoke the QueryPage method, it returns the details of the attachment such as ID, extension, size, and so on. The attachment contents are a Base64 encoded string value in the `Res Attachment Id` field.

- **URI: http://ServerName:port/siebel/v1.0/service/RetrieveAttachmentService/QueryPage**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body": {
 "LOVLanguageMode": "LDC",
 "SiebelMessageIn": {
 "IntObjectName": "AttachmentDocIO",
 "ListOfAttachmentDocIO": {
 "SR Resolution Item": {
 "Id": "='88-1WV8GL'",
 "ResFileExt": "",
 "ResFileName": "",
 "Res Attachment Id": ""
 }
 }
 }
 }
```

```
}
```

- **Response body:**

```
{
 "SiebelMessageOut":{
 "IntObjectFormat":"Siebel Hierarchical",
 "MessageId":"",
 "IntObjectName":"AttachmentDocIO",
 "MessageType":"Integration Object",
 "SR Resolution Item":[
 {
 "ResFileExt":"pdf",
 "Res Attachment Id":"DQpodHRwOi8vc2xBwcmVzY2hvb2wgc2V0dGluZ3Mu",
 "Id":"88-1WV8GL",
 "ResFileName":"Test"
 }
 ]
 }
}
```

Using Siebel REST API Business Services to Access Attachments XML Examples

You can use Siebel REST API to invoke business services to access transactional attachments. You can use these to insert and retrieve attachments.

This topic includes the following information:

- *Invoking a Business Service to Insert an Attachment*
- *Invoking a Business Service to Retrieve Attachment Contents*

Invoking a Business Service to Insert an Attachment

In this example, the business service `RetrieveAttachmentService` contains the Insert method, which accepts the `AttachmentDocIO` integration object as input. The `AttachmentDocIO` integration object contains the `SR Resolution Item` integration component, which has a field named `Res Attachment Id`. The attachment contents are provided as Base64 encoded string value to the `Res_spcAttachment_spcId` field. When you invoke the Insert method, the attachment with the file contents provided in the `Res_spcAttachment_spcId` field is inserted. The response contains the attachment ID, for example: `<Id>88-1YPNAZ</Id>`.

- **URI: http://ServerName:port/siebel/v1.0/service/RetrieveAttachmentService/Insert**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Transfer-Encoding:** chunked
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<body>
 <LOVLanguageMode>LDC</LOVLanguageMode>
 <SiebelMessageIn>
```

```
 <MessageId></MessageId>
 <MessageType>Integration Object</MessageType>
 <IntObjectName>AttachmentDocIO</IntObjectName>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <ListOfAttachmentDocIO>
 <SR_spcResolution_spcItem>
 <ResFileName>SGService_Demo44</ResFileName>
 <ResFileExt>txt</ResFileExt>
 <Id>11111111</Id>
 <Res_spcAttachment_spcId>//5XAHcAMABLAEkAQwBCADcARABRAG8AZwBJAEMAQQBnAEkA.....</Res_spcAttachment_spcId>
 </SR_spcResolution_spcItem>
 </ListOfAttachmentDocIO>
 </SiebelMessageIn>
</body>
Response Body
```

- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <SiebelMessageOut>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <MessageId></MessageId>
 <IntObjectName>AttachmentDocIO</IntObjectName>
 <MessageType>Integration Object</MessageType>
 <SR_spcResolution_spcItem>
 <Id>88-1YPNAZ</Id>
 <Mod_spcId>0</Mod_spcId>
 </SR_spcResolution_spcItem>
 </SiebelMessageOut>
</response>
```

Invoking a Business Service to Retrieve Attachment Contents

In this example, the business service `RetrieveAttachmentService` contains the QueryPage method, which accepts the `AttachmentDocIo` integration object as input. The `AttachmentDocIo` integration object contains the `SR Resolution Item` integration component, which has a field named `Res Attachment Id`. When you invoke the QueryPage method, it returns the details of the attachment such as ID, extension, size, and so on. The attachment contents are a Base64 encoded string value in the `Res_spcAttachment_spcId` field.

- **URI: http://ServerName:port/siebel/v1.0/service/RetrieveAttachmentService/QueryPage**
- **HTTP Method:** POST
- **Content-Type:** application/xml
- **Authorization:** Basic
- **Request body:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<body>
 <LOVLanguageMode>LDC</LOVLanguageMode>
 <SiebelMessageIn>
 <MessageId></MessageId>
 <MessageType>Integration Object</MessageType>
 <IntObjectName>AttachmentDocIO</IntObjectName>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <ListOfAttachmentDocIO>
 <SR_spcResolution_spcItem>
 <ResFileName></ResFileName>
 <ResFileExt></ResFileExt>
 <Id>= '88-21J1QM'</Id>
```

```
 <Res_spcAttachment_spcId></Res_spcAttachment_spcId>
 </SR_spcResolution_spcItem>
 </ListOfAttachmentDocIO>
 </SiebelMessageIn>
</body>
```

- **Response body:**

```
<?xml version="1.0" ?>
<response>
 <IntObjectFormat>Siebel Hierarchical</IntObjectFormat>
 <MessageId></MessageId>
 <IntObjectName>AttachmentDocIO</IntObjectName>
 <MessageType>Integration Object</MessageType>
 <SR_spcResolution_spcItem>
 <ResFileExt>txt</ResFileExt>
 <Res_spcAttachment_spcId> //5XAHcAMABLAEkAQwBCADcARADkATABBADAASwBJAEMAQgA3AEQAUQBvAGcASQBDAEEAZwBJAGwAUgMABLAF
 </Res_spcAttachment_spcId>
 <Id>88-21J1QM</Id>
 <ResFileName>SGService_DemoHu45</ResFileName>
 </SR_spcResolution_spcItem>
</response>
```

9 Using Siebel REST API For Siebel Clinical

Using Siebel REST API For Siebel Clinical

This chapter describes Siebel Representational State Transfer (REST) Services for Siebel Clinical users and how to use them. It includes the following topics:

- *Configuring Siebel Clinical Users*
- *Using the Siebel REST API with Siebel Clinical*

Configuring Siebel Clinical Users

You can use Siebel Tools to configure Siebel Clinical users to pass default position and responsibility information in the Siebel REST response. For more information about Siebel Clinical, see *Siebel Clinical Trial Management System Guide* .

To configure Siebel Clinical Users

1. Log into Siebel Tools as an administrator.
2. Navigate to the Administration - Application screen and then the Business Service view.
3. In the Business Service list, select the Clinical User Provisioning Service Business Service.
4. In the Business Service Method list, select the CreateUser method.

   The selected Business Service method appears in the Business Service Method list view.
5. In the Business Service Method Arguments list, select Default Position.

   Note: The Default Position must belong to the Default Organization.
6. In the Business Service Method Arguments list, select Default Responsibility.
7. Step off the record to save changes.

Using the Siebel REST API with Siebel Clinical

You can use the Siebel REST API to create, synchronize, and delete Siebel Clinical users. For more information about Siebel Clinical, see *Siebel Clinical Trial Management System Guide* .

You can also use the Siebel REST API to invoke Business Services using Siebel server script techniques. For more information, see *Using Siebel Tools* .

When you create or synchronize Siebel Clinical users, you can pass default position and responsibility information in the Siebel REST API response by configuring the LS Clinical User Provisioning Service Business method. For more information, see *Configuring Siebel Clinical Users*.

Creating a Siebel Clinical User

This topic includes the following:

Creating a Siebel Clinical User

The following request creates a Siebel clinical user on the Siebel CRM Server:

- **URI: http://ServerName:port/siebel/v1.0/service/LS Clinical User Provisioning Service/CreateUser**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "Employee":
 {
 "First Name": "Cathy",
 "Last Name": "Rogers",
 "Login Name":"Cathy.Rogers",
 "Email Address":"Cathy.Rogers@oracle.com"
 }
 "ListOfPosition":
 {
 "Position":
 {
 "Name":"Cathy.Rogers",
 "Division":"Default Organization"
 }
 },
 "ListOfEmployee_Responsibility":
 {
 "Employee_Responsibility":
 {
 "Responsibility":"Clinical Research Associate
 }
 }
 }
}
```

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:** None.

Synchronizing a Siebel Clinical User

The following request synchronizes a Siebel clinical user on the Siebel CRM Server:

- **URI: http:/ServerName:port//siebel/v1.0/service/LS Clinical User Provisioning Service/SynchronizeUser**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "Employee":
 {
 "Login Name": "Cathy.Rogers",
 "Field 1": "Value 1",
 "Field 2": "Value 2",
 ……..
 }
 }
}
```

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:** None.

Deleting a Siebel Clinical User

When you use the Siebel REST API to delete a Siebel clinical user, the Siebel Clinical application performs a soft delete of the clinical user by stamping the Termination Date field with the Current Date value and removing the user's responsibilities.

The following request deletes a Siebel clinical user on the Siebel CRM Server:

- **URI: http://ServerName:port/siebel/v1.0/service/LS Clinical User Provisioning Service/TerminateUser**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body":
 {
 "Employee":
 {
 "First Name": "Cathy",
 "Last Name": "Rogers",
 "Login Name":"Cathy.Rogers",
 "EMail Addr":"Cathy.Rogers@oracle.com"
 }
 "ListOfPosition":
 {
```

```
"Position":
{
"Name":"Cathy.Rogers",
"Division":"Default Organization"
}
},
"ListOfEmployee_Responsibility":
{
"Employee_Responsibility":
{
"Responsibility":"Clinical Research Associate
}
}
}
}
```

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:** None.

Inserting SVT from External Application to Siebel Clinical

“Subject Visit Template” can also be created in Siebel Clinical using this REST API call apart from UI. This API allows you to insert the SVT with multiple versions, Visits and Activities. The Versions will be inserted with Approved Status and hence no modification will be allowed on these versions.

> **Note:** The user invoking this API should have “Digital Gateway Integration Responsibility” or “Siebel Administrator” responsibility.

A sample payload is given below. The payload has the following required fields:

- "SVT"- Name, ProtocolNumber (Name uniquely identifies SVT record)
- "Versions"- Version (Version uniquely identifies Versions record)
- "Visits"- ClinicalItem, Name, VisitType (ClinicalItem + Name combination uniquely identifies Visit record in the SVT). ClinicalItem is a 255 character length field.
- "Activities"- ClinicalItem, Description, FormId (ClinicalItem + FormId combination uniquely identifies Activities record). ClinicalItem is a 255 character length field. FormId is a 50 character length field.
- Repeating and Repeating Max count indicate that this particular Visit can be repeated for the specified number of times. However, this is neither restricted nor enforced based on these values.

External application uses below URI to insert SVT in to Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/service/LS Clinical SVT Service/upsert/**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body" : {
```

```
"SVTData" : {
"IntObjectFormat" : "Siebel Hierarchical",
"MessageId" : "1-51I",
"IntObjectName" : "LS Clinical SVT IO",
"MessageType" : "Integration Object",
"ListOfLS Clinical SVT IO" : {
"SVT" : [ {
"Name" : "SVT_001",
"ProtocolNumber" : "PT_Protocol_001",
"ListOfVersions" : {
"Versions" : [ {
"EndDate" : "09/08/2021",
"Version" : "Version 1",
"StartDate" : "09/08/2021",
"ChangeSummary" : "TestSummary",
"Comments" : "TestComments",
"ListOfVisits" : {
"Visits" : [ {
"Lead" : "0",
"ClinicalItem" : "ScreeningVisit_601",
"Name" : "SCR",
"Min" : "0",
"Repeating" : "N",
"LeadUnits" : "days",
"RepeatingMaxCount" : "",
"Priority" : "1",
"MinMaxUnits" : "days",
"Max" : "0",
"VisitType" : "Screening",
"CRFPages" : "10",
"Status" : "Screened",
"StatusTrackingVisitFlag" : "Y",
"SDVRequired" : "N",
"PageNumberstoVerify" : "8",
"ListOfActivities" : {
"Activities" : [ {
"Description" : "Date of Birth",
"ClinicalItem" : "ScreeningAct601",
"Required" : "Y",
"FormName" : "Demographics",
"FormId" : "Act601",
"Comments" : "TestActivityComments",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "15"

}, {
"Description" : "Gender",
"ClinicalItem" : "ScreeningAct602",
"Required" : "Y",
"FormName" : "Demographics",
"ItemSDVRequired" : "N",
"FormId" : "Act602",
"Comments" : "TestActivityComments2",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "30"

} ]
}
}, {
"Lead" : "4198",
"ClinicalItem" : "ScheduleableVisit601",
"Name" : "BWOCV3-BWOC(9)",
```

```
"Min" : "1",
"Repeating" : "Y",
"LeadUnits" : "days",
"RepeatingMaxCount" : "10",
"Priority" : "41",
"MinMaxUnits" : "days",
"Max" : "1",
"VisitType" : "Scheduleable",
"CRFPages" : "1",
"Status" : "Completed",
"StatusTrackingVisitFlag" : "Y",
"SDVRequired" : "N",
"PageNumberstoVerify" : "8",
"ListOfActivities" : {
"Activities" : [ {
"Description" : "Would you like to register visit?",
"ClinicalItem" : "ScheduleableAct601",
"Required" : "Y",
"FormName" : "Register Visit",
"FormId" : "Activity601",
"Comments" : "TestActivityComments",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "15"
} ]
}
}, {
"ClinicalItem" : "UnScheduleableVisit601",
"Name" : "UNSV",
"Repeating" : "N",
"LeadUnits" : "days",
"RepeatingMaxCount" : "",
"Priority" : "42",
"MinMaxUnits" : "days",
"VisitType" : "UnScheduleable",
"CRFPages" : "1",
"Status" : "FollowupDone",
"StatusTrackingVisitFlag" : "Y",
"SDVRequired" : "N",
"PageNumberstoVerify" : "8",
"ListOfActivities" : {
"Activities" : [ {
"Description" : "Would you like to register visit?",
"ClinicalItem" : "UnScheduleableAct601",
"Required" : "Y",
"FormName" : "Register Visit",
"FormId" : "Acti601",
"Comments" : "TestActivityComments",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "15"
} ]
}
} ]
}
},
{
"EndDate" : "10/11/2021",
"Version" : "Version 2",
"StartDate" : "10/11/2021",
"ChangeSummary" : "TestSummary",
"Comments" : "TestComments",
"ListOfVisits" : {
"Visits" : [ {
```

```
"Lead" : "0",
"ClinicalItem" : "ScreeningVisit602",
"Name" : "SCR",
"Min" : "0",
"Repeating" : "N",
"LeadUnits" : "days",
"RepeatingMaxCount" : "",
"Priority" : "1",
"MinMaxUnits" : "days",
"Max" : "0",
"VisitType" : "Screening",
"CRFPages" : "4",
"Status" : "Screened",
"StatusTrackingVisitFlag" : "Y",
"SDVRequired" : "N",
"PageNumberstoVerify" : "8",
"ListOfActivities" : {
"Activities" : [ {
"Description" : "Date of Birth",
"ClinicalItem" : "ScreeningAct602_1",
"Required" : "Y",
"FormName" : "Demographics",
"ItemSDVRequired" : "N",
"FormId" : "FormId602",
"Comments" : "TestActivityComments",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "15"
}, {
"Description" : "Phase",
"ClinicalItem" : "ScreeningAct602_2",
"Required" : "Y",
"FormName" : "Phase",
"FormId" : "FormId0602",
"Comments" : "TestActivityComments",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "15"
} ]
}
}, {
"Lead" : "527",
"ClinicalItem" : "ScheduleableVisit602",
"Name" : "BWOCV3-BWOC(9)",
"Min" : "1",
"Repeating" : "Y",
"LeadUnits" : "days",
"RepeatingMaxCount" : "10",
"Priority" : "43",
"MinMaxUnits" : "days",
"Max" : "1",
"VisitType" : "Scheduleable",
"CRFPages" : "1",
"Status" : "Screened",
"StatusTrackingVisitFlag" : "Y",
"SDVRequired" : "N",
"PageNumberstoVerify" : "8",
"ListOfActivities" : {
"Activities" : [ {
"Description" : "Would you like to register visit?",
"ClinicalItem" : "ScheduleableActivity602",
"Required" : "Y",
"FormName" : "Register Visit",
"FormId" : "FormId00602",
```

```
"Comments" : "TestActivityComments",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "15"
} ]
}
}, {
"ClinicalItem" : "UnScheduleableVisit602",
"Name" : "UNSV",
"Repeating" : "N",
"LeadUnits" : "days",
"RepeatingMaxCount" : "",
"Priority" : "44",
"MinMaxUnits" : "days",
"VisitType" : "UnScheduleable",
"CRFPages" : "1",
"Status" : "Screened",
"StatusTrackingVisitFlag" : "Y",
"SDVRequired" : "N",
"PageNumberstoVerify" : "8",
"ListOfActivities" : {
"Activities" : [ {
"Description" : "Would you like to register visit?",
"ClinicalItem" : "UnScheduleableActivity602",
"Required" : "Y",
"FormName" : "Register Visit",
"FormId" : "FormId604",
"Comments" : "TestActivityComments",
"Type" : "Procedure",
"Status" : "Active",
"PaymentFlag" : "Y",
"DurationMinutes" : "15"
} ]
}
} ]
}
} ]
}
} ]
}
}
}
```


Once the payload is inserted successfully:

- SVT is created with Versions, Visits and Activities.
- Unique Activities from the Payload (ClinicalItem + FormId) will also be inserted under new View 'EDC Item Mapping'.

For more info on EDC Item Mapping View, refer *Siebel Clinical Trial Management System Guide*.

The user can also enable the email notifications on successful insertion of SVT via API, by following the instructions provided in the “Setting Up Integration Between CTMS and Oracle Clinical One” chapter of *Siebel Clinical Trial Management System Guide*.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
```

```
 "ResponseMessage": "Success"
}
```


Querying Sites in Siebel Clinical

This API is used to query the Site details from Siebel Clinical. Along with Site information, it fetches all Contacts, Accounts and Addresses information from the particular Site.

Protocol Number is the required field in the payload. You must provide a valid Protocol Number in the payload.

The following sample request will query all Sites under given Protocol in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/service/LS Clinical Protocol Site Query/querypage/**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**


```
{
 "body": {
 "Protocol Number": "PT_Protocol_001",
 "PageSize": "100",
 "StartRowNum": "0"
 }
}
```


Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "NumOutputObjects": "2",
 "ResponseMessage": "Success",
 "LastPage": "true",
 "SiteData": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "88-873FX",
 "IntObjectName": "LS Clinical Protocol Site Query IO",
 "MessageType": "Integration Object",
 "Sites": [
 {
 "PI Salutation": "Dr.",
 "Site Number": "CT_Site_002",
 "PI Email": "sbaniker@line.com",
 "Site Row Id": "88-3H91E",
 "Site Time Zone": "",
 "Site Created": "12/12/2013 00:58:19",
 "Site Name": "Hibbing Mfg",
 "Site Initiation Date": "06/01/2013",
 "Protocol Number": "CT_Protocol_002",
 "Status": "Planned",
 "PI Phone Number": "2013254455",
 "PI Last Name": "Baniker",
 "Site Updated": "03/06/2014 12:18:43",
 "PI First Name": "Steven",
 "Contacts": {
 "Site State": "",
```

```
"Site Contact Role": "Principal Investigator",
"Site Country": "",
"Site Contact Fax": "2013254466",
"Site Contact Last Name": "Baniker",
"Site Contact Created": "12/12/2013 00:59:30",
"Site Province": "",
"Site Zip": "",
"Site Contact Updated": "12/12/2013 00:59:30",
"Site Contact First Name": "Steven",
"Site City": "",
"Site Address 2": "",
"Site Address 1": "",
"Site Contact Phone": "2013254455",
"Site Contact Email": "sbaniker@line.com"
},
"Addresses": {
"Site State": "CA",
"Site Address Type": "Payment",
"Site Address Updated": "12/12/2013 01:12:53",
"Site Country": "Russia",
"Site Address Name": "Address2",
"Site Province": "Quebec",
"Site Zip": "84944-6888",
"Site Address Created": "12/12/2013 01:03:24",
"Site City": "Actaeon",
"Site Address 2": "",
"Site Address 1": "27189 Lovelace Rd., Igor, Ri"
}
},
{
"PI Salutation": "",
"Site Number": "CT_Site_005",
"PI Email": "",
"Site Row Id": "88-3KNL6",
"Site Time Zone": "",
"Site Created": "03/06/2014 12:10:39",
"Site Name": "Hibbing Mfg",
"Site Initiation Date": "",
"Protocol Number": "CT_Protocol_002",
"Status": "Planned",
"PI Phone Number": "",
"PI Last Name": "Adler",
"Site Updated": "03/06/2014 12:18:02",
"PI First Name": "Mike",
"Contacts": {
"Site State": "",
"Site Contact Role": "Principal Investigator",
"Site Country": "",
"Site Contact Fax": "",
"Site Contact Last Name": "Adler",
"Site Contact Created": "03/06/2014 12:11:54",
"Site Province": "",
"Site Zip": "",
"Site Contact Updated": "03/06/2014 12:11:54",
"Site Contact First Name": "Mike",
"Site City": "",
"Site Address 2": "",
"Site Address 1": "",
"Site Contact Phone": "",
"Site Contact Email": ""
}
}
]
}
}
```

Querying Item Library in Siebel Clinical

This API is used to query the Item Library from EDC Item Mapping in SVT. It fetches all Item Libraries of an SVT under given Protocol where Integrate Flag is Y.

> **Note:** The user invoking this API should have “Digital Gateway Integration Responsibility” responsibility in Siebel. Otherwise, the user will get "access denied" error.

Protocol Number is the required field in the payload.

The following sample request will query Item Library under given Protocol in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/service/LS Clinical Item Library Service/querypage/**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body": {
 "Protocol Number": "PT_Protocol_001",
 "NewQuery": True,
 "PageSize": "10",
 "StartRowNum": "0"
 }
}
```

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
 "NumOutputObjects": "2",
 "ResponseMessage": "Success",
 "LastPage": "true",
 "ItemData": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "88-873FM",
 "IntObjectName": "LS Clinical Item Library IO",
 "MessageType": "Integration Object",
 "LS Clinical Item Library": [
 {
 "Description": "Date of Birth",
 "Form Name": "Demographics",
 "Integration Flag": "Y",
 "Name": "Act601ScreeningAct601",
 "Payment Amount": "",
 "Payment Flag": "N",
 "Protocol Number": "PT_Protocol_001",
 "Clinical Item": "ScreeningAct601",
 "Form Id": "Act601"
 },
 {
 "Description": "Would you like to register visit?",
 "Form Name": "Register Visit",
```

```
"Integration Flag": "Y",
"Name": "FormId00602ScheduleableActivity602",
"Payment Amount": "",
"Payment Flag": "N",
"Protocol Number": "PT_Protocol_001",
"Clinical Item": "ScheduleableActivity602",
"Form Id": "FormId00602"
}
]
}
}
```

Upserting Source Data Verification in Siebel Clinical

This API is used to update Source Data Verification (SDV) information at Subject level and CRF level. Before using this API, make sure Subjects & CRF records are created in Siebel Application. This API is also used to send SDV Completed flag & SDV Completed Date details to mark the CRF records as source verified. Multiple instances of Protocol, Subject and CRF can be used in a single payload.

Note: The user invoking this API should have “Digital Gateway Integration Responsibility” responsibility in Siebel. Otherwise, the user will get "access denied" error.

The payload has the following required fields:

- "Clinical Protocol"- Protocol Number
- "Clinical Subject"- EDC Internal Id , Screening Number
- "LS CRF Tracking BC"- Clinical Item, Screening Number, Subject EDC Internal Id

EDC Internal Id and Subject EDC Internal Id refer to the Integration Id for the Subject in Siebel Clinical. Subject Integration ID is the subject unique identifier in a Protocol. Screening Number refers to the Subject ID in Siebel Clinical.

The following request will upsert SDV data in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/service/LS Clinical SDV Service/upsert/**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"SDVData": {
"MessageId": "",
"MessageType": "Integration Object",
"IntObjectName": "LS Clinical SDV IO",
"IntObjectFormat": "Siebel Hierarchical",
"ListOfLS Clinical SDV IO": {
"Clinical Protocol": [
{
"Protocol Number": "PT_Protocol_001",
"ListOfClinical Subject": {
"Clinical Subject": [
{
"EDC Internal Id": "Int_Test_SVT_02",
"SDV Required": "No",
"Screening Number": "Test_SVT_02"
```

```
},
{
"EDC Internal Id": "Int_Test_SVT_01",
"SDV Required": "No",
"Screening Number": "Test_SVT_01"
}
]
},
"ListOfLS CRF Tracking BC": {
"LS CRF Tracking BC": [
{
"SDV Completed": "Y",
"SDV Completed Date": "01/06/2023",
"Screening Number": "Test_SVT_02",
"Subject EDC Internal Id": "Int_Test_SVT_02",
"Name": "Screening Visit",
"Clinical Item": "Screening Visit"
},
{
"SDV Completed": "Y",
"SDV Completed Date": "01/07/2023",
"Screening Number": "Test_SVT_01",
"Subject EDC Internal Id": "Int_Test_SVT_01",
"Name": "Screening Visit",
"Clinical Item": "Screening Visit"
}
]
}
}
]
}
}
}
```

This sample payload will result in:

- Updating the two Subjects with SDV Required Field to 'No'.
- It will also update the 'SDV Completed' and 'SDV Completed Date' fields of the given CRF Tracking records.

Note that:

- CRF Tracking Records may have Repeat Number = 0 or 1 or 2 or 3 etc.
- The above sample payload will update the CRF Records for Repeat Number = 0.
- Using this API, if the user wants to update the CRF Records which has Repeat Number other than 0, then the user needs to add the repeat number in parenthesis along with the 'Name' in the payload.

  For example:

  ```
  Screening
   Visit(1)
  ```

  Here “Screening Visit” is the visit name and number “1” inside parentheses is the repeat number. This is how the user can update the CRF records whose repeat number is other than 0.

  ```
  {
   "body": {
   "SDVData": {
  ```

```
"MessageId": "",
"MessageType": "Integration Object",
"IntObjectName": "LS Clinical SDV IO",
"IntObjectFormat": "Siebel Hierarchical",
"ListOfLS Clinical SDV IO": {
"Clinical Protocol": [
{
"Protocol Number": "PT_Protocol_001",
"ListOfClinical Subject": {
"Clinical Subject": [
{
"EDC Internal Id": "Int_Test_SVT_02",
"SDV Required": "No",
"Screening Number": "Test_SVT_02"
},
{
"EDC Internal Id": "Int_Test_SVT_01",
"SDV Required": "No",
"Screening Number": "Test_SVT_01"
}
]
},
"ListOfLS CRF Tracking BC": {
"LS CRF Tracking BC": [
{
"SDV Completed": "Y",
"SDV Completed Date": "01/06/2023",
"Screening Number": "Test_SVT_02",
"Subject EDC Internal Id": "Int_Test_SVT_02",
"Name": "Screening Visit(1)",
"Clinical Item": "Screening Visit"
},
{
"SDV Completed": "Y",
"SDV Completed Date": "01/07/2023",
"Screening Number": "Test_SVT_01",
"Subject EDC Internal Id": "Int_Test_SVT_01",
"Name": "Screening Visit(1)",
"Clinical Item": "Screening Visit"
}
]
}
}
]
}
}
}
}
```


If we need to update the CRF Record with "Name": "Screening Visit", having Repeat Number = 1, then we need to provide "Name": "Screening Visit(1)" in the payload.

If we need to update the CRF Record with "Name": "Screening Visit", having Repeat Number = 2, then we need to provide "Name": "Screening Visit(2)" in the payload.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "ResponseMessage": "Success"
```

```
}
```

Marking Visits as Planned or UnPlanned

This API accepts list of subject visits and marks them as 'Planned Visits'. Rest of the subject visits, which are not included in this API will be marked as 'Unplanned Visits' for that particular subject.

> **Note:** The user invoking this API should have “Digital Gateway Integration Responsibility” responsibility in Siebel. Otherwise, the user will get "access denied" error.

The Clinical Item field is used for Clinical One Integration.

Integration Id and Protocol Number are mandatory fields in Clinical Subject section and Clinical Item is mandatory field in Visit Plan section to use this API.

Populate “Clinical Item” field of a visit during Insert SVT API call. Once Subject is scheduled, this field's data will be copied from SVT to Subject Visit.

The following request will Mark Visits Planned/Unplanned in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/workflow/LS Clinical Subject Visit Plan**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "DATA": {
 "MessageId": "",
 "MessageType": "Integration Object",
 "IntObjectName": "LS Clinical Visit Plan Mark",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfLS Clinical Visit Plan Mark": {
 "Clinical Subject": [
 {
 "Protocol Number": "PT_Protocol_001",
 "Integration Id": "Int_Test_SVT_01",
 "ListOfVisit Plan": {
 "Visit Plan": [
 {
 "Clinical Item": "Re-screening Visit"
 },
 {
 "Clinical Item": "Screening Visit"
 }
 ]
 }
 }
 ]
 }
 }
}
```

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
 "Error Code": "",
 "ResponseMessage": "Success",
 "Error Message": ""
}
```

Using Inbound Subject API in Siebel Clinical

• This API can be used for the following multiple purposes:

- *Create and Schedule Subject in Siebel Clinical*
- *Complete Subject Visits and Activities*
- *Create and Complete on-the-fly Visits and Activities*
- *Undo Completed Subject Visit Activities*

Create and Schedule Subject in Siebel Clinical

This API can be used to create and schedule the Subject in Siebel Clinical. "Integration Id" is required to be sent and it should be the unique identifier of the Subject in a study. If the same Integration Id is used, the corresponding Subject data will be updated. If the same Integration Id is used with a different Site, then the Subject will be transferred to the new site, which is an existing functionality. It is required to send the subject status information while creating the subject, otherwise an error will be thrown. A single instance of Subject and Visit is allowed in a payload with multiple instances of activities.

Encounter Date, Screening Number, Site Number and Protocol Number are the required fields for Clinical Subject.

Status and Status Date are the required fields for Clinical Subject Status.

Screening Number refers to the Subject ID in Siebel Clinical.

Integration Id is the Unique Identifier of a Subject in a Study.

The following request will create and schedule Subject in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/workflow/LS Clinical Subject Inbound-Subject**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "OutputIntObjectName": {
 "MessageId": "",
 "MessageType": "Integration Object",
 "IntObjectName": "LS Clinical Subject Internal",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfLS Clinical Subject Internal": {
```

```
"Clinical Subject": [
{
"Encounter Date": "01/01/2023",
"Protocol Number": "PT_Protocol_001",
"Screening Number": "Test_SVT_01",
"Screen Date": "01/01/2023",
"Site Number": "PT_Site_001",
"Cohort": "10",
"Integration Id": "Int_Test_SVT_01",
"Enrollment Date": "01/01/2023",
"ListOfClinical Subject Status": {
"Clinical Subject Status": [
{
"Status Date": "01/01/2023",
"Status": "Screened",
"Comments": "test",
"Visit Type": "Screening"
}
]
},
"ListOfClinical Subject_Informed Consent": {
"Clinical Subject_Informed Consent": [
{
"Informed Consent Signed Date": "01/01/2023"
}
]
}
}
]
}
}
```


As a result of a successful call of this API:

- It creates a Subject with given details.
- It also Schedules the Subject if there is a valid SVT is attached to the Site.

  If no SVT is attached to the site (where subject is going to create), this API will return the "UNDEFINED_SITE_TEMPLATE" error message.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "#value": "<?xml version=\"1.0\" encoding=\"UTF-8\"?><SiebelMessage\n MessageId=\"\"\n IntObjectName=
\"LS Clinical Subject Internal\"\n MessageType=\"Integration Object\"\n IntObjectFormat=
\"Siebel Hierarchical\"\n><ListOfClinicalSubjectInternal\n><ClinicalSubject\n><Cohort\n>10</
Cohort\n><DateofBirth\n>01/01/2023</DateofBirth\n><EnrollmentDate\n>01/01/2023</EnrollmentDate
\n><IntegrationId\n>Int_Test_SVT_01</IntegrationId\n><ProtocolNumber\n>PT_Protocol_001</
ProtocolNumber\n><ProtocolSiteId\n>88-2WGGP</ProtocolSiteId\n><ScreenDate\n>01/01/2023</ScreenDate
\n><SiteNumber\n>PT_Site_001</SiteNumber\n><SubjectInitials\n>Test_SVT_01</SubjectInitials
\n><ListOfClinicalSubjectStatus\n><ClinicalSubjectStatus\n><Status\n>Screened</Status\n><StatusDate
\n>01/01/2023</StatusDate\n><Comments2\n>test</Comments2\n><VisitType\n>Screening</VisitType\n></
ClinicalSubjectStatus\n></ListOfClinicalSubjectStatus\n><ListOfClinicalSubject_InformedConsent
\n><ClinicalSubject_InformedConsent\n><InformedConsentSignedDate\n>01/01/2023</InformedConsentSignedDate
\n></ClinicalSubject_InformedConsent\n></ListOfClinicalSubject_InformedConsent\n></ClinicalSubject\n></
ListOfClinicalSubjectInternal\n></SiebelMessage\n>",
 "Error Code": "",
```

```
 "Siebel Operation Object Id": "88-34YFFX",
 "Process Instance Id": "88-34YFFV",
 "Integration Id": "Int_Test_SVT_01",
 "Error Message": ""
}
```

Complete Subject Visits and Activities

This API can also be used to complete Subject visits and activities in Siebel Clinical. The API supports completion of one visit with multiple activities at a time.

Protocol Number and Integration Id uniquely Identifies a Subject.

Clinical Item and Repeating Number uniquely Identifies a Visit.

The following sample request will complete Subject Visits and Activities in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/workflow/LS Clinical Subject Inbound-Subject**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

  The following sample payload will update the completed date for Clinical Item : "Re-screening Visit" and Repeat Number = 0

```
{
 "OutputIntObjectName": {
 "MessageId": "",
 "MessageType": "Integration Object",
 "IntObjectName": "LS Clinical Subject Internal",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfLS Clinical Subject Internal": {
 "Clinical Subject": [
 {
 "Protocol Number": "PT_Protocol_001",
 "Site Number": "SVT_001",
 "Integration Id": "Int_SVT_001",
 "ListOfVisit Plan": {
 "Visit Plan": [
 {
 "Clinical Item": "Re-screening Visit",
 "Completed Date": "11/22/2022",
 "Name": "Re-screening Visit",
 "ListOfVisit Plan Action": {
 "Visit Plan Action": [
 {
 "Clinical Item": "Re-screening Visit - act",
 "Done": "11/22/2022",
 "Form Id": "ReScreeningAct01",
 "Repeating Number": "0"
 },
 {
 "Clinical Item": "Re-screening Visit - act",
 "Done": "11/22/2022",
 "Form Id": "ReScreeningAct02",
 "Repeating Number": "0"
 }
 ]
 }
```

```
 }

 ]
 }
 }
 ]
 }
 }
}
```

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
 "#value": "<?xml version=\"1.0\" encoding=\"UTF-8\"?><SiebelMessage\n MessageId=\"\"\n IntObjectName=
\"LS Clinical Subject Internal\"\n MessageType=\"Integration Object\"\n IntObjectFormat=\"Siebel
 Hierarchical\"\n><ListOfClinicalSubjectInternal\n><ClinicalSubject\n><IntegrationId\n>Int_Test_SVT_02</
IntegrationId\n><ProtocolNumber\n>PT_Protocol_001</ProtocolNumber\n><ProtocolSiteId\n>88-2WGGP</
ProtocolSiteId\n><SiteNumber\n>PT_Site_001</SiteNumber\n><ListOfVisitPlan\n><VisitPlan\n><ClinicalItem
\n>Screening Visit</ClinicalItem\n><CompletedDate\n>1/11/2023</CompletedDate\n><Name\n>Screening Visit</
Name\n><ListOfVisitPlanAction\n><VisitPlanAction\n><ClinicalItem\n>Screening Visit - act02</ClinicalItem
\n><Done\n>1/11/2023</Done\n><FormId\n>ScreeningAct01</FormId\n><RepeatingNumber\n>0</RepeatingNumber
\n></VisitPlanAction\n><VisitPlanAction\n><ClinicalItem\n>Screening Visit - act01</ClinicalItem\n><Done
\n>1/11/2023</Done\n><FormId\n>ScreeningAct02</FormId\n><RepeatingNumber\n>2</RepeatingNumber\n></
VisitPlanAction\n></ListOfVisitPlanAction\n></VisitPlan\n></ListOfVisitPlan\n></ClinicalSubject\n></
ListOfClinicalSubjectInternal\n></SiebelMessage\n>",
 "Error Code": "",
 "Siebel Operation Object Id": "",
 "Process Instance Id": "88-34YFJX",
 "Integration Id": "Int_Test_SVT_02",
 "Error Message": ""
}
```

Create and Complete on-the-fly Visits and Activities

This API can also be used to create on-the-fly dynamic Visits and Activities by providing incremental repeating number. A dynamic visit is a subject visit which is created on-the-fly with Repeat Number other than 0 when the actual visit has happened.

Example Dynamic Visit names: Screening Visit(1), Rescreening Visit(1) etc.

The number included inside parentheses is called Repeat Number. When Siebel receives these kind of visit names, they will be treated as dynamic visits. The dynamic visits will be created under the given subject with visit name as “Screening Visit” and Repeat Number as 1 for example.

Protocol Number and Integration Id uniquely Identifies a Subject.

Clinical Item and Repeating Number uniquely Identifies a Visit.

The following request will create on-the-fly Subject Visits and Activities in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/workflow/LS Clinical Subject Inbound-Subject**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic

Request body:

The following sample payload will create and complete a new visit with Clinical Item : UnScheduleableVisit and Repeat Number = 1. This is called as Dynamic Visit. Observe the visit name in the following sample payload. It is UnScheduleableVisit(1). When the visit name comes to Siebel Clinical with some number enclosed inside parenthesis, system treats it as a dynamic visit, and the same will be created as a new visit in Siebel Clinical with Name as UnScheduleableVisit and Repeat Number as “1”. In this way, dynamic schedulable visits can be created on-the-fly, and dynamic unschedulable visits also can be created on-the-fly.

```
{
 "OutputIntObjectName": {
 "MessageId": "",
 "MessageType": "Integration Object",
 "IntObjectName": "LS Clinical Subject Internal",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfLS Clinical Subject Internal": {
 "Clinical Subject": [
 {
 "Protocol Number": "PT_Protocol_001",
 "Site Number": "SVT_001",
 "Integration Id": "Int_SVT_001",
 "ListOfVisit Plan": {
 "Visit Plan": [
 {
 "Clinical Item": "UnScheduleableVisit",
 "Completed Date": "11/20/2022",
 "Name": "UnScheduleableVisit(1)",
 "ListOfVisit Plan Action": {
 "Visit Plan Action": [
 {
 "Clinical Item": "UnScheduleableAct",
 "Done": "11/20/2022",
 "Form Id": "UnScheduleableAct01",
 "Repeating Number": "2"
 },
 {
 "Clinical Item": "UnScheduleableAct",
 "Done": "11/20/2022",
 "Form Id": "UnScheduleableAct02",
 "Repeating Number": "2"
 }
 ]
 }
 }
 ]
 }
 }
 ]
 }
 }
}
```

If you need to create further Dynamic/On-the-fly Visit for Repeat Number =2, you can provide Name in the above sample as "Name": "UnScheduleableVisit(2)"

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

  ```
  {
  ```

```
 "#value": "<?xml version=\"1.0\" encoding=\"UTF-8\"?><SiebelMessage\n MessageId=\"\"\n IntObjectName=
\"LS Clinical Subject Internal\"\n MessageType=\"Integration Object\"\n IntObjectFormat=
\"Siebel Hierarchical\"\n><ListOfClinicalSubjectInternal\n><ClinicalSubject\n><IntegrationId
\n>Int_Test_SVT_02</IntegrationId\n><ProtocolNumber\n>PT_Protocol_001</ProtocolNumber\n><ProtocolSiteId
\n>88-2WGGP</ProtocolSiteId\n><SiteNumber\n>PT_Site_001</SiteNumber\n><ListOfVisitPlan\n><VisitPlan
\n><ClinicalItem\n>UnScheduleableVisit</ClinicalItem\n><CompletedDate\n>1/10/2023</CompletedDate
\n><Name\n>UnScheduleableVisit(1)</Name\n><ListOfVisitPlanAction\n><VisitPlanAction\n><ClinicalItem
\n>UnScheduleableAct02</ClinicalItem\n><Done\n>1/10/2023</Done\n><FormId\n>UnScheduleableAct02</
FormId\n><RepeatingNumber\n>0</RepeatingNumber\n></VisitPlanAction\n><VisitPlanAction\n><ClinicalItem
\n>UnScheduleableAct02</ClinicalItem\n><Done\n>1/11/2023</Done\n><FormId\n>UnScheduleableAct02</FormId
\n><RepeatingNumber\n>1</RepeatingNumber\n></VisitPlanAction\n></ListOfVisitPlanAction\n></VisitPlan
\n></ListOfVisitPlan\n></ClinicalSubject\n></ListOfClinicalSubjectInternal\n></SiebelMessage\n>",
 "Error Code": "",
 "Siebel Operation Object Id": "",
 "Process Instance Id": "88-34YFKD",
 "Integration Id": "Int_Test_SVT_02",
 "Error Message": ""
}
```


Undo Completed Subject Visit Activities

This API is also used to mark a subject visit activity as Incomplete (where the activity was previously completed) if there is No Payment Associated to it. If the activity has payment associated to it, it will not be marked as Incomplete.

The following request will undo Completed Activities in Siebel Clinical:

- **URI: https://ServerName:port/siebel/v1.0/workflow/LS Clinical Subject Inbound-Subject**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "OutputIntObjectName": {
 "MessageId": "",
 "MessageType": "Integration Object",
 "IntObjectName": "LS Clinical Subject Internal",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfLS Clinical Subject Internal": {
 "Clinical Subject": [
 {
 "Protocol Number": "PT_Protocol_001",
 "Site Number": "SVT_001",
 "Integration Id": "Int_SVT_001",
 "ListOfVisit Plan": {
 "Visit Plan": [
 {
 "Clinical Item": "Screening Visit",
 "Completed Date": "12/02/2022",
 "Name": "Screening Visit",
 "ListOfVisit Plan Action": {
 "Visit Plan Action": [
 {
 "Clinical Item": "Screening Visit - act2",
 "Done": "",
 "Form Id": "ScreeningAct01"
 },
 {
 "Clinical Item": "Screening Visit - act1",
 "Done": "",
 "Form Id": "ScreeningAct02"
 }
```

```
 ]
 }
 }
 ]
 }
 }
 ]
 }
 }
}
```

The above sample payload will undo the `"Screening Visit – act2", "Screening Visit – act1"` activities in the `"Screening Visit"` visit with Repeat Number=0, if there is no payment associated with these activities. If there is any payment amount associated with these activities, then no updates are done to these activities and an email is sent.

If the user wants to undo the activities of a dynamic visit, then the user should pass the visit name including the repeat number in parentheses.

Example Visit name: Screening Visit(1)

The user can also enable email notification in case of failed Undo of Activities, by following the instructions provided in the “Setting Up Integration Between CTMS and Oracle Clinical One” chapter of *Siebel Clinical Trial Management System Guide*.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "#value": "<?xml version=\"1.0\" encoding=\"UTF-8\"?><SiebelMessage\n MessageId=\"\"\n IntObjectName=
\"LS Clinical Subject Internal\"\n MessageType=\"Integration Object\"\n IntObjectFormat=\"Siebel
 Hierarchical\"\n><ListOfClinicalSubjectInternal\n><ClinicalSubject\n><IntegrationId\n>Int_Test_SVT_02</
IntegrationId\n><ProtocolNumber\n>PT_Protocol_001</ProtocolNumber\n><ProtocolSiteId\n>88-2WGGP</
ProtocolSiteId\n><SiteNumber\n>PT_Site_001</SiteNumber\n><ListOfVisitPlan\n><VisitPlan\n><ClinicalItem
\n>Screening Visit</ClinicalItem\n><CompletedDate\n>1/11/2023</CompletedDate\n><Name\n>Screening
 Visit</Name\n><ListOfVisitPlanAction\n><VisitPlanAction\n><ClinicalItem\n>Screening Visit – act02</
ClinicalItem\n><Done\n></Done\n><FormId\n>ScreeningAct01</FormId\n></VisitPlanAction\n><VisitPlanAction
\n><ClinicalItem\n>Screening Visit – act01</ClinicalItem\n><Done\n></Done\n><FormId\n>ScreeningAct02</
FormId\n></VisitPlanAction\n></ListOfVisitPlanAction\n></VisitPlan\n></ListOfVisitPlan\n></
ClinicalSubject\n></ListOfClinicalSubjectInternal\n></SiebelMessage\n>",
 "Error Code": "",
 "Siebel Operation Object Id": "",
 "Process Instance Id": "88-34YFLH",
 "Integration Id": "Int_Test_SVT_02",
 "Error Message": ""
}
```

10 Using Siebel REST API For Siebel Telco

Using Siebel REST API For Siebel Telco

This chapter describes Siebel Representational State Transfer (REST) Services for Siebel Telco users and how to use them. It includes the following topics:

- *Configuring Siebel Telco Business Services*
- *Using Siebel REST API with Siebel Telco*

More information is also available in *Siebel CRM Web Services Reference* and *Siebel Order Management Guide* .

Configuring Siebel Telco Business Services

Before you can invoke Siebel REST APIs for Siebel Telco, you use Siebel Application to add responsibilities to specific business services.

To add responsibilities to Siebel Telco business services

1. Log in as an administrator.
2. Navigate to the Administration - Application screen and then the Business Service Access view.
3. Add the following business services to the Business Service list:
    - ISS Promotion Definition WS Loader
    - Workflow Process Manager
    - PDS Order
    - PDS Quote
    - Dynamic Matrix Retrieval Service
    - SWI Product Attributes
    - SWI Product
    - SWI Promotion Import
    - SWI Product Import
    - SWI Product Class Import
    - SWI Attribute Import
    - SWI Product Line
    - ISS Utility Service
    - SWI External Integration Service
    - SWI Asset Management Promotion
    - SelfServiceAddress

- ISS Authoring Import Export Service
- ISS Web Commerce Export Attributes
- ISS Web Commerce Export Class
- Asset Management
- Cfg Object Broker
- SWI Product Class
- SWI Order Upsert
- PDS Product Data Service
- SWIPriceListItem
- SWI Order Update
- ISS Promotion Management Service
- SWI VOD Versions

> **Note:** For more information about associating a business service with a responsibility, see *Siebel Security Guide* .

4. For each business service, provide access to each required responsibility. For more information about adding responsibilities, see *Configuring Business Service Methods for REST Access*.
5. (Optional) Add other custom responsibilities that allow these business services to be accessed from the Siebel UI.
6. Click Clear Cache in both the Business Service and Responsibilities applets.

Using Siebel REST API with Siebel Telco

This topic provides a list of REST APIs for self-service commerce using Siebel Customer Order Management. You can use these REST APIs to build shopping cart scenarios using product catalog and checkout flow. In this list of REST APIs are APIs that support modifying, resuming, and disconnecting services, along with upgrading and downgrading bundles. These are an integral part of Oracle's Digital Experience for Communications offerings.

More information is also available in *Siebel CRM Web Services Reference* and *Siebel Order Management Guide* .

This topic includes the following:

- *Applying a Promotion to a Quote or an Order*
- *Adding an Item to a Promotion*
- *Replacing an Item in a Promotion*
- *Deleting an Item in a Promotion*
- *Disconnecting a Promotion*
- *Resuming a Promotion*
- *Returning a List of Upgrade or Downgrade Paths for a Promotion*
- *Upgrading or Downgrading a Promotion*
- *Returning a Promotion Definition*

- *Returning a List of Eligible Products in a Promotion*
- *Refreshing the Cache*
- *Performing a Paginated Product Search*
- *Performing an Order Checkout*

Applying a Promotion to a Quote or an Order

When you use the Siebel REST API to apply a promotion to a quote or an order, the Siebel application takes the bundle promotion definition ID and adds an instance of the promotion, along with its default components, to a quote or to an order. We can also use this API to associate an existing line item to an existing promotion instance.

The following request applies a specific promotion to a quote or an order:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body": {
 "ProcessName": "ISS Promotion WS - ApplyProductPromotion - Order",
 "ProdPromId": "88-255DX",
 "PricingMode": "Y",
 "EligibilityMode": "1",
 "SiebelMessage": {
 "MessageId": "",
 "MessageType": "Integration Object",
 "IntObjectName": "PDS Order",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfPDS Order": {
 "Header": {
 "Account": "R_Account",
 "Account Id": "88-1V9XIA",
 "Order Type": "Sales Order",
 "Order Number": "88-1X9C57"
 }
 }
 }
 }
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	One of the following processes is required: • ISS Promotion WS - ApplyProductPromotion - Order • ISS Promotion WS - ApplyProductPromotion (This is the process name for Quote.)
ProdPromId	Required. Promotion ID that is applied to the quote or order.

Name	Description
PricingMode	Optional. The default value is Y. If the value set to Y, then the pricing is calculated for all line items in the in-memory document.
EligibilityMode	Optional. Valid values are 0, 1, or 2. • If set to 0, then the procedure is not run. • If set to 1, then the procedure is run, and all ineligible products are displayed with messages. • If set to 2, then the procedure is run, and ineligible products are not included in the pricing calculations. **Note:** This behavior is different from how other APIs behave where EligibilityMode is set to 2.
Quantity	Optional. The default value is 1. Quantity of the promotion to be applied.
HeaderId	Optional, depending on usage. However, this is required if SiebelMessage is not provided. Quote ID or Order ID.
Sync	Optional. The default value is N. If set to Y, then an order or a quote is created or updated in the database.
ProdPromInstanceId	Optional. Integration ID of the Promotion. If the promotion already exists in the document, and one more product needs to be added to the promotion instance, then this ID is passed.
LineItemId	Optional. Line item ID of the product to which the promotion should be applied.
SiebelMessage	Optional, depending on usage. However, this is required if HeaderId is not provided. PDS quote or PDS Order.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
"Error Code": "",
"Non Orderable Error": "",
"Error Message": "",
"SiebelMessage": {
```

```
"IntObjectName": "PDS Order",
"MessageType": "Integration Object",
"Header": {
"Order Number": "88-1X9C57",
"Id": "88-1VAFOS",
"Account Id": "88-1V9XIA",
"Account": "R_Account",
"Order Type": "Sales Order",
"Line Item": [
{
"Asset Integration Id": "88-1VAFOQ",
"Name": "Millenium Max 1000 Package",
"Product Id": "88-255DX",
"Product Structure Type": "None"
},
{
"Asset Integration Id": "88-1VAFOW",
"Name": "Wireless Service",
"Product Id": "88-231H5",
"Product Prom Id": "88-255DX",
"Product Structure Type": "Customizable"
}
]
}
}
}
```

Adding an Item to a Promotion

When you use the Siebel REST API to add an optional component to a promotion, the Siebel application adds a specific product to an existing promotion either in a quote or in an order. This promotion could be the one you created in *Applying a Promotion to a Quote or an Order*, or, it could be an existing promotion that is being modified. This API is primarily used to add optional components, and throws an error message if the newly added item violates the maximum cardinality. An optional product can be either a promotion component, or an aggregate relationship specified using ProdPromRuleId.

The following request adds an optional item to a promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body for order:**

```
{
 "body": {
 "ProcessName": "ISS Edit Promotion WS - Add Item Process",
 "ProdPromId": "88-1W9L6Y",
 "PricingMode": "Y",
 "EligibilityMode": "1",
 "ProdPromInstanceId": "88-1W9MMH",
 "ProdPromRuleId": "88-1W9L7B",
 "ProductId": "88-1W9L4O",
 "External Integration Id": "12345678",
 "Quantity": "1",
 "Target Document": "Quote",
 "Header Id": "88-1W9MM8"
 }
```

```
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	The following process is required: • ISS Edit Promotion WS - Add Item Process
ProdPromId	Required. Promotion ID that is already applied to the quote or order. A new item is added to this promotion.
PricingMode	Optional. The default value is Y. If the value is set to Y, then the pricing is calculated for all the line items in the in-memory document.
EligibilityMode	Optional. Valid values are 0, 1, or 2. The default value is 2. • If set to 0, then the procedure is not run. • If set to 1, then the procedure is run, and all ineligible products are displayed with messages. • If set to 2, then the procedure is run, and ineligible products are not included in the pricing calculations. **Note:** This behavior is different from how other APIs behave where EligibilityMode is set to 2.
ProdPromInstanceId	Required. Asset integration ID of the promotion.
ProdPromRuleId	Required. Product promotion rule ID.
SiebelMessage	Optional, depending on usage. However, this is required if HeaderId is not provided. PDS quote or PDS order.
ProductId	Required. Product ID that needs to be added to the promotion instance.
External Integration Id	Not used.
Quantity	Required. The default value is 1. Quantity of the product that needs to be added and associated with the promotion.

Name	Description
Header Id	Optional, depending on usage. However, this is required if SiebelMessage is not provided. Quote ID or order ID.
Sync	Optional. The default value is N. If set to Y, then an order or a quote is created or updated in the database.
Target Document	Required. The default value is Quote. Quote or Order.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
"Error Code": "",
"Error Message": "",
"SiebelMessage": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Order",
"MessageType": "Integration Object",
"Header": {
"Account": "EPUI_Account",
"Order Number": "EPUI_Modify_Order",
"Order Type": "Sales Order",
"Line Item": [
{
"Name": "EPUI_CP1",
"Product Id": "88-1W9L4O",
"Line Item": {
"Name": "EPUI_SP3"
}
}
]
}
}
}
```

Replacing an Item in a Promotion

When you use the Siebel REST API to replace an item in a promotion, the Siebel application replaces an old component of an existing promotion with another eligible component from the same product line or product class. This promotion could be the one you created in *Applying a Promotion to a Quote or an Order*, or, it could be an existing promotion that is being modified. Because mandatory components cannot be deleted, this API is used to replace a mandatory component using just one operation, instead of using the DELETE and ADD operations.

The following request replaces an item in a promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"ProcessName": "ISS Edit Promotion WS - Replace Item Process",
"PricingMode": "Y",
"EligibilityMode": "1",
"Siebel Operation Object Id": "88-1WMCXR",
"New Product Id": "88-231F2",
"Target Document": "Quote",
"Header Id": "88-1WMCWZ",
"Sync": "N"
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	The following process is required: • ISS Edit Promotion WS - Replace Item Process
Header Id	Optional, depending on usage. However, this is required if SiebelMessage is not provided. Quote ID or Order ID.
Siebel Operation Object Id	Required. Row ID of either the quote item or order item that needs to be replaced.
Target Document	Required. The default value is Quote. Quote or Order.
SiebelMessage	Optional, depending on usage. However, this is required if HeaderId is not provided. PDS quote or PDS order.
Pricing Mode	Optional. The default value is Y. If set to Y, then pricing is calculated for all the line items in the in-memory document.
Eligibility Mode	Optional. Valid values are 0, 1, or 2. The default value is 2. • If set to 0, then the procedure is not run. • If set to 1, then the procedure is run, and all ineligible products are displayed with messages.

Name	Description
	• If set to 2, then the procedure is run, and ineligible products are not included in the pricing calculations. **Note:** This behavior is different from how other APIs behave where EligibilityMode is set to 2.
New Product Id	Required. ID of new product that replaces the previous product.
Sync	Optional. The default value is N. If set to Y, then the order or quote is created or updated in the database.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"Error Code": "",
"Siebel Operation Object Id": "88-1WMCXR",
"Process Instance Id": "88-1WMAA3",
"PricingMode": "Y",
"EligibilityMode": "1",
"Error Message": "",
"Object Id": "",
"SiebelMessage": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Quote",
"MessageType": "Integration Object",
"Header": {
"Quote Type": "Quotation",
"Id": "88-1WMCWZ",
"Name": "NEW_QUOTE",
"Quote Number": "88-1WMCWZ",
"Line Item": [
{
"Product Structure Type": "None",
"Product Type Code": "Promotion",
"Product Id": "88-231M5",
"Name": "Millennium Max 200 Package",
"Action Code": "Add",
"Id": "88-1WMCX8"
},
{
"Id": "88-1WMAA8",
"Name": "Daytech 300",
"Product Id": "88-231F2",
"Product Structure Type": "Customizable",
"Action Code": "Add",
"Prod Prom Id": "88-231M5",
"Product Type Code": "Product",
"Line Item": [
{
```

```
"Id": "88-1WMAAA",
"Product Id": "88-231DT",
"Action Code": "Add",
"Product Type Code": "Product"
}
]
}
]
}
}
}
```

Deleting an Item in a Promotion

When you use the Siebel REST API to delete an item in a promotion, the Siebel application deletes a product from an existing promotion instance either in a quote or in an order. This promotion could be the one you created in *Applying a Promotion to a Quote or an Order*, or, it could be an existing promotion that is being modified. This API is primarily used to remove optional components, and throws an error message when the deleted item violates the minimum cardinality. If a component level commitment is defined, then deleting the component adds penalty charges to the response.

The following request deletes an item in a promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"ProcessName": "ISS Edit Promotion WS - Delete Item Process",
"Header Id": "88-1W2C59",
"Siebel Operation Object Id": "88-1W2C60",
"Target Document": "Order",
"CheckPricingEligibilityFlag": "Y",
"Sync": "N"
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	The following process is required: • ISS Edit Promotion WS - Delete Item Process
Header Id	Optional, depending on usage. However, this is required if SiebelMessage is not provided. Quote ID or Order ID
Siebel Operation Object Id	Required. Row ID of the quote item or order item that is being deleted.

Name	Description
Target Document	Required. Quote or Order.
SiebelMessage	Optional, depending on usage. However, this is required if HeaderId is not provided. PDS quote or PDS order.
Sync	Optional. The default value is N. If set to Y, then the order or quote is created or updated in the database.
CheckPricingEligibilityFlag	Optional. The default value is N. If set to Y, then the Pricing and Eligibility procedure is executed with EligibilityMode set to default value of 1.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"Error Code": "",
"Siebel Operation Object Id": "88-1W1BSC",
"Process Instance Id": "88-1W10NP",
"Error Message": "",
"Object Id": "",
"SiebelMessage": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Quote",
"MessageType": "Integration Object",
"Header": {
"Id": "88-1W2C59",
"Line Item": [
{
"Asset Integration Id": "88-1W2BYU",
"Action Code": "Delete",
"Product Type Code": "Product",
"Prod Prom Name": "Millennium Max 500 Package",
"Id": "88-1W2C60",
"Prod Prom Id": "88-231OB",
"Name": "Daytech 300",
"Product Id": "88-231F2",
"Line Item": [
{
"Product Type Code": "Product",
"Id": "88-1W2C61",
"Name": "Charger",
"Product Id": "3SIA-2MOYY"
}
]
},
{
```

```
"Product Type Code": "Promotion",
"Id": "88-1W2C6O",
"Name": "Millennium Max 500 Package",
"Product Id": "88-231OB"
}
]
}
}
}
```

Disconnecting a Promotion

When you use the Siebel REST API to cancel a promotion, the Siebel application disconnects an existing promotion asset. If the promotion commitment is breached, then a penalty line item is created during this process. Any components that can exist outside of the promotion either are disconnected, or are retained based on the value of Component Prompt Action input.

The following request disconnects a promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"ProcessName": "SISOMDisconnectWebService-Order",
"Due Date": "9/17/2020",
"AssetIntegrationId": "88-1WNT2N",
"AccountId": "88-1WNT23",
"AssetPromptAction": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "ISS Promotion Component Prompt Action",
"MessageType": "Integration Object",
"ListOfISS Promotion Component Prompt Action": {
"Asset": [
{
"Asset Integration Id": "88-1WNT2S",
"Component Prompt Action": "Disconnect"
},
{
"Asset Integration Id": "88-1WNT2W",
"Component Prompt Action": "Disconnect"
}
]
}
}
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	One of the following processes is required: • SISOMDisconnectWebService-Order • SISOMDisconnectWebService-Quote
Due Date	Optional. The default value is today's date, plus one day. Disconnection date of the promotion.
AssetIntegrationId	Required. Integration ID of the bundle promotion asset you disconnect.
AccountId	Required. Account ID associated with the asset.
EligibilityMode	Optional. Valid values are 0, 1, or 2. • If set to 0, then the procedure is not run. • If set to 1, then the procedure is run, and all ineligible products are displayed with messages. • If set to 2, then the procedure is run, and all ineligible products are not included in selections.
AssetPromptAction:ListOfISS Promotion Component Prompt Action:Asset:AssetIntegrationId	Required for prompt components. Integration ID of promotion component with the Prompt Action code.
AssetPromptAction:ListOfISS Promotion Component Prompt Action:Asset:Component Prompt Action	Required for prompt components. Disconnect or Modify. If the value is Disconnect, then the asset is disconnected. If the value is Modify, then the asset is continued.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"ActiveDocumentId": "88-1WO67Z",
"Error Code": "",
"Error Message": "",
"SiebelMessage": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-26HTT",
"IntObjectName": "PDS Order",
"MessageType": "Integration Object",
"Header": {
"Id": "88-1WO67Z",
"Line Item": [
{
"Product Structure Type": "None",
"Asset Status": "Active",
```

```
"Name": "ND_SP2",
"Asset Integration Id": "88-1WNT2W",
"Action Code": "Delete",
"Id": "88-1WO69J"
},
{
"Product Structure Type": "None",
"Asset Status": "Active",
"Name": "ND_SP1",
"Asset Integration Id": "88-1WNT2S",
"Action Code": "Delete",
"Id": "88-1WO69K"
}
]
}
}
}
```

Resuming a Promotion

When you use the Siebel REST API to resume a promotion, the Siebel application resumes an existing promotion asset that had previously been suspended. All promotion components are resumed in addition to the promotion.

The following request resumes a promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body for Order:**

```
{
"body": {
"ProcessName": "SISOMResumeWebService-Quote",
"AccountId": "88-1VG0YB",
"AssetIntegrationId": "88-1VG0YU"
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	One of the following processes is required: • SISOMResumeWebService-Quote • SISOMResumeWebService-Order
Asset Integration Id	Required. Integration ID of the bundle promotion asset you want to resume.
Account Id	Required. Account ID associated with the asset.

Name	Description

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
"ActiveDocumentId": "88-1VI735",
"Error Code": "",
"Error Message": "",
"SiebelMessage": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-22F13",
"IntObjectName": "PDS Quote",
"MessageType": "Integration Object",
"Header": {
"Status": "In Progress",
"Id": "88-1VI735",
"Name": "88-1VI735",
"Quote Number": "88-1VI735",
"Line Item": [
{
"Product Structure Type": "None",
"Product Type Code": "Promotion",
"Name": "Millennium Max 500 Package",
"Asset Integration Id": "88-1VG0YU",
"Action Code": "Resume",
"Id": "88-1VI73U"
}
]
}
}
}
```

Returning a List of Upgrade or Downgrade Paths for a Promotion

A bundle promotion asset can be eligible for an upgrade or a downgrade to multiple target bundle promotions. This REST API allows you to fetch a list of those target promotions. When you use the Siebel REST API to fetch a list of upgrade or downgrade paths for a specific promotion, the Siebel application returns a list of possible upgrade and downgrade options for a specific promotion asset, as well as the eligibility, commitments and penalties associated with each option.

The following request returns a list of upgrades or downgrades for a specific promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/ISS Promotion Management Service/GetPromotionUpgradePaths**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
```

```
"Mode": "Account",
"Header Id": "88-1VG35J",
"Asset Integration Id": "88-1VG364"
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Mode	Required. Account
Header Id	Required. Account ID associated with the asset.
AssetIntegrationId	Required. Integration ID of the promotion asset you want to upgrade.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
"SiebelMessage": {
"PARSEIONAME": "GetPromotionUpgradePaths",
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"Integration Object": "GetPromotionUpgradePaths",
"MessageType": "Integration Object",
"ISS Promotion Upgrade Aggregate VBC": [
{
"Asset Integration Id": "88-1VFWUR",
"Prorate Plan Name": "10% for first 10 months",
"Original Promotion Id": "88-3N951",
"Target Promotion Name": "500SRS_Tgt_Prom2",
"Target Promotion Id": "88-3N95K",
"Operation Id": "88-3N77D",
},
{
"Asset Integration Id": "88-1VFWUR",
"Original Promotion Id": "88-3N951",
"Target Promotion Name": "500SRS_Tgt_Prom3",
"Target Promotion Id": "88-3N95T",
"Operation Id": "88-3M83F",
}
]
}
}
```

Upgrading or Downgrading a Promotion

When you use the Siebel REST API to upgrade or downgrade a promotion, the Siebel application upgrades or downgrades an existing promotion asset to the selected target promotion. If the existing promotion's components are part of the target promotion definition, then they are transferred to the target promotion. The other mandatory components of the target promotion are then added. Any components in the existing promotion asset that cannot be moved to the target promotion are either disconnected or retained, depending on the value of the Component Prompt Action in the input. The result of this API is a Quote or an Order.

The following request upgrades or downgrades a promotion asset:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
 {
 "body": {
 "ProcessName": "ISS Promotion WS - In Memory Upgrade Process-Order",
 "Account Id": "88-1VYRV7",
 "Asset Integration Id": "188-1VYRW1",
 "Operation Id": "88-231PR",
 "AssetPromptAction": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "1234",
 "IntObjectName": "ISS Promotion Component Prompt Action",
 "MessageType": "Integration Object",
 "ListOfISS Promotion Component Prompt Action": {
 "Asset": {
 "Asset Integration Id": "88-1VYRWA",
 "Component Prompt Action": "Disconnect"
 }
 }
 }
 }
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	One of the following processes is required: • ISS Promotion WS - In Memory Upgrade Process-Order • ISS Promotion WS - In Memory Upgrade Process-Quote
Asset Integration Id	Required. Integration ID of the promotion asset you want to upgrade.
Account Id	Required. Account ID associated with the asset.

Name	Description
Operation Id	Required. Unique ID for upgrade path, this is the returned Operation Id from GetPromotionUpgradePaths REST API.
New Promotion Id	Required. Row ID of the target promotion.
AssetPromptAction:Asset:Asset Integration Id	Required for prompt components. Integration ID of the Promotion component with the Prompt Action code.
AssetPromptAction:Asset:Component Prompt Action	Required for prompt components. Disconnect or Modify. If the value is Disconnect, then the asset is disconnected. If the value is Modify, then the asset is continued.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "Error Code": "",
 "Siebel Operation Object Id": "88-231PR",
 "Account Id": "88-1VYRV7",
 "Asset Integration Id": "88-1VYRW1",
 "New Promotion Id": "88-231OB",
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "88-24PPB",
 "IntObjectName": "PDS Order",
 "MessageType": "Integration Object",
 "Header": {
 "Account Id": "88-1VYRV7",
 "Id": "88-1W2KLG",
 "Order Number": "296-114333809",
 "Order Type": "Sales Order",
 "Line Item": [
 {
 "Asset Integration Id": "88-1VYRWA",
 "Action Code": "Update",
 "Product Type Code": "Product",
 "Id": "88-1W2KLJ",
 "Name": "ND_SP1",
 "XA": [
 {
 "Display Name": "Color",
 "XA Id": "88-1V6BGF",
 "Id": "88-1W2KLK",
 "Action Code": "-"
 }
 ]
```

```
},
{
"Asset Integration Id": "88-1VYRW1",
"Action Code": "Delete",
"Product Type Code": "Promotion",
"Id": "88-1W2KLZ",
"Name": "Millennium Max 200 Package"
}
]
}
}
}
```

Returning a Promotion Definition

When you use the Siebel REST API to return the entire structure of a specific promotion, the Siebel application returns the entire promotion definition or structure, including relationship and component hierarchy details, the pricing for components and aggregate, and the promotion commitment terms for the given promotion. This REST API returns the promotion definition defined using promotion designer in the Product Promotions view in Administration - Product.

The following request returns the entire structure of a specific promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"ProcessName": "ISS Promotion WS - GetProductPromotionDetails",
"ProdPromId": "88-3N1ZN",
"LoadPromDefFlag": "Y",
"LoadStructFlag": "Full",
"LoadAllDomainFlag": "Y",
"LoadDefaultDomainFlag": "N",
"ProdPromRuleType": "Product"
"Context": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Catalog Context",
"MessageType": "Integration Object",
"ListOfPDS Catalog Context": {
"Context": {}
}
}
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	The following process is required: • ISS Promotion WS - GetProductPromotionDetails
ProdPromId	Required. Row ID of the promotion.
LoadPromDefFlag	Optional. The default value is Y. Returns the promotion definition structure.
LoadStructFlag	Optional. There is no default value. Valid values are Full, Modified or no value. • If set to Full, then it returns all the promotion rules. • If set to Modified, then it returns all the promotion rules that are different from the product definition. • When no value is specified and is empty, and ProdPromRuleId is passed, then it only returns the rule with the ID that is passed.
LoadAllDomainFlag	Optional. The default value is N. If the value is Y, then it returns the structure of the underlying product.
LoadDefaultDomainFlag	Optional. The default value is Y. Returns the default product under the promotion aggregate relationship.
ProdPromRuleType	Optional. The default value is Product. Valid values are Product and Pricing.
ProdPromRuleId	Required when no value is set for LoadStructFlag field, or when the LoadDefaultDomainFlag value is Y.
Context	Optional. However required if LoadDefaultDomainFlag is Y. Filters the promotion aggregate relationship based on the ProdPromRuleId and the ProductSearchSpec in the context.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"Error Code": "",
"Error Message": "",
"DomainProductDetail": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-23C8R",
"IntObjectName": "SWI Admin ISS Product Definition",
```

```
"MessageType": "Integration Object",
"SWI Internal Product VOD": [
{
"Product Type Code": "Product",
"Product Def Type Code": "Customizable",
"Name": "ACR722GetPromoWS_CP1",
"Price Type": "One-Time",
"Integration Id": "88-3MERG",
"Id": "88-3MERG",
"SWI ISS Product VOD BusComp": {
"VOD Id": "88-3MERG",
"VOD Name": "ACR722GetPromoWS_CP1",
"VOD Type": "ISS_PROD_DEF",
"SWI ISS Product Details BusComp": {
"Orig Id": "88-3MERO",
"Object Id": "88-3MERM",
"Pre Pick CD": "",
"Parametric Search Flg": "",
"Last Version": "0"
},
"SWI ISS Product Structure Admin BusComp": {
"Prod Id": "88-3MEQB",
"Object Id": "88-3MERM",
"Item Name": "SP1",
"Type": "Product",
"Class Name": "",
"Orig Id": "88-3MERQ",
"Product Name": "ACR722GetPromoWS_SP1",
"Sequence Num": ""
},
"SWI ISS Product Versions BusComp": {
"Version": "2",
"VOD Id": "88-3MERM",
"Version Id": "88-3MERN"
}
},
"SWI Internal Product Organization": {
"Product Id": "88-3MERG",
"Organization Id": "0-R9NH"
}
},
{
"Parent Internal Product Name": "",
"Image Primary Organization Name": "",
"Product Type Code": "Product",
"Product Def Type Code": "Customizable",
"VOD Id": "88-3MESC",
"Special Rating Max Items": "",
"Sequence Number": "",
"Part #": "",
"Fulfillment Criteria Code": "Default",
"Auto Allocate Flag": "Y",
"Name": "ACR722GetPromoWS_CP2",
"SWI ISS Product VOD BusComp": {
"VOD Id": "88-3MES6",
"VOD Name": "ACR722GetPromoWS_CP2",
"VOD Type": "ISS_PROD_DEF",
"SWI ISS Product Details BusComp": {
"Parent VOD Id": "",
"Orig Id": "88-3MESE",
"SPN Definition Id": "",
"Object Id": "88-3MESC"
},
"SWI ISS Product Structure Admin BusComp": [
{
"Prod Id": "88-3MEQW",
```

```
"Object Id": "88-3MESC",
"Item Name": "SP2",
"Min Cardinality": "1",
"Type": "Product",
"Object Number": "88-3MEQW",
"Orig Id": "88-3MESI",
"Product Name": "ACR722GetPromoWS_SP2"
},
{
"Prod Id": "88-3MEQB",
"Max Cardinality": "1",
"Object Id": "88-3MESC",
"Item Name": "SP1",
"Type": "Product",
"Forcastable Flag": "Y",
"Orig Id": "88-3MESG",
"Product Name": "ACR722GetPromoWS_SP1"
}
],
"SWI ISS Product Versions BusComp": {
"VOD Id": "88-3MESC",
"Latest Released Flag": "N"
}
},
"SWI Internal Product Organization": {
"Product Id": "88-3MES6",
"Organization Id": "0-R9NH"
}
},
{
"Product Type Code": "Product",
"Product Def Type Code": "Simple",
"VOD Id": "88-3MER2",
"Name": "ACR722GetPromoWS_SP2",
"Primary Organization Id": "0-R9NH",
"Integration Id": "88-3MEQW",
"Id": "88-3MEQW",
"Dynamic Discount Method": "Amount",
"SWI ISS Product VOD BusComp": {
"VOD Id": "88-3MEQW",
"VOD Name": "ACR722GetPromoWS_SP2",
"SWI ISS Product Details BusComp": {
"Orig Id": "88-3MER4",
"Object Name": "0",
"Object Id": "88-3MER2"
},
"SWI ISS Product Versions BusComp": {
"Version": "2",
"VOD Id": "88-3MER2",
"Version Id": "88-3MER3"
}
},
"SWI Internal Product Organization": {
"Product Id": "88-3MEQW",
"Organization Id": "0-R9NH"
}
}
]
},
"ProdPromDefinition": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-23C8Q",
"IntObjectName": "ISS Promotion",
"MessageType": "Integration Object",
"ISS Promotion": {
"Configuration Model Id": "88-3N1ZN",
```

```
"Name": "ACR722GetProm3",
"Promotion Type": "Bundled Promotions",
"Promotion Id": "88-3N1ZN",
"ISS Promotion Items for Import": [
{
"XA Modified Flag": "N",
"Prod Id": "88-3MES6",
"Prod Cfg Model Id": "88-3MES6",
"Integration Id": "88-3N1ZW",
"Promotion Id": "88-3N1ZN",
"Maximum Quantity": "1",
"Default Quantity": "1",
"Product Name": "ACR722GetPromoWS_CP2"
},
{
"Prod Id": "88-3MEQW",
"Prod Cfg Model Id": "88-3MEQW",
"Integration Id": "88-3N200",
"Promotion Id": "88-3N1ZN",
"Disconnect Action": "Modify",
"Grace Period UOM": "Days",
"Product Name": "ACR722GetPromoWS_SP2"
},
{
"Prod Id": "88-3MERG",
"Integration Id": "88-3N202",
"Maximum Quantity": "1",
"Default Quantity": "1",
"Product Name": "ACR722GetPromoWS_CP1"
}
],
"ISS Promotion Pricing Rules Summary": [
{
"Adjustment Type": "% Discount",
"Prod Cfg Model Id": "88-3MES6",
"Matrix Rule Num": "296-6114527",
"Promotion Id": "88-3N1ZN",
"Product Id": "88-3MES6",
"Product Name": "ACR722GetPromoWS_CP2"
}
],
"ISS Promotion Pricing Components": [
{
"Matrix Rule Num": "296-6114527",
"Product Id": "88-3MES6",
"Product Name": "ACR722GetPromoWS_CP2",
"Product Promotions Relationship VBC": [
{
"Class Name": "",
"Min Cardinality": "",
"Parent Product Name": "ACR722GetPromoWS_CP2",
"Promotion Id": "88-3N1ZN",
"Product Name": "ACR722GetPromoWS_SP1",
"Prod Id": "88-3MEQB"
}
]
},
{
"Promotion Id": "88-3N1ZN",
"Product Id": "88-3MEQW",
"Product Name": "ACR722GetPromoWS_SP2"
},
{
"Promotion Id": "88-3N1ZN",
"Product Id": "88-3MERG",
"Product Name": "ACR722GetPromoWS_CP1",
```

```
"Product Promotions Relationship VBC": {
"Parent Product Name": "ACR722GetPromoWS_CP1",
"Parent Product Id": "88-3MERG",
"Promotion Id": "88-3N1ZN",
"Relationship Name": "SP1",
"Product Name": "ACR722GetPromoWS_SP1",
"Prod Id": "88-3MEQB"
}
}
]
}
},
"PromDomainProduct": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Product Interface",
"MessageType": "Integration Object",
"Product": {}
}
}
```


Returning a List of Eligible Products in a Promotion

You use this Siebel REST API to check which products are eligible from a list of products before adding or replacing a component that belongs to a promotion aggregate rule, in an already existing promotion instance, in a quote, or in an order.

The following request returns a list of products that belong to a promotion aggregate rule, while checking eligibility on the list of products:

- **URI: http://ServerName:port/siebel/v1.0/service/ISS Promotion Management Service/GetEligibleComponents**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**


```
{
"body": {
"Rule Id": "88-1VTT1C",
"EligibilityMode": "1",
"Context": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-21EYX",
"IntObjectName": "PDS Catalog Context",
"MessageType": "Integration Object",
"ListOfPDS Catalog Context": {
"Context": {
"ListOfExternal Context": {
"External Context": {
"Account Id": "88-1VTT1D"
}
}
}
}
```

```
 }
 }
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Rule Id	Required. Promotion rule ID for which we need to get all the list of products that can be associated. An option to obtain the Rule Id is to use GetPromotionDetails REST API. The Integration Id of the Promotion Items for Import section is the Rule Id.
Eligibility Mode	Optional. Valid values are 0, 1, or 2. • If set to 0, then the eligibility procedure is not run. • If set to 1, then the eligibility procedure is run, and all ineligible products are displayed with messages. • If set to 2, then the eligibility procedure is run, and all ineligible products are not included in selections.
External Context	Optional. Specifies the context for running the eligibility procedure, for example passing standard eligibility criteria, such as Account Id, Account Type, and so on.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"SiebelMessage": {
"PARSEIONAME": "ISS Promotion Related Product Data",
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"Integration Object": "ISS Promotion Related Product Data",
"MessageType": "Integration Object",
"ISS Promotion Related Product Data": [
{
"Name": "HLine2Prod1",
"Id": "88-1VTSZV",
"Pre Pick": "Y",
"Eligibility Reason": "This product is not sold to: [Account] = HAccLine2",
"Class Id": "",
"Description": "",
"Product Line Id": "88-1VTSZU",
"Eligibility Status": "N"
},
{
"Name": "HLine2Prod2",
"Id": "88-1VTT0H",
"Pre Pick": "Y",
"Eligibility Reason": "",
"Class Id": "",
"Description": "",
```

```
 "Product Line Id": "88-1VTSZU",
 "Eligibility Status": "Y"
 }
 ]
 }
}
```

Refreshing the Cache

You use this Siebel REST API to refresh the cache for the product, versioned object key, list of value, and eligibility and compatibility matrices, and to invalidate outdated caches so that the latest versions are loaded and available for any subsequent calls.

The following request refreshes the cache for the product, versioned object key, list of value, and eligibility and compatbility matrices:

- **URI: http://ServerName:port/siebel/v1.0/service/SWI External Integration Service/RefreshCache**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
Input Payload…
{
 "body": {
 "Type": "LOV"
 }
}

Input Payload…
{
 "body": {
 "Type": "EligibilityCompatibility"
 }
}

Input Payload…
{
 "body": {
 "Type": "Product"
 }
}

Input Payload…
{
 "body": {
 "Type": "Versioned Object Key"
 }
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Type	Required. Type of cache that should be refreshed, such as list of values, eligibility, compatibility, product, versioned object key, and so on.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **No Response**

Performing a Paginated Product Search

You use this Siebel REST API to search products starting at a row number, and to return only a limited number of records that are specified in the input. This executes the search on products using the input search option, such as product class, class attributes, attribute constraints, pricing criteria, and so on. This search generates output in pages corresponding to the information requested in the input, which enables you to present information in a structured format.

Two types of search are supported: SearchByProduct and SearchByClass. SearchOption is a required input to indicate the type of search required.

The following request returns a list of products that belong to a promotion aggregate rule, while checking eligibility on the list of products:

- **URI: http://ServerName:port/siebel/v1.0/service/PDS Product Data Service/ExecuteSearch**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"EligibilityMode": "0",
"PricingMode": "N",
"PageSize": "4",
"RecordCountNeeded": "True",
"ContextMode": "SiebelContext",
"StartRowNum": "0",
"SearchOption": "SearchByClass",
"Context": {
"MessageId": "",
"IntObjectName": "PDS Catalog Context",
"MessageType": "Integration Object",
"ListOfPDS Catalog Context": {
"Context": {
"ListOfSiebel Context": {
"Siebel Context": {
"BusObjName": "Quote",
"ContextRowId": "0V-1706Z7",
"SignalMode": "Quote",
"BusCompName": "Quote"
}
```

```
}
}
}
},
"ClassAttributes": {
"MessageId": "",
"IntObjectName": "PDS Product Class Attributes Interface",
"MessageType": "Integration Object",
"ListOfPDS Product Class Attributes Interface": {
"ISS PS Direct Subclass": {
"Class Id": "88-1WVXT7",
"ListOfISS Class Attribute": {
"ISS Class Attribute": {
"Display Name": "Attr1",
"Value": "0"
}
}
}
}
}
}
}
```


For a description of the request message associated with this API, see the following table.

Name	Description
SearchOption	One of the following input arguments is required: • SearchByProduct. This is an instance of the integration object PDS Product Interface. • SearchByClass. This is an instance of the integration object PDS Product Class Attributes Interface. You must specify class ID, attribute ID, and attribute value in an attribute list of value structure. That is, an array with one element for either a freeform, or a list of value, type attribute.
PricingMode	Optional. Valid values are N or Y, and the default value is Y. This determines if a price calculation is made.
EligibilityMode	Optional. Valid values are 0, 1, or 2. • If set to 0, then the procedure is not run. • If set to 1, then the procedure is run, and all ineligible products are displayed with messages. • If set to 2, then the procedure is run, and ineligible products are not included in the pricing calculations.
ContextMode	Optional. However, if this is not provided then pricing and eligibility are not processed. Valid values are: • SiebelContext. Uses existing Quote and Order information from the Siebel database. The API then constructs the context using the provided row ID. • ExternalContext. The external call provides the context information. The API then passes this information to the pricing and eligibility processes.
Context	Required. However, this is optional if ContextMode is not provided.

Name	Description
	This is an instance of the integration object PDS Catalog Context. It either contains SiebelContext or ExternalContext.
PriceSearchExpression	Optional. The expression for performing a price search. For example, searching for a price that is greater than 100 and less than 1000.
ProductSearchSpec	Optional. This is the search specification for a product business component, for example: [Name] like "Oracle*"
ResultSize	Optional. The default value is 100. The number of products returned.
StartRowNum	Optional. The default value is 0. Starting row number.
PageSize	Optional. Number of records to be returned.
RecordCountNeeded	Optional. Valid values are TRUE or FALSE.
RecordCount	This is search output. If the value of RecordCountNeeded is TRUE, then this parameter returns the total number of records returned as defined by the search specification.
LastPage	This is search output. Indicates with the values TRUE or FALSE if the row set is the last.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{{
"LastPage": "False",
"RecordCount": "10",
"SiebelMessage": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Product Interface",
"MessageType": "Integration Object",
"Product": [
{
```

```
"Name": "TestPr1",
"Product Type Code": "Product"
},
{
"Name": "TestPr10",
"Product Type Code": "Product"
},
{
"Name": "TestPr2",
"Product Type Code": "Product"
},
{
"Name": "TestPr3",
"Product Type Code": "Product"
}
]
}
}
```

Performing an Order Checkout

When you use the Siebel REST API to checkout an order, the Siebel application calculates pricing, checks eligibility, and verifies the promotion on the given order structure. All operations are performed in memory. When the customer confirms the order and the SyncOrderFlag flag is set, then the order is created in Siebel CRM.

The following request performs an order checkout:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"ProcessName": "Web Channel Ordering Workflow",
"PricingFlag": "Y",
"CheckEligibilityFlag": "Y",
"SyncOrderFlag": "Y",
"VerifyPromotionFlag": "N",
"RepricingFlag": "Y",
"OrderIO": {
"MessageId": "",
"MessageType": "Integration Object",
"IntObjectName": "PDS Order",
"IntObjectFormat": "Siebel Hierarchical",
"ListOfPDS Order": {
"Header": {
"Account Id": "1-63Q9",
"Account": "3Com",
"Id": "0CX-1W1W9Y3",
"Price List Id": "88-23AQ7",
"Revision": "1",
"Account Type": "Residential",
"Order Number": "Test_OWS_sl3",
```

```
"Currency Code": "USD",
"Order Type": "Sales Order"
}
}
}
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	The following process is required: • Web Channel Ordering Workflow
OrderIO	Required. Order structure based on the PDS Order IO.
CheckEligibilityFlag	Optional. The default value is N. If the value is set to Y, then this flag triggers the eligibility and compatibility check sub-process.
PricingFlag	Optional. The default value is N. If the value is set to Y, then this flag triggers the Repricing step on the OrderIO.
SyncOrderFlag	Optional. The default value is N. If the value is set to Y, then this flag triggers the Synch Order operation.
VerifyPromotionFlag	Optional. The default value is N. If the value is set to Y, then this flag triggers the Verify the Order's Promotions step for an existing order on Siebel CRM.
QueryOrderFlag	Optional. The default value is N. If the value is set to Y, then this flag triggers another query to the Order step, which returns the updated OrderIO to the requesting application.
RepricingFlag	Optional. The default value is N. If the value is set to Y, then this flag triggers the Reprice step a second time after any updates have been made for an existing order in Siebel CRM.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
```

```
 "Error Code": "",
 "Process Instance Id": "88-1XBZA3",
 "Error Message": "",
 "OrderIO": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "",
 "IntObjectName": "PDS Order",
 "MessageType": "Integration Object",
 "Header": {
 "Mod Id": "0",
 "Discount": "",
 "Opty Id": "",
 "Conflict Id": "0",
 "Updated": "03/30/2021 12:11:35",
 "Order Number": "Test_OWS_sl3",
 "Revision": "1",
 "Integration Id": "88-1XBZAE",
 "Price List Id": "88-23AQ7",
 "Status": "Pending",
 "Account": "3Com",
 "Order Type": "Sales Order",
 "Id": "88-1XBZAE",
 "Created": "03/30/2021 12:11:33",
 "Contact Id": "1-6S9M",
 "Account Location": "HQ-Distribution",
 "Account Id": "1-63Q9",
 "Currency Code": "USD"
 }
 }
}
```

11 Using Siebel REST API For Siebel Product Administration

Using Siebel REST API For Siebel Product Administration

This chapter describes Siebel Representational State Transfer (REST) Services for Siebel Product Administration entities and how to use them. It includes the following topics:

- *Configuring Siebel Product Administration Business Services*
- *Using Siebel REST API with Product Administration*

More information is available in *Siebel CRM Web Services Reference* and *Siebel Order Management Guide* .

To view the Postman collection of REST APIs for Siebel Product Administration entities, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

Note: This Postman collection requires use of the third-party API platform, Postman™. The data contained in this zip file is provided for knowledge transfer and testing only. You are allowed to update this data as it exists in its current state. For supported versions of third-party software, see the Certifications tab on My Oracle Support.

Configuring Siebel Product Administration Business Services

Before you can invoke Siebel REST APIs for Siebel Product Administration, you use Siebel Application to add responsibilities to specific business services.

To add responsibilities to Siebel Product Administration business services

1. Log in as an administrator.
2. Navigate to the Administration - Application screen and then the Business Service Access view.
3. Add the following business services to the Business Service list:
    - SWI Catalog Admin
    - SWI Product
    - SWI Product Class
    - SWI Product Attribute
    - SWI Attribute Import

- SWI Product Class Import
- SWI Product Import
- SWI Smart Part Number
- SWI Price List Item
- SWI Product Line
- SWI Eligibility Compatibility Matrix
- SWI Product Base
- SWI Promotion Import
- SWI Cfg Object Broker
- Literature Web Service
- Workflow Process Manager
- ISS Promotion Definition Loader QP
- SWI VOD Versions

> **Note:** For more information about associating a business service with a responsibility, see *Siebel Security Guide* .

4. For each business service, provide access to each required responsibility. For more information about adding responsibilities, see *Configuring Business Service Methods for REST Access*.
5. (Optional) Add other custom responsibilities that allow these business services to be accessed from the Siebel UI.
6. Click Clear Cache in both the Business Service and Responsibilities applets.

Resetting State Attributes in Product Administration

Sometimes variables that store the state of invoked REST APIs, such as profile attributes and session global attributes, can affect the functionality of subsequent REST API calls.

To reset these profile attributes, you must define them in the SWI External Integration Service business service. This business service uses the ResetSessionCache method to accept a list of attributes as business service user properties, and then resets them. After you specify the attributes that you need to reset, you use the EAI REST Adapter Service business service to invoke the ResetSessionCache business method before or after specified REST API calls.

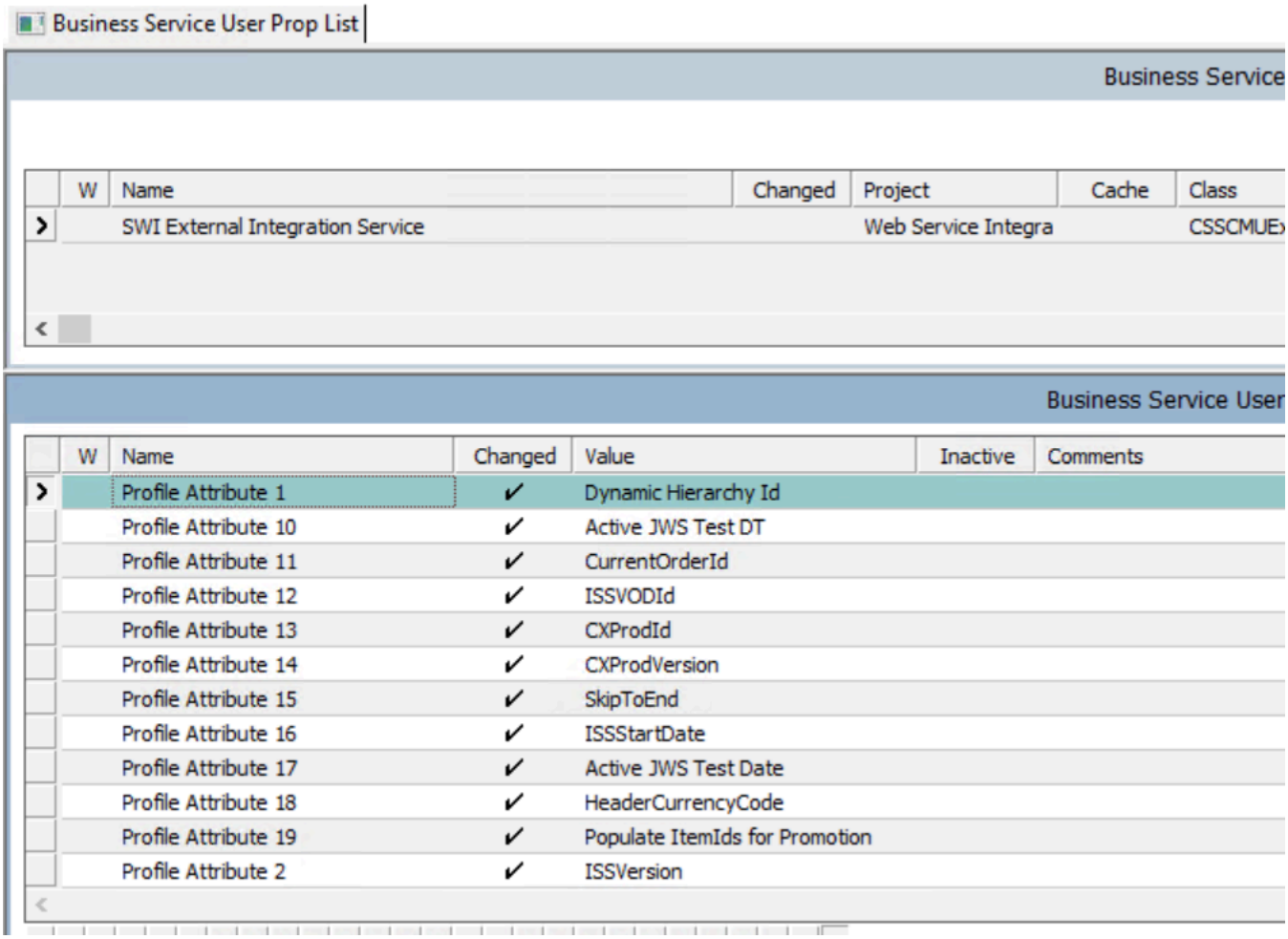
> **Note:** For more information about business service user properties, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

Before you can reset variables in Siebel REST API, you must enable preprocessing and postprocessing. For more information on the EAI REST Adapter Service, and enabling preprocessing and postprocessing, see *Configuring Preprocessing and Postprocessing Steps in Inbound REST API*.

To reset attribute values in product administration

1. Open the SWI External Integration Service business service, and define the attributes you want to reset.

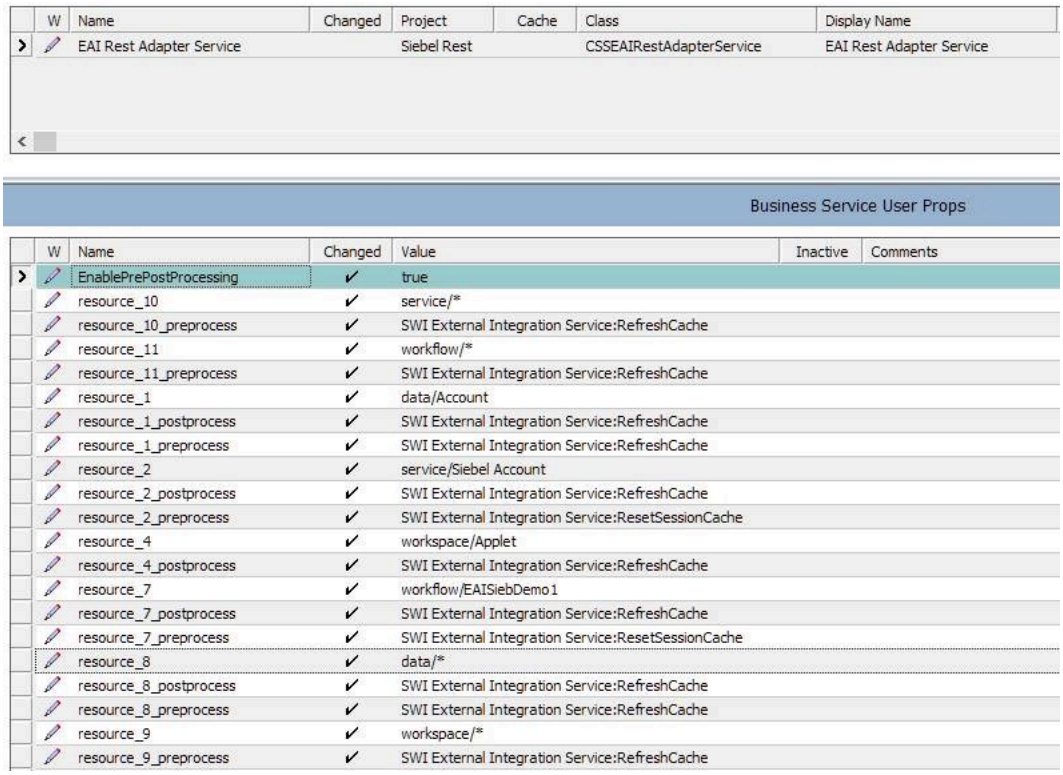
   By default, some attributes are already listed in the Business Service User Props pane. The following figure shows the Business Service User Props pane and some sample profile attributes and values:



2. Open the EAI REST Adapter Service business service, and map the URI or URI pattern, as well as the business service and business method that is responsible for each preprocessing and postprocessing function.
   The following table contains a sample URI and its corresponding user properties and values to configure preprocessing and postprocessing steps.

User Property Key	User Property Value
resource_1	data/*
resource_1_preprocess	Not applicable.
resource_1_postprocess	SWI External Integration Service:ResetSessionCache

The following figure shows the Business Service User Props pane and sample resource mapping user property keys and values:



W	Name	Changed	Project	Cache	Class	Display Name
	EAI Rest Adapter Service		Siebel Rest		CSSEAIRestAdapterService	EAI Rest Adapter Service

Business Service User Props

W	Name	Changed	Value	Inactive	Comments
	EnablePrePostProcessing	✓	true		
	resource_10	✓	service/*		
	resource_10_preprocess	✓	SWI External Integration Service:RefreshCache		
	resource_11	✓	workflow/*		
	resource_11_preprocess	✓	SWI External Integration Service:RefreshCache		
	resource_1	✓	data/Account		
	resource_1_postprocess	✓	SWI External Integration Service:RefreshCache		
	resource_1_preprocess	✓	SWI External Integration Service:RefreshCache		
	resource_2	✓	service/Siebel Account		
	resource_2_postprocess	✓	SWI External Integration Service:RefreshCache		
	resource_2_preprocess	✓	SWI External Integration Service:ResetSessionCache		
	resource_4	✓	workspace/Applet		
	resource_4_postprocess	✓	SWI External Integration Service:RefreshCache		
	resource_7	✓	workflow/EAISiebDemo1		
	resource_7_postprocess	✓	SWI External Integration Service:RefreshCache		
	resource_7_preprocess	✓	SWI External Integration Service:ResetSessionCache		
	resource_8	✓	data/*		
	resource_8_postprocess	✓	SWI External Integration Service:RefreshCache		
	resource_8_preprocess	✓	SWI External Integration Service:RefreshCache		
	resource_9	✓	workspace/*		
	resource_9_preprocess	✓	SWI External Integration Service:RefreshCache		

Using Siebel REST API with Product Administration

This topic provides a list of product administration REST APIs for Siebel Customer Order Management. You can use these REST API services to simplify omni-channel ordering with a centralized product catalog. The product administration REST APIs primarily include create, read, update and delete operations on catalog-related entities such as Catalog, Product Class, Product Definition, Product Promotion, and so on. This integration with product administration enables Oracle customers to create new product offerings faster, with a centrally-managed product catalog, and as a result reducing time-to-market.

More information is also available in *Siebel CRM Web Services Reference* and *Siebel Order Management Guide* .

This topic includes the following:

- *SWI Catalog Admin*
- *SWI Product*
- *SWI Product Class*
- *SWI Product Attribute*
- *SWI Attribute Import*
- *SWI Product Class Import*

- *SWI Product Import*
- *SWI Smart Part Number*
- *SWI Price List Item*
- *SWI Product Line*
- *SWI Eligibility Compatibility Matrix*
- *SWI Product Base*
- *SWI Promotion Import*
- *SWI Cfg Object Broker*
- *Literature Web Service*
- *Returning a Promotion Definition by ID*
- *Returning a Promotion Definition by Name*
- *ISS Promotion Definition Loader QP*

SWI Catalog Admin

Use this REST API service to create, update, and query catalogs, catalog translations, catalog categories and catalog category translations.

SWI Catalog Admin Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
InsertOrUpdate	Creates a new catalog or adds a category to existing catalog.
Update	Updates an existing catalog.
QueryPage	Queries for existing catalogs using the QueryPage method.
QueryById	Queries for an existing catalog using Row ID.
QueryByExample	Queries for existing catalogs using the QueryPage method.
Delete	Deletes a catalog.

Message Description: SWI Catalog Admin - InsertOrUpdate

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object

Node	Description	Type
Product Catalog	An integration component (XML tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under ListOfBase Catalog Admin to process different product catalogs at one time.	Integration Component
Name	Name of the catalog.	String
Catalog Type	Type of catalog.	String
Effective Start Date	Start date for a catalog.	String
Effective End Date	End date for a catalog.	String
ListOfCatalog Category Admin	List of categories within a catalog.	Integration Component
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Name	Category name.	String
ListOfInternal Product	Internal products associated with the catalog.	Integration Component
Name	Product name.	String
ListOfCatalog Category Translation	Translation for the catalog categories.	Integration Component
Language Code	Language code.	String
ListOfProduct Catalog Translation	Translation for the catalog.	Integration Component
Language Code	Language code.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object
Product Catalog	An integration component (XML tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under	Integration Component

Node	Description	Type
	ListOfBase Catalog Admin to process different product catalogs at one time.	
Name	Catalog name.	String
Catalog Type	Type of catalog.	String
Effective Start Date	Start date for a catalog.	String
Effective End Date	End date for a catalog.	String
ListOfCatalog Category Admin	List of categories within a catalog.	Integration Component
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Name	Category name.	String
ListOfInternal Product	Internal products associated with the catalog.	Integration Component
Name	Product name.	String
ListOfCatalog Category Translation	Translation for the catalog categories.	Integration Component
Language Code	Language code.	String
ListOfProduct Catalog Translation	Translation for the catalog.	Integration Component
Language Code	Language code.	String

Message Description: SWI Catalog Admin - Update

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object
Product Catalog	An integration component (XML tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under ListOfBase Catalog Admin to process different product catalogs at one time.	Integration Component

Node	Description	Type
Name	Catalog name.	String
Catalog Type	Type of catalog.	String
Effective Start Date	Start date for a catalog.	String
Effective End Date	End date for a catalog.	String
ListOfCatalog Category Admin	List of categories within a catalog.	Integration Component
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Name	Category name.	String
ListOfInternal Product	Internal products associated with the catalog.	Integration Component
Name	Product name.	String
ListOfCatalog Category Translation	Translation for the catalog categories.	Integration Component
Language Code	Language code.	String
ListOfProduct Catalog Translation	Translation for the catalog.	Integration Component
Language Code	Language code.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object
Product Catalog	An integration Component (XML Tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under ListOfBase Catalog Admin to process different product catalogs at one time.	Integration Component
Name	Catalog name.	String
Catalog Type	Type of catalog.	String

Node	Description	Type
Effective Start Date	Start date for a catalog.	String
Effective End Date	End date for a catalog.	String
ListOfCatalog Category Admin	List of categories within a catalog.	Integration Component
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Name	Category name.	String
ListOfInternal Product	Internal products associated with the catalog.	Integration Component
Name	Product name.	String
ListOfCatalog Category Translation	Translation for the catalog categories.	Integration Component
Language Code	Language code.	String
ListOfProduct Catalog Translation	Translation for the catalog.	Integration Component
Language Code	Language code.	String

Message Description: SWI Catalog Admin - QueryPage

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PageSize	Default is 10. Indicates the maximum number of integration object instances to be returned.	String
StartRowNum	Default is 0 (first page). Indicates the row in the result set for the QueryPage method to start retrieving a page of records.	String
ViewMode	Default is All. You must apply Visibility mode to the business object. Valid values are: • Manager • Sales Rep • Personal • Organization	String

Node	Description	Type
	• Sub-Organization • Group • Catalog • All **Note:** The ViewMode user property on the integration object has priority over the ViewMode method argument.	
SiebelMessage	The input or the output integration object instance.	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object
Product Catalog	An integration component (XML tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under ListOfBase Catalog Admin to process different product catalogs at one time.	Integration Component
Name	Catalog name.	String
Catalog Type	Type of catalog.	String
Effective Start Date	Start date for a catalog.	String
Effective End Date	End date for a catalog.	String
ListOfCatalog Category Admin	List of categories within a catalog.	Integration Component
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Name	Category name.	String
ListOfInternal Product	Internal products associated with the catalog.	Integration Component
Name	Product name.	String

Node	Description	Type
ListOfCatalog Category Translation	Translation for the catalog categories.	Integration Component
Language Code	Language code.	String
ListOfProduct Catalog Translation	Translation for the catalog.	Integration Component
Language Code	Language code.	String

Message Description: SWI Catalog Admin - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The row ID of the literature.	String
OutputIntObjectName	Name of the output integration object.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object
Product Catalog	An integration component (XML tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under ListOfBase Catalog Admin to process different product catalogs at one time.	Integration Component
Name	Catalog name.	String
Catalog Type	Type of catalog.	String
Effective Start Date	Start date for a catalog.	String
Effective End Date	End date for a catalog.	String
ListOfCatalog Category Admin	List of categories within a catalog.	Integration Component
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component

Node	Description	Type
Name	Name of category.	String
ListOfInternal Product	Internal products associated with the catalog.	Integration Component
Name	Product name.	String
ListOfCatalog Category Translation	Translation for the catalog categories.	Integration Component
Language Code	Language code.	String
ListOfProduct Catalog Translation	Translation for the catalog.	Integration Component
Language Code	Language code.	String

Message Description: SWI Catalog Admin - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object ListOfBase Catalog Admin.	Integration Object
Product Catalog	Integration component corresponding to the Product Catalog business component.	Integration Component
Name	Name of the catalog to be queried.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object
Product Catalog	An integration component (XML tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under ListOfBase Catalog Admin to process different product catalogs at one time.	Integration Component
Name	Catalog name.	String

Node	Description	Type
Catalog Type	Type of catalog.	String
Effective Start Date	Start date for a catalog.	String
Effective End Date	End date for a catalog.	String
ListOfCatalog Category Admin	List of categories within a catalog.	Integration Component
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Name	Category name.	String
ListOfInternal Product	Internal products associated with the catalog.	Integration Component
Name	Product name.	String
ListOfCatalog Category Translation	Translation for the catalog categories.	Integration Component
Language Code	Language code.	String
ListOfProduct Catalog Translation	Translation for the catalog.	Integration Component
Language Code	Language code.	String

Message Description: SWI Catalog Admin - Delete

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SiebelMessage	The input or the output integration object instance.	Integration Object
Name	Catalog name.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Catalog Admin	An instance of the integration object Base Catalog Admin.	Integration Object

Node	Description	Type
Product Catalog	An integration component (XML tag) corresponding to the Product Catalog business component. You can create multiple instances of Product Catalog under ListOfBase Catalog Admin to process different product catalogs at one time.	Integration Component
Name	Catalog name.	String

Swagger Definitions for SWI Catalog Admin

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWICatalogAdmin
```

SWI Product

Use this REST API service to perform queries on products.

SWI Product Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
QueryPage	Queries for existing products using the QueryPage method.
QueryById	Queries for existing products using Row ID.
QueryByExample	Queries for existing catalogs using the QueryByExample method.

Message Description: SWI Product - QueryPage

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PageSize	Default is 10. Indicates the maximum number of integration object instances to be returned.	String
StartRowNum	Default is 0 (first page). Indicates the row in the result set for the QueryPage method to start retrieving a page of records.	String

Node	Description	Type
searchSpec	String containing the search specification.	String
ViewMode	Default is All. You must apply Visibility mode to the business object. Valid values are: • Manager • Sales Rep • Personal • Organization • Sub-Organization • Group • Catalog • All **Note:** The ViewMode user property on the integration object has priority over the ViewMode method argument.	String
SiebelMessage	The input or the output integration object instance.	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
ListOfSWI Admin ISS Product Definition	An instance of the integration object SWI Admin ISS Product Definition.	Integration Component
SWI Internal Product VOD	An integration component (XML tag) corresponding to the Internal Product business component. You can position one SWI Internal Product VOD under ListOfSWI Admin ISS Product Definition at one time.	Integration Component
Name	Name of the product.	String
Price Type	Pricing Type, for example Recurring.	String
External Integration Id	The record ID in the external system.	String
Product Def Type Code	The product definition type, for example, Customizable.	String

Node	Description	Type
ListOfSWI ISS Product VOD BusComp	Container element for the product structure information.	Integration Component
Vod Name	Name of the product.	String
ListOfPrice List Item	Price list item information.	Integration Component
List Price	The listed price.	String
ListOfProduct Compatibility - Subject	The compatibility rules under which this product is the subject.	Integration Component
Type	Type of the compatibility rule.	String
Adjustment Group	The adjustment group.	String
ListOfSWI Product User Defined Attribute	User-defined attributes	Integration Component
Name	Name of the attribute.	String
Value	Value of the attribute.	String
Last Page	Indicates if the user needs to continue the query for the next set of records, or if this is the last page.	String

Message Description: SWI Product - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The row ID of the product.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfSWI Admin ISS Product Definition	An instance of the integration object SWI Admin ISS Product Definition.	Integration Component
SWI Internal Product VOD	An integration component (XML tag) corresponding to the Internal Product business component. You can	Integration Component

Node	Description	Type
	position one SWI Internal Product VOD under ListOfSWI Admin ISS Product Definition at one time.	
Name	Name of the product.	String
Price Type	Pricing type, for example Recurring.	String
External Integration Id	The record ID in the external system.	String
Product Def Type Code	The product definition type, for example, Customizable.	String
ListOfSWI ISS Product VOD BusComp	Container element for the product structure information.	Integration Component
Vod Name	Name of the product.	String
ListOfPrice List Item	Price list item information.	Integration Component
List Price	The listed price.	String
ListOfProduct Compatibility - Subject	The compatibility rules under which this product is the subject.	Integration Component
Type	Type of the compatibility rule.	String
Adjustment Group	The adjustment group.	String
ListOfSWI Product User Defined Attribute	User-defined attributes.	Integration Component
Name	Name of the attribute.	String
Value	Value of the attribute.	String

Message Description: SWI Product - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfSWI ISS Product VOD BusComp	An instance of the integration object SWI Admin ISS Product Definition.	Integration Component
Name	Name of the catalog to be queried.	String

Node	Description	Type
External Integration Id	The product ID from the external system.	String
searchSpec	String containing the search specification.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfSWI Admin ISS Product Definition	An instance of the integration object SWI Admin ISS Product Definition.	Integration Component
SWI Internal Product VOD	An integration component (XML tag) corresponding to the Internal Product business component. You can position one SWI Internal Product VOD under ListOfSWI Admin ISS Product Definition at one time.	Integration Component
Name	Name of the product.	String
Price Type	Pricing Type, for example Recurring.	String
External Integration Id	The record ID in the external system.	String
Product Def Type Code	The product definition type, for example, Customizable.	String
ListOfSWI ISS Product VOD BusComp	Container element for the product structure information.	Integration Component
Vod Name	Name of the product.	String
ListOfPrice List Item	Price list item information.	Integration Component
List Price	The listed price.	String
ListOfProduct Compatibility - Subject	The compatibility rules under which this product is the subject.	Integration Component
Type	Type of compatibility rule.	String
Adjustment Group	The adjustment group.	String
ListOfSWI Product User Defined Attribute	User-defined attributes.	Integration Component

Node	Description	Type
Name	Name of the attribute.	String
Value	Value of the attribute.	String

Swagger Definitions for SWI Product

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIProduct
```

SWI Product Class

Use this REST API service to perform queries on product classes.

SWI Product Class Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
QueryPage	Queries high-volume, existing product classes using the QueryPage method.
QueryById	Queries for a single existing product class using Row ID.
QueryByExample	Queries for existing product classes matching the criteria provided in the request.

Message Description: SWI Product Class - QueryPage

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PageSize	Default is 10. Indicates the maximum number of integration object instances to be returned.	String
StartRowNum	Default is 0 (first page). Indicates the row in the result set for the QueryPage method to start retrieving a page of records.	String
searchSpec	String containing the search specification.	String

Node	Description	Type
ViewMode	Default is All. You must apply Visibility mode to the business object. Valid values are: • Manager • Sales Rep • Personal • Organization • Sub-Organization • Group • Catalog • All **Note:** The ViewMode user property on the integration object has priority over the ViewMode method argument.	String
SiebelMessage	The input or the output integration object instance.	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
ListOfSWIAdmin ISS Product Definition	An instance of the integration object SWIAdminISSProductDefinition.	Integration Component
SWI ISS Class VOD BusComp	An integration component corresponding to the internal Product business component. You can position SWI ISS Class VOD BusComp under ListOfSWIAdminISSClassDefinitionIO at one time.	Integration Component
Name	Product class name.	String
Price Type	Pricing type, for example, Recurring.	String
Vod Name	Product name.	String
Vod ID	The product class object ID.	String
Vod Name	The record ID in the external system.	String

Node	Description	Type
Active Flag	Indicates if the product class is active.	String
Locked Flag	Indicates if the product class is locked.	String
ListOfSWI ISS Product Attribute VBC	Container element for the product structure information.	Integration Component
Name	Name of the attribute in the class.	String
Attribute Name	The attribute definition name.	String
Default Value	The default value of the attribute.	String
ListOfSWI ISS Attribute Definition Lang BusComp	Container element for the price list item information.	Integration Component
Vod Name	Attribute name.	String
Lang CD	Language code.	String
ListOfSWI ISS Product Details BusComp	Container element for the product details.	Integration Component
Object Name	The product name.	String
Object ID	The product ID.	String
Last Page	Indicates if the user needs to continue the query for the next set of records, or if this is the last page.	String

Message Description: SWI Product Class - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The product class row ID.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfSWIAdmin ISS Product Definition	An instance of the integration object SWIAdminISSProductDefinition.	Integration Component
SWI ISS Class VOD BusComp	An integration component corresponding to the internal Product business component. You can position SWI ISS Class VOD BusComp under ListOfSWIAdminISSClassDefinitionIO at one time.	Integration Component
Name	Product class name.	String
Price Type	Pricing type, for example, Recurring.	String
Vod Name	Product name.	String
Vod ID	The class object ID.	String
Vod Name	The record ID in the external system.	String
Active Flag	Indicates if the class is active.	String
Locked Flag	Indicates if the class is locked.	String
ListOfSWI ISS Product Attribute VBC	Container element for the product structure information.	Integration Component
Name	Name of the attribute in the class.	String
Attribute Name	The attribute definition name.	String
Default Value	The default value of the attribute.	String
ListOfSWI ISS Attribute Definition Lang BusComp	Container element for the price list item information.	Integration Component
Vod Name	Attribute name.	String
Lang CD	Language code.	String
ListOfSWI ISS Product Details BusComp	Container element for the product details.	Integration Component
Object Name	The product name.	String
Object ID	The product ID.	String

Message Description: SWI Product Class - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfSWIAdminISSClassDefinitionIO	An instance of the integration object ListOfSWIAdminISSClassDefinitionIO.	Integration Component
Name	Name of the product to query.	String
External Integration Id	The product ID from the external system.	String
searchSpec	String containing the search specification.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfSWIAdmin ISS Product Definition	An instance of the integration object SWIAdminISSProductDefinition.	Integration Component
SWI ISS Class VOD BusComp	An integration component corresponding to the internal Product business component. You can position SWI ISS Class VOD BusComp under ListOfSWIAdminISSClassDefinitionIO at one time.	Integration Component
Name	Product class name.	String
Price Type	Pricing type, for example, Recurring.	String
Vod Name	Product name.	String
Vod ID	The product class object ID.	String
Vod Name	The record ID in the external system.	String
Active Flag	Indicates if the product class is active.	String
Locked Flag	Indicates if the product class is locked.	String
ListOfSWI ISS Product Attribute VBC	Container element for the product structure information.	Integration Component
Name	Name of the attribute in the class.	String

Node	Description	Type
Attribute Name	The attribute definition name.	String
Default Value	The default value of the attribute.	String
ListOfSWI ISS Attribute Definition Lang BusComp	Container element for the price list item information.	Integration Component
Vod Name	Attribute name.	String
Lang CD	Language code.	String
ListOfSWI ISS Product Details BusComp	Container element for the product details.	Integration Component
Object Name	The product name.	String
Object ID	The product ID.	String

Swagger Definitions for SWI Product Class

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIProductClass
```

SWI Product Attribute

Use this REST API service to query product attributes.

SWI Product Attribute Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
QueryPage	Queries high-volume, existing product attributes using the QueryPage method.
QueryById	Queries for single existing product attribute using Row ID.
QueryByExample	Queries existing product attributes matching the criteria provided in the request.

Message Description: SWI Product Attribute - QueryPage

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PageSize	Default is 10. Indicates the maximum number of integration object instances to be returned.	String
StartRowNum	Default is 0 (first page). Indicates the row in the result set for the QueryPage method to start retrieving a page of records.	String
searchSpec	String containing the search specification.	String
ViewMode	Default is All. You must apply Visibility mode to the business object. Valid values are: • Manager • Sales Rep • Personal • Organization • Sub-Organization • Group • Catalog • All **Note:** The ViewMode user property on the integration object has priority over the ViewMode method argument.	String
SiebelMessage	The input or the output integration object instance.	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
ListOfSWIAdminISSAttributeDefnIO	An instance of the integration object SWIAdminISSAttributeDefnIO.	Integration Object
SWI Attribute VOD BusComp	An integration component corresponding to the internal Product business component. You can	Integration Component

Node	Description	Type
	position one SWI Attribute VOD BusComp under ListOfSWIAdminISSClassDefinitionIO at one time.	
Name	Name of the class.	String
Vod Id	The class object ID.	String
Vod Name	The record id in the external system.	String
Vod Type	The versioned object type.	String
Active Flag	Indicates if the class is active.	String
Locked Flag	Indicates if the class is locked.	String
ListOfSWI Attribute Details BusComp	Container element for the attribute detail information.	Integration Component
Data Type	Indicates the attribute data type, for example, String or Integer.	String
Domain Type	Indicates the attribute domain type, for example, Enumerated.	String
Max Value	The maximum attribute value.	String
Min Value	The minimum attribute value.	String
Unit of Measure	The unit measure of the attribute.	String
ListOfSWI ISS Attribute Definition Lang BusComp	Language information of the attribute.	Integration Component
Vod Name	The attribute name.	String
Lang CD	Language code.	String
ListOfSWI Attribute Values BusComp	Container element for the domain values of the attribute.	Integration Component
Value	The attribute value.	String
Sequence	Sequence number of the attribute value.	String

Node	Description	Type
Last Page	Indicates if the user needs to continue the query for the next set of records, or if this is the last page.	String

Message Description: SWI Product Attribute - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The row ID of the product attribute.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfSWIAdminISSAttributeDefnIO	An instance of the integration object SWIAdminISSAttributeDefnIO.	Integration Object
SWI Attribute VOD BusComp	An integration component corresponding to the internal Product business component. You can position one SWI Attribute VOD BusComp under ListOfSWIAdminISSClassDefinitionIO at one time.	Integration Component
Name	Name of the class.	String
Vod Id	The class object ID.	String
Vod Name	The record id in the external system.	String
Vod Type	The versioned object type.	String
Active Flag	Indicates if the class is active.	String
Locked Flag	Indicates if the class is locked.	String
ListOfSWI Attribute Details BusComp	Container element for the attribute detail information.	Integration Component
Data Type	Indicates the attribute data type, for example, String or Integer.	String
Domain Type	Indicates the attribute domain type, for example, Enumerated.	String

Node	Description	Type
Max Value	The maximum attribute value.	String
Min Value	The minimum attribute value.	String
Unit of Measure	The unit measure of the attribute.	String
ListOfSWI ISS Attribute Definition Lang BusComp	Language information of the attribute.	Integration Component
Vod Name	The attribute name.	String
Lang CD	Language code.	String
ListOfSWI Attribute Values BusComp	Container element for the domain values of the attribute.	Integration Component
Value	The attribute value.	String
Sequence	Sequence number of the attribute value.	String

Message Description: SWI Product Attribute - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfSWIAdminISSAttributeDefnIO	An instance of the integration object SWIAdminISSAttributeDefnIO.	Integration Object
VOD Name	Attribute name to be queried.	String
External Integration Id	The attribute ID from the external system.	String
searchSpec	String containing the search specification.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfSWIAdminISSAttributeDefnIO	An instance of the integration object SWIAdminISSAttributeDefnIO.	Integration Object

Node	Description	Type
SWI Attribute VOD BusComp	An integration component corresponding to the internal Product business component. You can position one SWI Attribute VOD BusComp under ListOfSWIAdminISSClassDefinitionIO at one time.	Integration Component
Name	Name of the class.	String
Vod Id	The class object ID.	String
Vod Name	The record id in the external system.	String
Vod Type	The versioned object type.	String
Active Flag	Indicates if the class is active.	String
Locked Flag	Indicates if the class is locked.	String
ListOfSWI Attribute Details BusComp	Container element for the attribute detail information.	Integration Component
Data Type	Indicates the attribute data type, for example, String or Integer.	String
Domain Type	Indicates the attribute domain type, for example, Enumerated.	String
Max Value	The maximum attribute value.	String
Min Value	The minimum attribute value.	String
Unit of Measure	The unit measure of the attribute.	String
ListOfSWI ISS Attribute Definition Lang BusComp	Language information of the attribute.	Integration Component
Vod Name	The attribute name.	String
Lang CD	Language code.	String
ListOfSWI Attribute Values BusComp	Container element for the domain values of the attribute.	Integration Component
Value	The attribute value.	String
Sequence	Sequence number of the attribute value.	String

Node	Description	Type

Swagger Definitions for SWI Product Attribute

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIProductAttribute
```

SWI Attribute Import

Use this REST API service to insert or update of product details that can be imported into external product information in the Siebel database. All the APIs share the same request and response payloads.

SWI Attribute Import Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
Upsert	Inserts or updates an external attribute.

Message Description: SWI Attribute Import Service - Upsert

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SyncChild	If this flag is set to Y, then the upsert operation behaves like a synchronize operation. The default value of this flag is Y.	String
UTCCanonical	When this value is set to Y, all date-type fields in the payload are in UTC standard format.	String
SiebelMessage	The input or the output integration object instance.	Integration Object
ListOfSWIAttributeIntegrationIO	Required. An instance of the integration object SWIAttributeIntegrationIO.	Integration Object
SWI Attribute Integration VBC	An integration component corresponding to the SWI Attribute Integration VBC business component under ListOfSWIAttributeIntegrationIO.	Integration Component

Node	Description	Type
Workspace Name	The name of the product administration workspace.	String
Workspace Reuse Flag	Indicates if the workspace can be used again for other imports.	String
Workspace Release Flag	Indicates if the workspace should be released after importing.	String
Active Workspace Flag	Indicates if this is the active flag for scenario builder.	String
ListOfSWI Attribute Details VBC	Container element for attribute details of ListOfSWI Attribute Details VBC.	Integration Component
Data Type	Indicates the attribute data type, for example, String, or Integer.	String
Domain Type	Indicates the attribute domain type, for example, Enumerated.	String
Max Value	The maximum value of the attribute.	String
Min Value	The minimum value of the attribute.	String
Unit of Measure	The unit measure of the attribute.	String
ListOfSWI ISS Attribute Definition Lang VBC	Container element for the attribute definition language.	Integration Component
Language Code	Language code	String
Display Name	The display name of the language.	String
ListOfSWI Attribute Values VB	The domain values of the attribute.	Integration Component
Value	The attribute value.	String
Sequence	Sequence number of the attribute value.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
SiebelMessage	The input or the output integration object instance.	Integration Object

Node	Description	Type
SyncChild	If this flag is set to Y, then the upsert operation behaves like a synchronize operation. The default value of this flag is Y.	String
UTCCanonical	When this value is set to Y, all date-type fields in the payload are in UTC standard format.	String
SiebelMessage	The input or the output integration object instance.	Integration Object
ListOfSWIAttributeIntegrationIO	Required. An instance of the integration object SWIAttributeIntegrationIO.	Integration Object
SWI Attribute Integration VBC	An integration component corresponding to the SWI Attribute Integration VBC business component under ListOfSWIAttributeIntegrationIO.	Integration Component
Workspace Name	The name of the product administration workspace.	String
Workspace Reuse Flag	Indicates if the workspace can be used again for other imports.	String
Workspace Release Flag	Indicates if the workspace should be released after importing.	String
Active Workspace Flag	Indicates if this is the active flag for scenario builder.	String
ListOfSWI Attribute Details VBC	Container element for attribute details of ListOfSWI Attribute Details VBC.	Integration Component
Data Type	Indicates the attribute data type, for example, String, or Integer.	String
Domain Type	Indicates the attribute domain type, for example, Enumerated.	String
Max Value	The maximum value of the attribute.	String
Min Value	The minimum value of the attribute.	String
Unit of Measure	The unit measure of the attribute.	String
ListOfSWI ISS Attribute Definition Lang VBC	Container element for the attribute definition language.	Integration Component

Node	Description	Type
Language Code	Language code.	String
Display Name	The display name of the language.	String
ListOfSWI Attribute Values VB	The domain values of the attribute.	Integration Component
Value	The attribute value.	String
Sequence	Sequence number of the attribute value.	String

Swagger Definitions for SWI Attribute Import

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIAttributeImport
```

SWI Product Class Import

Use this REST API service to insert or update of product class details that can be imported into external product information in the Siebel database. All the APIs share the same request and response payloads.

SWI Product Class Import Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
Upsert	Used to insert or update an external product class.

Message Description: SWI Product Class Import - Upsert

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SyncChild	If this flag is set to Y, then the upsert operation behaves like a synchronize operation. The default value of this flag is Y.	String
UTCCanonical	When this value is set to Y, all date-type fields in the payload are in UTC standard format.	String

Node	Description	Type
SiebelMessage	The input or the output integration object instance.	Integration Object
ListOfSWIProductClassIntegrationIO	Required. An instance of the integration object SWIProductIntegration.	Integration Object
SWI Product Integration VBC	An integration component corresponding to the SWI Attribute Integration VBC business component under ListOfSWIAttributeIntegrationIO.	Integration Component
Workspace Name	The name of the product administration workspace.	String
Workspace Reuse Flag	Indicates if the workspace can be used again for other imports.	String
Workspace Release Flag	Indicates if the workspace should be released after importing.	String
Active Workspace Flag	Indicates if this is the active flag for scenario builder.	String
ListOfSWI Product Class Defn VBC	Container element for class definition information.	Integration Component
VOD Name	Class name.	String
External Integration Id	The external record ID.	String
Parent VOD Name	The parent class name.	String
Release Flag	To release the class after importing.	String
ListOfSWI ISS Product Attribute VBC	The class attributes.	Integration Component
Name	Name of the attribute in the class.	String
Attribute Name	Container element for the attribute definition name.	String
Default Value	Default value.	String
Sequence Number	The sequence number of the attribute.	String
ListOfSWI Product Structure VBC	Container element for the class structure and relationships.	String
Component Product Name	If this is a port, this is the product name.	String

Node	Description	Type
Component Class Name	If this is a class port, this is the class name.	String
Domain Type	Indicates if the domain type is: Product, Port, or DynPort.	String
Max Cardinality	Maximum cardinality.	String
Min Cardinality	Minimum cardinality.	String
Default Cardinality	Default cardinality.	String
Relationship Name	The name of the relationship.	String
ListOfSWI Product Structure Domain VBC	If this is a class port, this is the domain product.	Integration Component

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
SiebelMessage	The input or the output integration object instance.	Integration Object
ListOfSWIProductClassIntegrationIO	Required. An instance of the integration object SWIProductIntegration.	Integration Object
SWI Product Integration VBC	An integration component corresponding to the SWI Attribute Integration VBC business component under ListOfSWIAttributeIntegrationIO.	Integration Component
Workspace Name	The name of the product administration workspace.	String
Workspace Reuse Flag	Indicates if the workspace can be used again for other imports.	String
Workspace Release Flag	Indicates if the workspace should be released after importing.	String
Active Workspace Flag	Indicates if this is the active flag for scenario builder.	String
ListOfSWI Product Class Defn VBC	Container element for class definition information.	Integration Component
VOD Name	Class name.	String

Node	Description	Type
External Integration Id	The external record ID.	String
Parent VOD Name	The parent class name.	String
Release Flag	Indicates if the class can be released after importing.	String
ListOfSWI ISS Product Attribute VBC	Container element for the class attributes.	Integration Component
Name	Name of the attribute in the class.	String
Attribute Name	The attribute definition name.	String
Default Value	Default value.	String
Sequence Number	The sequence number of the attribute.	String
ListOfSWI Product Structure VBC	Container element for class structure and relationships.	String
Component Product Name	If this is a port, this is the product name.	String
Component Class Name	If this is a class port, this is the class name.	String
Domain Type	Indicates if the domain type is: Product, Port, or DynPort.	String
Max Cardinality	Maximum cardinality.	String
Min Cardinality	Minimum cardinality.	String
Default Cardinality	Default cardinality.	String
Relationship Name	The name of the relationship.	String
ListOfSWI Product Structure Domain VBC	If this is a class port, this is the domain product.	Integration Component

Swagger Definitions for SWI Product Class Import

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIProductClassImport
```

SWI Product Import

Use this REST API service to insert or update external product information that can be imported into the Siebel database. All the APIs share the same request and response payloads.

SWI Product Import Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
Insert	Inserts external product information into Siebel database.
Update	Updates external product information.
Upsert	Inserts or updates an external product.

Message Description: SWI Product Import Service - Insert/Update/Upsert

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SWIProductIntegrationIO	Required. An instance of the integration object SWIProductIntegration.	Integration Object
SyncChild	If this flag is set to Y, then the upsert operation behaves like a synchronize operation. The default value of this flag is Y.	String
UTCCanonical	When this value is set to Y, all date-type fields in the payload are in UTC standard format.	String
Active Flag	Indicates if the record is active.	String
Billable Flag	Indicates if the record is billable.	String
Business Unit ID	Business unit ID associated with the product record.	String
Description	Description.	String
Effective End Date	Effective end date of record.	String

Node	Description	Type
Effective Start Date	Effective start date of record.	String
External Inventory System Ref	Reference number from the external inventory system.	String
External Integration ID	The record ID from the external system.	String
Integration ID	Integration ID.	String
Inventory Flag	Indicates if the product is tracked in inventory.	String
Lead Time	Lead time for product delivery.	String
Product Cost	Product cost.	String
Product Name	Product name.	String
Product Serialized Flag	This serialized flag indicates if instances of this product are tracked as serialized assets, or are tracked as quantities of product. Instances of a serialized product each carry a unique serial number, such as a license number, or policy number. Quantities of non-serialized assets are sometimes tracked as lots which may have a unique lot number.	String
Product Version	Version of product.	String
Release Flag	Indicates if a product will be released when synchronized.	String
Sales Product Flag	Sales product flag. Sales products appear on the product menu when a user defines products for an opportunity.	String
Service Terms	Service terms associated with product.	String
Ship Flag	Indicates if a product has to be shipped.	String
Status	The product status can be Active or Inactive.	String
Structure Type	Depending on the product type, users are allowed to create a simple bill of materials without using Siebel Configurator.	String
Unit of Measure	Unit of measure.	String

Node	Description	Type
ListofProduct Inv loc	Location of product in inventory.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
SWIProductIntegrationIO	Returns an instance of SWIProductIntegration. **Note:** This returns only the fields specified in the status key of integration components belonging to SWI Admin ISS Product Definition integration object.	Integration Object
ErrorMessage	Error message (if any).	String
ID	Product ID.	String
Active Flag	Indicates if the record is active.	String
Auto Explode Flag	Indicates if the product needs to be automatically ungrouped and then to reset the ungrouped quantity to 1.	String
Check Eligibility Flag	Indicates if eligibility and compatibility rules are required for this product.	String
Description	Description.	String
Effective End Date	Effective end date of record.	String
Effective Start Date	Effective start date of record.	String
External Inventory System Ref	Reference number from the external inventory system.	String
External Integration ID	The record ID from the external system.	String
Inclusive Eligibility Flag	Indicates whether or not to process a particular product for inclusive eligibility rules. Inclusive eligibility rules are defined in the eligibility matrix with the matrix rule type code set to Inclusive.	String
Integration ID	Integration ID.	String
Inventory Flag	Indicates if product is tracked in inventory.	String

Node	Description	Type
Lead Time	Lead time for product delivery.	String
Orderable Flag	Indicates if a product can be ordered.	String
Price Type	Price type, such as One-time, Recurring, Usage, and so on.	String
Primary Organization ID	ID of primary organization associated with the product record.	String
Product Cost	Cost of product.	String
Product ID	ID of product.	String
Product Name	Name of product.	String
Product Type	Product type.	String
Product Version	Version of product.	String
Release Flag	Indicates if a product will be released when synchronized.	String
Status	Product order status.	String
ListofProductInvLoc	Location of product in inventory.	String
Type	Product type.	String
Unit of Measure	Unit of measure.	String
Vendor ID	ID of product vendor.	String

Swagger Definitions for SWI Product Import

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIProductImport
```

SWI Smart Part Number

Use this REST API service to perform basic operations such as create, update, and query Smart part numbers.

SWI Smart Part Number Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
InsertOrUpdate	Creates or updates a smart part number.
Update	Updates a smart part number.
Synchronize	Synchronizes a smart part number.
QueryById	Queries for single existing product class using the Row ID.
QueryByExample	Queries for existing product classes matching the sample in the request.
Delete	Deletes a smart part number.

Message Description: SWI Smart Part Number - InsertOrUpdate

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod Id	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component

Node	Description	Type
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod Id	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String

Node	Description	Type
Attribute %i value	i in the range (0..29) The predefined matrix.	String

Message Description: SWI Smart Part Number - Update

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod ID	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod ID	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

Message Description: SWI Smart Part Number - Synchronize

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod Id	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component

Node	Description	Type
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod ID	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String

Node	Description	Type
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

Message Description: SWI Smart Part Number - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The row ID of the Smart part number.	String
OutputIntObjectName	Name of the output integration object.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod Id	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String

Node	Description	Type
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

Message Description: SWI Smart Part Number - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod Id	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod Id	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component

Node	Description	Type
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

Message Description: SWI Smart Part Number - Delete

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod Name	The parent class name in the external system.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Admin ISS Class Definition	An instance of the integration object Base Admin ISS Class Definition.	Integration Object
Vod ID	The parent class object ID.	String
Vod Name	The parent class name in the external system.	String
ListOfSPN Generation Methods	List of SPN generation methods.	Integration Component
Name	Name of the SPN generation method.	String
Type	Type of the SPN generation method, for example, Predefined.	String
ListOfSPN Definition Attributes	List of SPN definition attributes.	Integration Component

Node	Description	Type
Attribute Name	The attribute name.	String
Attribute Index	The attribute index.	String
Method Id	The SPN generation method ID.	String
ListOfSPN Definition Matrix	List of SPN definition matrix.	Integration Component
Part Number	The part number of the attribute.	String
Attribute %i value	i in the range (0..29) The predefined matrix.	String

Swagger Definitions for SWI Smart Part Number

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWISmartPartNumber
```

SWI Price List Item

Use this REST API service to perform basic operations, such as create, update, delete or query for price list line items.

SWI Price List Item Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
Insert	Creates a price list line item.
Update	Updates a price list line item.
InsertOrUpdate	Creates a new price list item or updates an existing price list item.
Delete	Deletes a price list line item.
QueryByExample	Queries for existing price list line items using the QueryPage method.
QueryById	Queries for existing price list line items using the Row ID.

Name	Description

Message Description: SWI Price List Item - Insert

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SWIISSPriceListItemIO	Required. An instance of the Integration Object SWIISSPriceListItem. To insert a price list line item, note that the payload must provide the following user keys: • PriceListId - price list • ProductId - product • StartDate - start date	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
PriceListItemID	The price list item ID created after the insert operation.	String
PriceListID	The price list ID associated with the price list item ID.	String
ProductID	The product ID associated with the price list item ID.	String
StartDate	Start date of the price list. StartDate is part of the user key.	String

Message Description: SWI Price List Item - Update

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SWIISSPriceListItemIO	Required. An instance of the Integration Object SWIISSPriceListItem. To insert a price list line item, note that the payload must provide the following user keys: • PriceListId - price list • ProductId - product • StartDate - start date	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
PriceListItemID	The price list item ID updated after the update operation.	String
PriceListID	The price list ID associated with the price list item ID.	String
ProductID	The product ID associated with the price list item ID.	String

Message Description: SWI Price List Item - InsertOrUpdate

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SWIISSPriceListItemIO	Required. An instance of the Integration Object SWIISSPriceListItem. The parent header is: Price List Item - Import	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
PriceListItemID	The price list item ID created or updated after the upsert operation.	String
PriceListID	The price list ID associated with the price list item ID.	String
ProductID	The product ID associated with the price list item ID.	String

Message Description: SWI Price List Item - Delete

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SWIISSPriceListItemIO	Required. An instance of the Integration Object SWIISSPriceListItem. The parent header is: Price List Item - Import	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
PriceListItemID	The price list item ID of the record that is deleted after the delete operation.	String
PriceListID	The price list ID that is associated with the price list item ID.	String
ProductID	The product ID associated with the price list item ID.	String

Message Description: SWI Price List Item - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
SWIISSPriceListItemIO	Required. An instance of the Integration Object SWIISSPriceListItem. The parent header is: Price List Item - Import	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
PriceListItemID	The ID of the price list item.	String
PriceListID	The price list ID associated with the price list item ID.	String
ProductID	The product ID associated with price list item ID.	String

Message Description: SWI Price List Item - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowID	Required.	Not applicable.
SWIISSPriceListItemIO	Required. An instance of the Integration Object SWIISSPriceListItem. The parent header is: Price List Item - Import	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
PriceListItemID	The ID of the price list item.	String
PriceListID	The price list ID associated with the price list item ID.	String
ProductID	The product ID associated with price list item ID.	String

Swagger Definitions for SWI Price List Item

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIPriceListItem
```

SWI Product Line

Use this REST API service to perform basic operations such as create, update, and query product lines. However, before you begin, you must deploy the SWI Admin Product Line integration object to the production environment as described below.

To deploy SWI Admin Product Line integration object

1. Log in to Siebel Tools.
2. In the Object Explorer, click Integration Object.
3. In the Integration Object list, locate and right-click: SWI Admin Product Line.
4. Click Undeploy.
5. Right-click the integration object and select Deploy to Runtime Database.

SWI Product Line Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
Insert	Creates a product line.
InsertOrUpdate	Creates a new product line or updates an existing product line
Synchronize	Synchronizes a new product line, or synchronizes data to an existing product line.
QueryById	Queries for existing product lines using Row ID.
QueryByExample	Queries for existing product lines using the QueryByExample method.

Name	Description
QueryPage	Queries for existing product lines using the QueryPage method.
Delete	Deletes a product line.

Message Description: SWI Product Line Service - Insert

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object
AdminProductLine	An integration component (XML tag) corresponding to the Admin Product Line business component. You can create multiple instances of AdminProductLine under ListOfAdminProductLine to process different product lines at one time.	Integration Component
Description	Description of the product line.	String
Name	Name of the product line.	String
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Position	Position, such as Administrator.	String
ProductLineManager	Product line manager.	String
Division	Division within the company.	String
ListofAdminProductLine_ InternalProduct	Container element for internal products associated with the product line.	Integration Component
Product	Product name.	String
IntegrationId	Integration ID of the product record.	String
Primary Organization	The primary organization for the product.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object
AdminProductLine	An integration component (XML tag) corresponding to the Admin Product Line business component. You can create multiple instances of AdminProductLine under ListOfAdminProductLine to process different product lines at one time.	Integration Component
Id	Product line ID.	String
Description	Description of the product line.	String
Name	Name of the product line.	String
PrimaryPositionId	Primary position ID.	String
PrimaryProductId	ID of the primary product.	String

Message Description: SWI Product Line - InsertOrUpdate

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object
Description	Description of the product line.	String
Name	Name of the product line.	String
ListofAdminProductLine_Position	Container element for positions associated with the product line.	Integration Component
Position	Position, such as Administrator.	String
ProductLineManager	Product line manager.	String
Division	Division within the company.	String
ListofAdminProductLine_ InternalProduct	Internal products associated with the product line.	Integration Component
Product	Product name.	String

Node	Description	Type
IntegrationId	Integration ID of the product record.	String
Primary Organization	The primary organization for the product.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object
AdminProductLine	An integration component (XML tag) corresponding to the Admin Product Line business component. You can create multiple instances of AdminProductLine under ListOfAdminProductLine to process different product lines at one time.	Integration Component
Description	Description of the product line.	String
Name	Name of the product line.	String
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
AdminProductLine_Position IsPrimaryMVG	An instance of the Admin Product Line_Position integration component that corresponds to a primary MVG. It is denoted by the attribute IsPrimaryMVG and is set to Y. This attribute is a hidden integration component field, and it does not have a corresponding business component field.	Integration Component
Position	Position, such as Administrator.	String
ProductLineManager	Product line manager.	String
Division	Division within the company.	String
ListofAdminProductLine_ InternalProduct	Internal products associated with the product line.	Integration Component
Product	Product name.	String
IntegrationId	Integration ID of the product record.	String
Primary Organization	The primary organization for the product.	String

Message Description: SWI Product Line - Synchronize

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object
Description	Description of the product line.	String
Name	Name of the product line.	String
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Position	Position, such as Administrator.	String
ProductLineManager	Product line manager.	String
Division	Division within the company.	String
ListofAdminProductLine_ InternalProduct	Internal products associated with the product line.	Integration Component
Product	Product name.	String
IntegrationId	Integration ID of the product record.	String
Primary Organization	The primary organization for the product.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object
Id	Product line ID.	String
Description	Description of the product line.	String
Name	Name of the product line.	String
PrimaryPositionId	Primary Position ID.	String

Node	Description	Type
PrimaryProductId	ID of the primary product.	String
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
Position	Position, such as Administrator.	String
PositionId	Position ID.	String
PrimaryEmployeeId	ID of the primary employee.	String
ProductLineManager	Product line manager.	String
Division	Division within the company.	String
ListofAdminProductLine_ InternalProduct	Internal products associated with the product line.	Integration Component
Product	Product name.	String
Part	Part number.	String
Primary Organization	The primary organization for the product.	String

Message Description: SWI Product Line - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The row ID of the product line.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object
AdminProductLine	An integration component (XML tag) corresponding to the Admin Product Line business component. You can create multiple instances of AdminProductLine under ListOfAdminProductLine to process different product lines at one time.	Integration Component

Node	Description	Type
Description	Description of the product line.	String
IntegrationId	Integration ID of the product record.	String
Name	Name of the product line.	String
PrimaryPositionId	ID of the primary position.	String
ListOfAdminProductLine_Position	An instance of the integration object Admin Product Line_Position.	Integration Object
AdminProductLine_Position IsPrimaryMVG	An instance of the Admin Product Line_Position integration component that corresponds to a primary MVG. It is denoted by the attribute IsPrimaryMVG and is set to Y. This attribute is a hidden integration component field, and it does not have a corresponding business component field.	Integration Component
Position	Position, such as Administrator.	String
Position_Id	ID of the position.	String
PrimaryEmployeeId	ID of the primary employee.	String
ProductLineManager	Product line manager.	String
WorkPhoneNumber	Work phone number.	String
DivisionIntegrationId	Division integration ID.	String
Division	Division within the company.	String
ListOfAdminProductLine_InternalProduct	Internal products associated with the product line.	Integration Component
AdminProductLine_ InternalProductIsPrimaryMVG	An instance of the Admin Product Line_Internal integration component that corresponds to a primary MVG. It is denoted by the attribute IsPrimaryMVG and is set to Y. This attribute is a hidden integration component field, and it does not have a corresponding business component field.	Integration Component
Product	Product name.	String
Part	Part number.	String

Node	Description	Type
IntegrationId	Integration ID of the product record.	String
PrimaryOrganization	The primary organization for the product.	String
VendorIntegrationId	Integration ID of the vendor.	String

Message Description: SWI Product Line - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfSWI Admin Product Line	An instance of the integration object ListOfAdminProductLine.	Integration Object
Admin Product Line	Integration Component (XML Tag) corresponding to the Admin Product Line Business component. You can create multiple instances of AdminProductLine under ListOfAdminProductLine to process different product lines at one time.	Integration Component
Id	Product ID	String
Created	System timestamp	String
Updated	System timestamp	String
ConflictId	Conflict ID	String
ModId	Modification Id	String
Description	Description of the product line.	String
IntegrationId	Integration ID of the product record.	String
PrimaryPositionId	ID of the primary position.	String
PrimaryProductId	ID of the primary product.	String
ListOfAdminProductLine_Position	Positions associated with the product line.	Integration Component
AdminProductLine_InternalProductIsPrimaryMVG	Admin Product Line_Internal Product integration component instance that corresponds to a primary MVG and is denoted by the attribute IsPrimaryMVG set	Integration Component

Node	Description	Type
	to Y. This attribute is a hidden integration component field and does not have a corresponding business component field.	
Position	Position, such as Administrator.	String
Position_Id	ID of the position.	String
PrimaryEmployeeId	ID of the primary employee.	String
ProductLineManager	Product line manager.	String
WorkPhoneNumber	Work phone number.	String
DivisionIntegrationId	Division integration ID.	String
Division	Division within the company.	String
ListOfAdminProductLine_InternalProduct	Internal products associated with the product line.	Integration Component
AdminProductLine_InternalProductIsPrimaryMVG	Admin Product Line_Internal Product integration component instance that corresponds to a primary MVG and is denoted by the attribute IsPrimaryMVG set to Y. This attribute is a hidden integration component field and does not have a corresponding business component field.	Integration Component
Product	Product name.	String
Part	Part number.	String
IntegrationId	Integration ID of the product record.	String
PrimaryOrganization	The primary organization for the product.	String
VendorIntegrationId	Integration ID of the vendor.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdminProductLine	An instance of the integration object ListOfAdminProductLine.	Integration Object

Node	Description	Type
AdminProductLine	An integration component (XML tag) corresponding to the Admin Product Line business component. You can create multiple instances of AdminProductLine under ListOfAdminProductLine to process different product lines at one time.	Integration Component
Description	Description of the product line.	String
IntegrationId	Integration ID of the product record.	String
PrimaryPositionId	ID of the primary position.	String
PrimaryProductId	ID of the primary product.	String
ListOfAdminProductLine_Position	Positions associated with the product line.	Integration Component
AdminProductLine_InternalProductIsPrimaryMVG	Admin Product Line_Internal Product integration component instance that corresponds to a primary MVG and is denoted by the attribute IsPrimaryMVG set to Y. This attribute is a hidden integration component field and does not have a corresponding business component field.	Integration Component
Position	Position, such as Administrator.	String
Position_Id	ID of the position.	String
PrimaryEmployeeId	ID of the primary employee.	String
ProductLineManager	Product line manager.	String
WorkPhoneNumber	Work phone number.	String
DivisionIntegrationId	Division integration ID.	String
Division	Division within the company.	String
ListOfAdminProductLine_InternalProduct	Internal products associated with the product line.	Integration Component
Product	Product name.	String
Part	Part number.	String
IntegrationId	Integration ID of the product record.	String

Node	Description	Type
PrimaryOrganization	The primary organization for the product.	String
VendorIntegrationId	Integration ID of the vendor.	String

Message Description: SWI Product Line - QueryPage

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
PageSize	Default is 10. Indicates the maximum number of integration object instances to be returned.	String
StartRowNum	Default is 0 (first page). Indicates the row in the result set for the QueryPage method to start retrieving a page of records.	String
ViewMode	Default is All. You must apply Visibility mode to the business object. Valid values are: • Manager • Sales Rep • Personal • Organization • Sub-Organization • Group • Catalog • All Note: The ViewMode user property on the integration object has priority over the ViewMode method argument.	String
SiebelMessage	The input or the output integration object instance.	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListofAdminProductLine	An instance of the integration object ListofAdminProductLine.	Integration Object
AdminProductLine	Integration Component (XML Tag) corresponding to the Admin Product Line Business component. You can	Integration Component

Node	Description	Type
	create multiple instances of AdminProductLine under ListOfAdminProductLine to process different product lines at one time.	
Description	Description of the product line.	String
IntegrationId	Integration ID of the product line.	String
Name	Name of the product line.	String
PrimaryPositionId	ID of the primary position.	String
PrimaryProductId	ID of the primary product.	String
ListofAdminProductLine_Position	Positions associated with the product line.	Integration Component
AdminProductLine_ PositionIsPrimaryMVG	An instance of the Admin Product Line_Position integration component that corresponds to aprimary MVG. It is denoted by the attribute IsPrimaryMVG and is set to Y. This attribute is a hidden integration component field, and it does not have a corresponding business component field.	Integration Component
Position	Position, such as Administrator.	String
PrimaryEmployeeId	ID of the primary employee.	String
ProductLineManager	Product line manager.	String
WorkPhoneNumber	Work phone number.	String
DivisionIntegrationId	Integration ID of the division.	String
Division	Division within the company.	String
ListOfAdminProductLine_InternalProduct	Internal products associated with the product line.	Integration Component
Product	Product name.	String
Part	Part number.	String
IntegrationId	Integration ID of the product record.	String
PrimaryOrganization	The primary organization for the product.	String
VendorIntegrationId	Integration ID of the vendor.	String

Message Description: SWI Product Line - Delete

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBaseAdminProductLine	An instance of the integration object ListOfBaseAdminProductLine.	Integration Object
Name	Product line name.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBaseAdminProductLin	An instance of the integration object ListOfBaseAdminProductLine.	Integration Object
ID	ID of the product line.	String
Description	Description of the product line.	String
Name	Name of the product line.	String
PrimaryPositionId	ID of the primary position.	String
PrimaryProductId	ID of the primary product.	String

Swagger Definitions for SWI Product Line

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIProductLine
```

SWI Eligibility Compatibility Matrix

Use this REST API service to perform basic operations such as create, update, and query the eligibility and compatibility matrices.

SWI Eligibility Compatibility Matrix Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
InsertOrUpdate	Creates or updates an eligibility or compatibility matrix.
Update	Updates an existing eligibility or compatibility matrix.
Synchronize	Synchronizes an eligibility or compatibility matrix.
QueryById	Queries for an existing eligibility or compatibility matrix using the Row ID.

Name	Description
QueryByExample	Queries for an existing eligibility or compatibility matrix using the QueryByExample method.
Delete	Deletes an eligibility or compatibility matrix.
Insert	Creates an eligibility or compatibility matrix.

Message Description: SWI Eligibility Compatibility Matrix - InsertOrUpdate

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Adjustment Group	An integration component (XML tag) corresponding to the Adjustment Group business component. You can create multiple instances of Adjustment Group under ListOfBase Adjustment Group.	Integration Component
Name	Name of the eligibility or compatibility matrix	String
Adjustment Group Type E&C	Start date for an eligibility or compatibility matrix.	String
Start Date	Start date for an eligibility or compatibility matrix	String
End Date	End date for an eligibility or compatibility matrix	String
ListOfProduct Eligibility	List of eligibility rules.	Integration component
Type	Type of eligibility rule.	String
Matrix Rule Num	Matrix rule number for eligibility rule.	String
ListOfProduct Compatibility	Contains list of compatibility rules.	Integration Component
Type	Type of compatibility rule.	String
Matrix Rule Num	Matrix rule number for compatibility rule.	String
Adjustment Group	Adjustment group name.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Name	Name of the eligibility or compatibility matrix	String
Adjustment Group Type E&C	Start date for an eligibility or compatibility matrix.	String
Start Date	Start date for an eligibility or compatibility matrix	String
End Date	End date for an eligibility or compatibility matrix	String
ListOfProduct Eligibility	List of eligibility rules.	Integration Component
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component

Message Description: SWI Eligibility Compatibility Matrix - Update

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Adjustment Group	An integration component (XML tag) corresponding to the Adjustment Group business component. You can create multiple instances of Adjustment Group under ListOfBase Adjustment Group.	Integration Component
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Start date for an eligibility or compatibility matrix.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix.	String
ListOfProduct Eligibility	List of eligibility rules.	Integration Component
Type	Type of eligibility rule.	String

Node	Description	Type
Matrix Rule Num	Matrix rule number for eligibility rule.	String
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component
Type	Type of compatibility rule.	String
Matrix Rule Num	Matrix rule number for compatibility rule.	String
Adjustment Group	Adjustment group name.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Start date for an eligibility or compatibility matrix.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix	String
ListOfProduct Eligibility	List of eligibility rules.	Integration Component
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component

Message Description: SWI Eligibility Compatibility Matrix - Synchronize

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Adjustment Group	An integration component (XML tag) corresponding to the Adjustment Group business component. You can create multiple instances of Adjustment Group under ListOfBase Adjustment Group.	Integration Component

Node	Description	Type
Name	Name of the eligibility or compatibility matrix .	String
Adjustment Group Type E&C	Type of eligibility or compatibility rules.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix .	String
ListOfProduct Eligibility	Container element for a list of eligibility rules.	Integration Component
Type	Type of eligibility rule.	String
Matrix Rule Num	Matrix rule number for eligibility rule.	String
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component
Type	Type of compatibility rule.	String
Matrix Rule Num	Matrix rule number for compatibility rule.	String
Adjustment Group	Adjustment group name.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Type of eligibility or compatibility rules.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix.	String
ListOfProduct Eligibility	Container element for a list of eligibility rules.	Integration Component
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component

Message Description: SWI Eligibility Compatibility Matrix - QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The row ID of the eligibility or compatibility matrix.	String
OutputIntObjectName	Name of the output integration object.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Start date for an eligibility or compatibility matrix.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix.	String
ListOfProduct Eligibility	List of eligibility rules.	Integration Component
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component

Message Description: SWI Eligibility Compatibility Matrix - QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object ListOfBase Adjustment Group.	Integration Object
Adjustment Group	Integration component corresponding to the Adjustment Group.	Integration Component
Name	Name of the eligibility or compatibility matrix to be queried.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Start date for an eligibility or compatibility matrix.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix.	String
ListOfProduct Eligibility	List of eligibility rules.	Integration Component
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component

Message Description: SWI Eligibility Compatibility Matrix - Delete

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object ListOfBase Adjustment Group.	Integration Object
Name	Name of the eligibility matrix or compatibility matrix.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Type of eligibility or compatibility rules.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix.	String

Node	Description	Type

Message Description: SWI Eligibility Compatibility Matrix - Insert

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object
Adjustment Group	An integration component (XML tag) corresponding to the Adjustment Group business component. You can create multiple instances of Adjustment Group under ListOfBase Adjustment Group.	Integration Component
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Type of eligibility or compatibility rules.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix.	String
ListOfProduct Eligibility	List of eligibility rules.	Integration Component
Type	Type of eligibility rule.	String
Matrix Rule Num	Matrix rule number for eligibility rule.	String
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component
Type	Type of compatibility rule.	String
Matrix Rule Num	Matrix rule number for compatibility rule.	String
Adjustment Group	Adjustment group name.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfBase Adjustment Group	An instance of the integration object Base Adjustment Group.	Integration Object

Node	Description	Type
Name	Name of the eligibility or compatibility matrix.	String
Adjustment Group Type E&C	Type of eligibility or compatibility rules.	String
Start Date	Start date for an eligibility or compatibility matrix.	String
End Date	End date for an eligibility or compatibility matrix.	String
ListOfProduct Eligibility	List of eligibility rules.	Integration Component
ListOfProduct Compatibility	Container element for a list of compatibility rules.	Integration Component

Swagger Definitions for SWI Eligibility Compatibility Matrix

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIEligibilityCompatibilityMatrix
```

SWI Product Base

Use this REST API service to perform queries on the Base Internal Product Integration object, using the QueryPage method.

SWI Product Base Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
QueryPage	Queries for existing products using the QueryPage method.

Message Description: SWI Product Base - QueryPage

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PageSize	Default is 10. Indicates the maximum number of integration object instances to be returned.	String

Node	Description	Type
StartRowNum	Default is 0 (first page). Indicates the row in the result set for the QueryPage method to start retrieving a page of records.	String
ViewMode	Default is All. You must apply Visibility mode to the business object. Valid values are: • Manager • Sales Rep • Personal • Organization • Sub-Organization • Group • Catalog • All **Note:** The ViewMode user property on the integration object has priority over the ViewMode method argument.	String
SiebelMessage	The input or the output integration object instance.	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned.	String
ListOfInternal Product	An instance of the integration object Base Internal Product.	Integration Component

Swagger Definitions for SWI Product Base

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIProductBase
```

SWI Promotion Import

Use this inbound REST API service to create and update promotions.

SWI Promotion Import Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
Upsert	Inserts or updates an existing promotion.

Message Description: SWI Promotion Import - Upsert

For a description of the most important fields in the request and response sections, see the following table.

Node	Description	Type
SyncChild	The upsert operation behaves like a synchronize operation if the SyncChild flag is set to Y. The default value of this flag is Y.	Boolean
ListofSWIPromotionIntegrationIO	Required. An instance of the integration object SWIPromotionIntegration.	Integration Object
Workspace Name	Workspace name.	String
Workspace Reuse Flag	Indicates if a workspace is reused.	Boolean
ListOfSWI Promotion VBC	Container element for Promotion details, which is an instance of the SWI Promotion VBC.	Integration Component
Active Flag	Indicates if a record is active.	Boolean
Name	Promotion name.	String
Organization	Organization name.	String
Type	Promotion type.	String
Track As Agreement Flag	Indicates if a record is tracked as an agreement.	Boolean
Track As Asset Flag	Indicates if a record is tracked as an asset.	Boolean

Node	Description	Type
Release Flag	Indicates if a record is to be released.	Boolean
ListOfSWI Promotion Product Components VBC	Container element for Promotion Product Component details, which is an instance of the SWI Promotion Product Components VBC.	Integration Component
Default Cardinality	Default value for product components.	Integer
Max Cardinality	Maximum number of product components.	Integer
Min Cardinality	Minimum number of product components.	Integer
Product Name	Product name.	String
Disconnect Action	The disconnect action, such as Prompt.	String
Apply Component Charge Flag	Indicates if a component charge is to be applied.	Boolean
Grace Period	Grace period number.	Integer
Grace Period UOM	Grace period unit of measure, such as days or weeks.	String
Prorate Plan Name	Name of the prorate plan.	String
Termination Charge	Termination charge.	String
ListOfSWI Promotion Pricing Components VBC	Container element for Promotion Pricing Component details, which is an instance of the SWI Promotion Pricing Components VBC.	Integration Component
Adjustment Type	Adjustment type.	String
Adjustment Value	Adjustment type value.	Integer
Currency Code	Currency code.	String
End Date	Adjustment end date.	Date
Exchange Date	Exchange date.	Date
Product Name	Product name.	String

Node	Description	Type
ListOfSWI Promotion Aggregate Product Class VBC	Container element for Promotion Aggregate Product Class details which is an instance of the SWI Promotion Aggregate Product Class VBC.	Integration Component
Class Id	The class ID value.	String
Class Name	The class name.	String
Default Cardinality	The default value.	Integer
Description	Description.	String
Disconnect Action	Disconnect action.	String
IsRecommendable Flag	Indicates if a product class can be recommended by users.	Boolean
Max Cardinality	Maximum number.	Integer
Min Cardinality	Minimum number.	Integer
ListOfSWI Promotion Aggregate Product Class - Default Products VBC	Container element for Promotion Aggregate Product Class Default Product details which is an instance of the SWI Promotion Aggregate Product Class - Default Products VBC	Integration Component
Default Cardinality	The default cardinality.	Integer
Product Id	The product ID.	String
Product Name	The product name.	String
ListOfSWI Promotion Aggregate Product Line VBC	Container element for Promotion Aggregate Product Line details which is an instance of the SWI Promotion Aggregate Product Line VBC.	Integration Component
Default Cardinality	Default value.	Integer
Description	Aggregate product line description.	String
Max Cardinality	Maximum number.	Integer
Min Cardinality	Minimum number.	Integer

Node	Description	Type
Product Line	The product line.	String
Disconnect Action	Disconnect action.	String
Product Line Id	Product line ID.	String
ListOfSWI Promotion Aggregate Product Line - Default Products VBC	Container element for Promotion Aggregate Product Line Default Products details which is an instance of the SWI Promotion Aggregate Product Line - Default Products VBC	Integration Component
Default Cardinality	Default cardinality.	Integer
Product Id	Product ID.	String
Product Name	Product name.	String
ListOfSWI Promotion Aggregate Product Class Messages VBC	Container element for Promotion Aggregate Product Class Default Product details which is an instance of the SWI Promotion Aggregate Product Class Messages VBC	Integration Component
Description	Description	String
Language Code	Language code.	String
ListOfSWI Promotion Upgrade Aggregate Product Line VBC	Container element for Promotion Upgrade Aggregate Product Line details which is an instance of the SWI Promotion Upgrade Aggregate Product Line VBC.	Integration Component
Display Commitment	Indicates whether or not to display the commitment.	Boolean
Display Prorated Charge	Indicates whether or not to display the prorated charge.	Boolean
Penalty Amount	Penalty amount.	String
Product Line	Product line.	String
Prorate Plan Name	Prorate plan name.	String
Reason	Commitment reason.	String
ListOfSWI Promotion Upgrade Aggregate Product Class VBC	Container element for Promotion Upgrade Aggregate Product Class details which is an instance of the SWI Promotion Upgrade Aggregate Product Class VBC	Integration Component

Node	Description	Type
Display Commitment	Indicates whether or not to display the commitment.	Boolean
Display Prorated Charge	Indicates whether or not to display the prorated charge.	Boolean
Penalty Amount	Penalty Amount	String
Product Line	Product line.	String
Prorate Plan Name	Prorate plan name.	String
Reason	Reason	String
ListOfSWI Promotion Upgrade To Aggregate Product Line VBC	Container element for Promotion Upgrade To Aggregate Product Line details which is an instance of the SWI Promotion Upgrade To Aggregate Product Line VBC	Integration Component
Display Commitment	Indicates whether or not to display the commitment.	Boolean
Display Prorated Charge	Indicates whether or not to display the prorated charge.	Boolean
Penalty Amount	Penalty amount.	String
Product Line	Product line name.	String
Prorate Plan Name	Prorate plan name.	String
Reason	Reason.	String
ListOfSWI Promotion Upgrade To Aggregate Product Class VBC	Container element for Promotion Upgrade To Aggregate Product Class details which is an instance of the SWI Promotion Upgrade To Aggregate Product Class VBC.	Integration Component
Display Commitment	Indicates whether or not to display the commitment.	Boolean
Display Prorated Charge	Indicates whether or not to display the prorated charge.	Boolean
Penalty Amount	Penalty amount.	String
Product Class Name	Product class name.	String
Prorate Plan Name	Prorate plan name.	String
Reason	Reason.	String

Node	Description	Type
ListOfSWI Promotion Charges_Credits VBC	Container element for the Promotion Charge and Credit details which is an instance of the SWI Promotion Charges_Credits VBC.	Integration Component
Charge Amount	Charge amount.	Integer
Type	Credit type.	String
ListOfSWI Promotion Product Override VBC	Container element for Promotion Product Override details which is an instance of the SWI Promotion Product Override VBC	Integration Component
Component Path	Component path.	String
Promotion Name	Promotion name.	String
Apply Component Charge Flag	Indicates whether or not to apply a component charge.	Boolean
Grace Period	Grace period value.	Integer
Grace Period UOM	Grace period unit of measure, such as days, weeks and so on.	String
Prorate Plan Name	Prorate plan name.	String
Termination Charge	Termination charge.	Integer
Default Cardinality	Default cardinality.	Integer
Max Cardinality	Optional. Maximum cardinality.	Integer
Min Cardinality	Optional Minimum cardinality.	Integer
ListOfSWI Promotion Pricing Override VBC	Container element for Promotion Pricing Override details which is an instance of the OfSWI Promotion Pricing Override VBC.	Integration Component
Adjustment Type	Adjustment type.	String
Adjustment Value	Adjustment value.	Integer
Component Path	Component path.	String

Node	Description	Type
End Date	End date.	Date
Maximum Price	Maximum price.	Integer
Minimum Price	Minimum price.	Integer
Promotion Name	Promotion name.	String
Start Date	Start date.	Date

Swagger Definitions for SWI Promotion Import

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWIPromotionImport
```

SWI Cfg Object Broker

Use this inbound REST API service to query a complete set of versioned objects with focus on the Object Id, External Integration Id, and name mapping.

SWI Cfg Object Broker Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
GetConfigObjectSet	Used to query product Id, product name, attribute, class Id, and class name present in the structure for a given product. This API returns a complete set of versioned products and classes that satisfies the data referential integrity in denormalized format.

Message Description: SWI Cfg Object Broker – GetConfigObjectSet

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
RootId	Queries for an existing product using Row ID.	String
RootName	Queries for an existing product using Product Name.	String

Node	Description	Type
Org	When RootName is used as a search criteria, Org must be provided.	String
Version	Can be used with RootName as search criteria. Valid values are: • "0" - returns workspace • "" - returns last released • other - treated as ""	String
Vendor	RootName and Vendor can be used as search criteria.	String
Full	If this flag is set to N, then only root product will be included in the response. The default value of this flag is Y.	String
ExternalIntegrationId	Queries for an existing product using External Integration Id.	String
FilterObjectAttr	Valid values are: • "Y" - XA attribute information for all objects viz class and products is filtered from response • "N" - returns attribute info for all objects • "" or other - treated as "N"	String
FilterProductAttr	Valid values are: • "Y" - XA attribute information for all products is filtered from response • "N" - returns attribute info for all objects • "" or other - treated as "N"	String
BasicInfoOnly	Valid values are: • "Y" - filters port cardinality and local strings in the object • "N" - keeps port cardinality and local strings • "" or other - treated as "N"	String

For a description of the important fields in the response section, see the following table.

Node	Description	Type
ProdStruct	A hierarchical structure that holds all product/class data.	Hierarchy

Swagger Definitions for SWI Cfg Object Broker

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\SWICfgObjectBroker
```

Literature Web Service

Use this REST API service to perform basic operations such as create, update, and query literatures.

Literature Web Service Operations

For a list of operations associated with this REST API service, see the following table.

Name	Description
InsertOrUpdate	Creates literature or update existing literature.
Update	Updates an existing literature.
Insert	Creates literature.
Delete	Deletes literature.
QueryById	Queries for literature using Row ID.
QueryByExample	Queries for literature using the QueryPage method.

Message Description: Literature Web Service - InsertOrUpdate

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String

Node	Description	Type
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date when the literature file was modified.	String
LitFileName	Literature file name.	String
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author.	String
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object

Node	Description	Type
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date when the literature file was modified.	String
LitFileName	Literature file name.	String
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author.	String
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

Message Description: Literature Web Service - Update

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management)	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date when the literature file was modified.	String
LitFileName	Literature file name.	String
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author.	String

Node	Description	Type
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date when the literature file was modified.	String
LitFileName	Literature file name.	String
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String

Node	Description	Type
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author.	String
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

Message Description: Literature Web Service - Insert

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date at which the literature file was modified.	String
LitFileName	Literature file name.	String

Node	Description	Type
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author	String
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String

Node	Description	Type
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date when the literature file was modified.	String
LitFileName	Literature file name.	String
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author.	String
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

Message Description: Literature Web Service - Delete

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String

Message Description: Literature Web Service – QueryById

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PrimaryRowId	The row ID of the literature.	String
OutputIntObjectName	Name of the output integration object.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String

Node	Description	Type
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date when the literature file was modified.	String
LitFileName	Literature file name.	String
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author.	String
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

Message Description: Literature Web Service – QueryByExample

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management)	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfAdmin Sales Tool (Content Management)	An instance of the integration object Admin Sales Tool (Content Management).	Integration Object
Admin Sales Tool	An integration component (XML tag) corresponding to the Admin Sales Tool business component. You can create multiple instances of Admin Sales Tool under ListOfAdmin Sales Tool (Content Management).	Integration Component
Name	Name of the literature record.	String
Description	Enter a description of the document. Because literature items can be in any language, you may want to indicate the language of the item in this field (or in the Name field).	String
Sales Tool Type	Select the type of the literature. You can create new literature types in the List of Values Administration view. The Literature Type field is used to filter literature items in various applets.	String
LitFileSize	The size of the literature file.	String
LitFileExt	The file extension of the literature file, for example, .doc or .pdf.	String
LitFileDate	The date when the literature file was modified.	String
LitFileName	Literature file name.	String
LitFileSrcType	Source of the Literature records, for example, a file or URL.	String
LitFileSrcPath	Source path of the literature record.	String
Abstract	Optionally, enter a synopsis of the document's content.	String
Primary Organization Name	Name of the primary organization of literature record.	String
Author	Name of the document's author.	String
ListOfAdmin Sales Tool_Internal Product	Select any products associated with the literature. Use this section to allow the user to display the literature under a product.	Integration Component
ListOfAdmin Sales Tool_Organization	Select organizations that will have visibility to this data.	Integration Component

Swagger Definitions for Literature Web Service

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\LiteratureWebService
```

Returning a Promotion Definition by ID

When you use the Siebel REST API to return the entire structure of a specific promotion by querying by ID, the Siebel application returns the entire promotion definition or structure, including relationship and component hierarchy details, the pricing for components and aggregate, and the promotion commitment terms for the given promotion. This REST API returns the promotion definition name, as defined using promotion designer in the Product Promotions view in Administration - Product.

The following request returns the entire structure of a specific promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"ProcessName": "ISS Promotion WS - GetProductPromotionDetails",
"ProdPromId": "88-3N1ZN",
"LoadPromDefFlag": "Y",
"LoadStructFlag": "Full",
"LoadAllDomainFlag": "Y",
"LoadDefaultDomainFlag": "N",
"ProdPromRuleType": "Product"
"Context": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Catalog Context",
"MessageType": "Integration Object",
"ListOfPDS Catalog Context": {
"Context": {}
}
}
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	The following process is required: • ISS Promotion WS - GetProductPromotionDetails
ProdPromId	Required.

Name	Description
	Row ID of the promotion.
LoadPromDefFlag	Optional. The default value is Y. Returns the promotion definition structure.
LoadStructFlag	Optional. There is no default value. Valid values are Full, Modified or no value. • If set to Full, then it returns all the promotion rules. • If set to Modified, then it returns all the promotion rules that are different from the product definition. • When no value is specified and is empty, and ProdPromRuleId is passed, then it only returns the rule with the ID that is passed.
LoadAllDomainFlag	Optional. The default value is N. If the value is Y, then it returns the structure of the underlying product.
LoadDefaultDomainFlag	Optional. The default value is Y. Returns the default product under the promotion aggregate relationship.
ProdPromRuleType	Optional. The default value is Product. Valid values are Product and Pricing.
ProdPromRuleId	Required when no value is set for LoadStructFlag field, or when the LoadDefaultDomainFlag value is Y.
Context	Optional. However required if LoadDefaultDomainFlag is Y. Filters the promotion aggregate relationship based on the ProdPromRuleId and the ProductSearchSpec in the context.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"Error Code": "",
"Error Message": "",
"DomainProductDetail": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-23C8R",
"IntObjectName": "SWI Admin ISS Product Definition",
"MessageType": "Integration Object",
"SWI Internal Product VOD": [
{
"Product Type Code": "Product",
"Product Def Type Code": "Customizable",
"Name": "ACR722GetPromoWS_CP1",
```

```
"Price Type": "One-Time",
"Integration Id": "88-3MERG",
"Id": "88-3MERG",
"SWI ISS Product VOD BusComp": {
"VOD Id": "88-3MERG",
"VOD Name": "ACR722GetPromoWS_CP1",
"VOD Type": "ISS_PROD_DEF",
"SWI ISS Product Details BusComp": {
"Orig Id": "88-3MERO",
"Object Id": "88-3MERM",
"Pre Pick CD": "",
"Parametric Search Flg": "",
"Last Version": "0"
},
"SWI ISS Product Structure Admin BusComp": {
"Prod Id": "88-3MEQB",
"Object Id": "88-3MERM",
"Item Name": "SP1",
"Type": "Product",
"Class Name": "",
"Orig Id": "88-3MERQ",
"Product Name": "ACR722GetPromoWS_SP1",
"Sequence Num": ""
},
"SWI ISS Product Versions BusComp": {
"Version": "2",
"VOD Id": "88-3MERM",
"Version Id": "88-3MERN"
}
},
"SWI Internal Product Organization": {
"Product Id": "88-3MERG",
"Organization Id": "0-R9NH"
}
},
{
"Parent Internal Product Name": "",
"Image Primary Organization Name": "",
"Product Type Code": "Product",
"Product Def Type Code": "Customizable",
"VOD Id": "88-3MESC",
"Special Rating Max Items": "",
"Sequence Number": "",
"Part #": "",
"Fulfillment Criteria Code": "Default",
"Auto Allocate Flag": "Y",
"Name": "ACR722GetPromoWS_CP2",
"SWI ISS Product VOD BusComp": {
"VOD Id": "88-3MES6",
"VOD Name": "ACR722GetPromoWS_CP2",
"VOD Type": "ISS_PROD_DEF",
"SWI ISS Product Details BusComp": {
"Parent VOD Id": "",
"Orig Id": "88-3MESE",
"SPN Definition Id": "",
"Object Id": "88-3MESC"
},
"SWI ISS Product Structure Admin BusComp": [
{
"Prod Id": "88-3MEQW",
"Object Id": "88-3MESC",
"Item Name": "SP2",
"Min Cardinality": "1",
"Type": "Product",
"Object Number": "88-3MEQW",
"Orig Id": "88-3MESI",
```

```
"Product Name": "ACR722GetPromoWS_SP2"
},
{
"Prod Id": "88-3MEQB",
"Max Cardinality": "1",
"Object Id": "88-3MESC",
"Item Name": "SP1",
"Type": "Product",
"Forcastable Flag": "Y",
"Orig Id": "88-3MESG",
"Product Name": "ACR722GetPromoWS_SP1"
}
],
"SWI ISS Product Versions BusComp": {
"VOD Id": "88-3MESC",
"Latest Released Flag": "N"
}
},
"SWI Internal Product Organization": {
"Product Id": "88-3MES6",
"Organization Id": "0-R9NH"
}
},
{
"Product Type Code": "Product",
"Product Def Type Code": "Simple",
"VOD Id": "88-3MER2",
"Name": "ACR722GetPromoWS_SP2",
"Primary Organization Id": "0-R9NH",
"Integration Id": "88-3MEQW",
"Id": "88-3MEQW",
"Dynamic Discount Method": "Amount",
"SWI ISS Product VOD BusComp": {
"VOD Id": "88-3MEQW",
"VOD Name": "ACR722GetPromoWS_SP2",
"SWI ISS Product Details BusComp": {
"Orig Id": "88-3MER4",
"Object Name": "0",
"Object Id": "88-3MER2"
},
"SWI ISS Product Versions BusComp": {
"Version": "2",
"VOD Id": "88-3MER2",
"Version Id": "88-3MER3"
}
},
"SWI Internal Product Organization": {
"Product Id": "88-3MEQW",
"Organization Id": "0-R9NH"
}
}
]
},
"ProdPromDefinition": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-23C8Q",
"IntObjectName": "ISS Promotion",
"MessageType": "Integration Object",
"ISS Promotion": {
"Configuration Model Id": "88-3N1ZN",
"Name": "ACR722GetProm3",
"Promotion Type": "Bundled Promotions",
"Promotion Id": "88-3N1ZN",
"ISS Promotion Items for Import": [
{
"XA Modified Flag": "N",
```

```
"Prod Id": "88-3MES6",
"Prod Cfg Model Id": "88-3MES6",
"Integration Id": "88-3N1ZW",
"Promotion Id": "88-3N1ZN",
"Maximum Quantity": "1",
"Default Quantity": "1",
"Product Name": "ACR722GetPromoWS_CP2"
},
{
"Prod Id": "88-3MEQW",
"Prod Cfg Model Id": "88-3MEQW",
"Integration Id": "88-3N200",
"Promotion Id": "88-3N1ZN",
"Disconnect Action": "Modify",
"Grace Period UOM": "Days",
"Product Name": "ACR722GetPromoWS_SP2"
},
{
"Prod Id": "88-3MERG",
"Integration Id": "88-3N202",
"Maximum Quantity": "1",
"Default Quantity": "1",
"Product Name": "ACR722GetPromoWS_CP1"
}
],
"ISS Promotion Pricing Rules Summary": [
{
"Adjustment Type": "% Discount",
"Prod Cfg Model Id": "88-3MES6",
"Matrix Rule Num": "296-6114527",
"Promotion Id": "88-3N1ZN",
"Product Id": "88-3MES6",
"Product Name": "ACR722GetPromoWS_CP2"
}
],
"ISS Promotion Pricing Components": [
{
"Matrix Rule Num": "296-6114527",
"Product Id": "88-3MES6",
"Product Name": "ACR722GetPromoWS_CP2",
"Product Promotions Relationship VBC": [
{
"Class Name": "",
"Min Cardinality": "",
"Parent Product Name": "ACR722GetPromoWS_CP2",
"Promotion Id": "88-3N1ZN",
"Product Name": "ACR722GetPromoWS_SP1",
"Prod Id": "88-3MEQB"
}
]
},
{
"Promotion Id": "88-3N1ZN",
"Product Id": "88-3MEQW",
"Product Name": "ACR722GetPromoWS_SP2"
},
{
"Promotion Id": "88-3N1ZN",
"Product Id": "88-3MERG",
"Product Name": "ACR722GetPromoWS_CP1",
"Product Promotions Relationship VBC": {
"Parent Product Name": "ACR722GetPromoWS_CP1",
"Parent Product Id": "88-3MERG",
"Promotion Id": "88-3N1ZN",
"Relationship Name": "SP1",
"Product Name": "ACR722GetPromoWS_SP1",
```

```
"Prod Id": "88-3MEQB"
}
}
]
}
},
"PromDomainProduct": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Product Interface",
"MessageType": "Integration Object",
"Product": {}
}
}
```

Returning a Promotion Definition by Name

When you use the Siebel REST API to return the entire structure of a specific promotion by querying by name, the Siebel application returns the entire promotion definition or structure, including relationship and component hierarchy details, the pricing for components and aggregate, and the promotion commitment terms for the given promotion. This REST API returns the promotion definition defined using promotion designer in the Product Promotions view in Administration - Product.

The following request returns the entire structure of a specific promotion:

- **URI: http://ServerName:port/siebel/v1.0/service/Workflow Process Manager/RunProcess**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"ProcessName": "ISS Promotion WS - GetProductPromotionDetails QBE",
"LoadPromDefFlag": "Y",
"LoadStructFlag": "Full",
"LoadAllDomainFlag": "Y",
"LoadDefaultDomainFlag": "N",
"ProdPromRuleType": "Product",
"SiebelMessage": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "ISS Promotion WS",
"MessageType": "Integration Object",
"ListOfISS Promotion WS": {
"ISS Promotion": {
"Name": "SDX Basic"
}
}
}
```

For a description of the request message associated with this API, see the following table.

Name	Description
Process Name	The following process is required: • ISS Promotion WS - GetProductPromotionDetails QBE
LoadPromDefFlag	Optional. The default value is Y.
LoadStructFlag	Optional. There is no default value. Valid values are Full, Modified or no value. • If set to Full, then it returns all the promotion rules. • If set to Modified, then it returns all the promotion rules that are different from the product definition. • When no value is specified and is empty, and ProdPromRuleId is passed, then it only returns the rule with the ID that is passed.
LoadAllDomainFlag	Optional. The default value is N. If the value is Y, then it returns the structure of the underlying product.
LoadDefaultDomainFlag	Optional. The default value is Y. Returns the default product under the promotion aggregate relationship.
ProdPromRuleType	Optional. The default value is Product. Valid values are Product and Pricing.
SiebelMessage	Required. A Siebel Message is a message containing the data of individual integration object instances. An Instance of “ISS Promotion WS” Integration Object is passed to query the promotion by name.

Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
"Error Code": "",
"Error Message": "",
"DomainProductDetail": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-2ABAH",
"IntObjectName": "SWI Admin ISS Product Definition",
"MessageType": "Integration Object",
"ListOfSWI Admin ISS Product Definition": {
"SWI Internal Product VOD": [
{
"Parent Internal Product Name": "",
"Image Primary Organization Name": "",
"Product Type Code": "Product",
"Territory Item Type": "",
"Price Book Currency Code": "",
"Sales Product Flag": "Y",
"Configuration Model": "",
"ImageFileSDX BasickReqFlg": "",
"Image Name": "",
```

```
"Configuration": "N",
"Terms": "",
"Standard Cost": "",
"Case Pack": "",
"Carrier": "",
"Unit of Measure": "",
"ThumbnImageFileSrcType": "",
"Configuration File": "",
"List Price Type Code": "",
"Thumbnail Image Description": "",
"Inclusive Eligibility Flag": "N",
"ThumbnImageFileSrcPath": "",
"Price List Item Id": "",
"Billing Type": "",
"Prod Eff Start Date": "",
"Tax Subcomponent Flag": "N",
"Service Type": "",
"Global Product Identifier": "",
"Product Def Type Code": "Customizable",
"VOD Id": "88-1X5VSA",
"Product Id": "",
"Tool Flag": "",
"Promotional Price": "",
"Override Rollup Flag": "",
"ThumbnImageFileSDX BasickReqFlg": "",
"Vendor Primary Organization Id": "",
"Volume Discount Method": "",
"Target Version Flag": "N",
"Parent Internal Product Part #": "",
"Version": "",
"Parent Internal Product Vendor": "",
"Start Date": "",
"Special Rating Max Items": "",
"Sequence Number": "",
"Part #": "",
"Fulfillment Criteria Code": "Default",
"Ext OT Pct Increase": "",
"Cost List Item Currency Code": "",
"Product Image Integration Id": "",
"CDA Project": "",
"ThumbnImageFileSize": "",
"Unit Conversion Factor": "",
"International Build": "",
"Units In Inventory": "",
"Inventory Flag": "",
"ImageFileSrcType": "",
"Type": "",
"Product Type": "",
"Vendor Part Number": "",
"Build": "",
"Defective Product State": "",
"Pricing Method": "",
"ImageFileSrcPath": "",
"ThumbnImageFileName": "",
"Vendor": "",
"Phone Number Flag": "N",
"Fulfillment Item Code": "",
"Parent Internal Product Id (Joined)": "",
"Next Cost": "",
"Average Cost": "",
"Lead Time": "",
"Maximum Quantity": "",
"Service Instance": "N",
"Parent Internal Product Primary Organization Id": "",
"Parent Internal Product Vendor Organization Id": "",
"Auto Allocate Flag": "Y",
```

```
"Global Product Classification Code": "",
"Cost": "",
"Product Cost": "",
"MTBF": "",
"Pricing Commit Type": "Dynamic",
"Allocate Below Safety Flag": "Y",
"List Price": "",
"Primary Product Line Id": "No Match Row Id",
"Name": "SDX Data Service",
"ThumbnImageFileRev": "",
"XA Class Id": "",
"External Inventory System Ref": "",
"Product Thumbnail Image Id": "",
"ThumbnImageFileDeferFlg": "",
"Vendor Primary Organization": "",
"Max Sale Price": "",
"Taxable Flag": "N",
"Image Primary Organization Id": "",
"Display Component Price": "",
"Targeted Country": "",
"Bill on First Service": "Y",
"Shipping Method": "",
"Targeted Postal Code": "",
"Price Book Name": "",
"ThumbnImageFileExt": "",
"ImageFileAutoUpdFlg": "",
"XA Class Name": "",
"ImageFileSize": "",
"Product Level": "",
"ThumbnImageFileDate": "",
"Primary Equivalent Product Id": "No Match Row Id",
"ThumbnImageFileSDX BasickStatFlg": "",
"Global Supply Chain Code": "",
"Std OT Pct Increase": "",
"Price List Item Start Date": "",
"Billable Flag": "Y",
"Parent Internal Product Primary Organization": "",
"MSRP": "",
"ImageFileName": "",
"Field Replacable Unit": "",
"CDA Project Id": "",
"Active Flag": "Y",
"Targeted Industry": "",
"Price Book Id": "",
"Auto Substitute Flag": "Y",
"Vendor Id": "",
"Pre Pick CD": "N",
"Last Cost": "",
"Billing Service Type": "",
"Parent Internal Product Vendor Organization": "",
"Volume Discount Id": "",
"Cfg Version": "2",
"Product Image Id": "",
"SAP Division Code": "",
"Compound Flag": "N",
"Version Status": "",
"SPN Definition Type": "",
"Vendor Integration Id": "",
"Parent Internal Product Integration Id": "",
"Unit Size": "",
"Price Type": "One-Time",
"ImageFileRev": "",
"Class Product Code": "Product",
"ImageFileSDX BasickStatFlg": "",
"Special Rating Type": "",
"Targeted Min Age": "",
```

```
"CDA Pageset": "",
"Service Price Percent": "",
"Auto Explode Flag": "N",
"Pricing Details": "",
"Cfg Released Flag": "Y",
"ThumbnImageFileAutoUpdFlg": "",
"Minimum Quantity": "",
"End Date": "",
"Units In Inventory As Of": "",
"Parent Internal Product Id": "",
"Primary Organization Id": "0-R9NH",
"Network Element Type": "",
"Vendor Location": "",
"Compensatable": "Y",
"Success Dependency": "Default",
"Global Assembly Level Code": "",
"ImageFileExt": "",
"Complex Product Price Model Name": "",
"Return if Defective": "",
"Orderable": "Y",
"Price List Item Integration Id": "",
"CDA Pageset Id": "",
"Integration Id": "88-1X5VS9",
"Id": "88-1X5VS9",
"Composition Type": "",
"SPN Definition Name": "",
"Serialized": "",
"Complex Product Price Model Id": "",
"Parent Internal Product Vendor Location": "",
"Product Version": "",
"Parent Internal Product Vendor Id": "",
"Payment Type": "",
"Purchase Price": "",
"Price List Item End Date": "",
"Format": "",
"ImageFileDate": "",
"Ship Flag": "N",
"Position Bill Product Flag": "N",
"Track As Asset Flag": "Y",
"Sales Service Flag": "N",
"Units per Case": "",
"Description": "",
"Detailed Description": "",
"Item Size": "",
"Image Description": "",
"Case Conversion Factor": "",
"MTTR": "",
"Service Terms": "",
"Service Flag": "N",
"External Integration Id": "",
"Volume Discount": "",
"Targeted Max Age": "",
"Object Number": "88-1X5VS9",
"Configuration Model Id": "88-1X5VS9",
"Dynamic Discount Method": "Amount",
"SPN Definition Id": "",
"Global Product Lifecycle Status Code": "",
"Pack Size": "",
"Accrual Rate": "",
"Min Sale Price": "",
"Featured": "N",
"Display Component": "",
"Cfg Model Id": "88-1X5VS9",
"Reference Price": "",
"ImageFileDeferFlg": "",
"ListOfPrice List Item": {
```

```
"Price List Item": [
{
"Purchase Price": "",
"Volume Discount End Date": "",
"Min Sale Price": "",
"Bundle Sequence Id": "",
"Integration Id": "444",
"Standard Overtime Price Percent": "",
"Product Name": "SDX Data Service",
"Average Cost": "",
"Dynamic Matrix Id": "",
"Reference Price": "",
"Last Cost": "",
"Pricing Method": "",
"Volume Discount": "SDX Volume Discount",
"End Date": "",
"Description": "",
"Tax Var4": "",
"Override Rollup Flag": "N",
"List Price Type Code": "STANDARD",
"Tax Var3": "",
"Service Price Percent": "",
"Dynamic Matrix": "",
"Tax Var2": "",
"Tax Var1": "",
"Price List Id": "88-23AQ7",
"Cost List Item Currency Code": "USD",
"Start Date": "09/22/2021 00:00:00",
"Extended Overtime Price Percent": "",
"Product Id": "88-1X5VS9",
"Next Cost": "",
"Product VOD Id": "88-1X5VSA",
"Pricing Procedure": "",
"Price List Currency Code": "USD",
"Original List Price": "1000",
"Volume Discount Method": "TIERED",
"List Price": "1000",
"Bundle Sequence Name": "",
"Bonus Threshold": "",
"Promotional Price": "",
"Price Model End Date": "",
"MSRP Price": "",
"Volume Discount Id": "88-1X5Y5B",
"External Integration Id": "",
"Cost": "",
"Price List Name": "CME Price List",
"Max Sale Price": "",
"Service Price Amount": ""
}
]
},
"ListOfSWI ISS Product VOD BusComp": {
"SWI ISS Product VOD BusComp": [
{
"VOD Id": "88-1X5VS9",
"Locked By": "",
"Locked Flag": "N",
"Description": "",
"VOD Name": "SDX Data Service",
"Locked Date": "",
"VOD Type": "ISS_PROD_DEF",
"ListOfSWI ISS Product Details BusComp": {
"SWI ISS Product Details BusComp": [
{
"First Version": "1",
"Class Name": "",
```

```
"Parent VOD Id": "",
"Orig Id": "88-1X5VSB",
"SPN Definition Id": "",
"Post Pick CD": "",
"Object Name": "0",
"XA Integration Id": "",
"Object Id": "88-1X5VSA",
"Pre Pick CD": "",
"SPN Definition Name": "",
"Parametric Search Flg": "",
"Last Version": "0"
}
]
},
"ListOfSWI ISS Product Structure Admin BusComp": {
"SWI ISS Product Structure Admin BusComp": [
{
"Last Version": "0",
"Prod Id": "88-1X5VNR",
"Max Cardinality": "1",
"Default Object Number": "",
"Object Id": "88-1X5VSA",
"Comments": "",
"Item Name": "SDX Wireless Data",
"Min Cardinality": "0",
"Default Prod Id": "",
"Type": "Product",
"Class Name": "",
"Object Number": "88-1X5VNR",
"Class Id": "",
"Default Cardinality": "0",
"First Version": "1",
"Forcastable Flag": "Y",
"Orig Id": "88-1X5VTW",
"Product Name": "SDX Wireless Data",
"Port Id": "",
"Relationship Type": "Has A",
"Sequence Num": ""
}
]
},
"ListOfSWI ISS Product Versions BusComp": {
"SWI ISS Product Versions BusComp": [
{
"Version": "3",
"Released Flag": "N",
"VOD Id": "88-1X5VSA",
"Latest Released Flag": "N",
"Comments": "",
"Required End Date": "",
"Required Start Date": "",
"Active Flg": "Y",
"Version Id": "88-1X5VSC"
}
]
}
}
]
},
"ListOfSWI Internal Product Organization": {
"SWI Internal Product Organization": [
{
"Product Id": "88-1X5VS9",
"Organization Id": "0-R9NH"
}
]
```

```
}
}
]
}
},
"ProdPromDefinition": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "88-2ABAG",
"IntObjectName": "ISS Promotion WS",
"MessageType": "Integration Object",
"ListOfISS Promotion WS": {
"ISS Promotion": [
{
"Configuration Model Id": "88-1XLIBC",
"Period": "1",
"Usage Plan Name": "",
"ImageFileSDX BasickStatFlg": "",
"NRC Amount": "",
"NRC Amount Exchange Date": "10/12/2021 00:00:00",
"RC Amount Currency Code": "USD",
"Pre Pick CD": "Y",
"ImageFileSDX BasickReqFlg": "",
"Manual Ambiguity Resolution Flag": "N",
"NRC Plan Id": "",
"Id": "88-1XLIBC",
"Commit Flag": "Y",
"Name": "SDX Basic",
"Payment Type": "",
"Sales Product Flag": "Y",
"ThumbnImageFileExt": "",
"Adjust Charge": "",
"Promotion Image Integration Id": "",
"RC Amount Exchange Date": "10/12/2021 00:00:00",
"End Date": "",
"Description": "",
"Instance": "",
"Grace Period": "",
"NRC Schedule": "",
"RC Schedule": "",
"RC Adjustment Amount": "",
"ImageFileRev": "",
"Start Date": "",
"Usage Schedule": "",
"Period UOM": "Years",
"Promotion Type": "Bundled Promotions",
"ThumbnImageFileSDX BasickStatFlg": "",
"ImageFileSize": "",
"Track As Instance Flag": "Y",
"RC Adjustment U/M": "",
"Track As Asset Flag": "Y",
"ImageFileName": "",
"NRC Plan Name": "",
"NRC Schedule Id": "",
"Promotion Id": "88-1XLIBC",
"Position Bill Product Flag": "N",
"ThumbnImageFileSize": "",
"ThumbnImageFileSDX BasickReqFlg": "",
"ImageFileExt": "",
"Usage Plan Id": "",
"Reason": "",
"NRC Qty": "1",
"Score": "",
"Track As Agreement Flag": "Y",
"ThumbnImageFileName": "",
"RC Amount": "",
"NRC Amount Currency Code": "USD",
```

```
"ThumbnImageFileRev": "",
"Grace Period UOM": "Days",
"Adjust Reason": "",
"Image Name": "",
"Product Def Type Code": "None",
"Inclusive Eligibility Flag": "N",
"Usage Schedule Id": "",
"Charge Basis": "Schedule",
"RC Schedule Id": "",
"ListOfISS Promotion Items for Import": {
"ISS Promotion Items for Import": [
{
"Period": "1",
"IsRecommendable Flag": "Y",
"Minimum Quantity": "1",
"Item Name": "",
"Termination Charge": "",
"Id": "88-1XLIBF",
"Prod Id": "88-1X5VS9",
"Product Line": "",
"Apply Component Charge Flag": "N",
"Class VodNum": "",
"Integration Id": "88-1XLIBF",
"XA Modified Flag": "N",
"Root Promotion Item Id": "88-1XLIBF",
"Product Name": "SDX Data Service",
"Is Constrained": "N",
"Prorate Plan Id": "",
"Description": "",
"Default Cardinality": "",
"Sub Obj Name": "",
"Grace Period": "",
"Default Quantity": "1",
"Sequence": "",
"Maximum Quantity": "1",
"Period UOM": "Years",
"Cardinality Modified Flag": "N",
"Product Line Id": "",
"Prod Cfg Model Id": "88-1X5VS9",
"Sub Obj VOD Num": "",
"Parent Promotion Item Integration Id": "",
"Domain Modified Flag": "N",
"Promotion Id": "88-1XLIBC",
"Lower Limit": "",
"Track As Agreement Flag": "N",
"Prorate Plan Name": "",
"Type": "Components",
"Parent Promotion Item Id": "",
"Product Item Id": "",
"Grace Period UOM": "Days",
"Upper Limit": "",
"Disconnect Action": "Disconnect"
},
{
"Period": "1",
"IsRecommendable Flag": "Y",
"Minimum Quantity": "1",
"Item Name": "",
"Termination Charge": "",
"Id": "88-1XLIBH",
"Prod Id": "",
"Product Line": "",
"Apply Component Charge Flag": "N",
"Class VodNum": "88-1X5VEB",
"Integration Id": "88-1XLIBH",
"XA Modified Flag": "N",
```

```
"Root Promotion Item Id": "88-1XLIBH",
"Product Name": "",
"Is Constrained": "N",
"Prorate Plan Id": "",
"Description": "Product Class",
"Default Cardinality": "",
"Sub Obj Name": "",
"Grace Period": "",
"Default Quantity": "1",
"Sequence": "",
"Maximum Quantity": "1",
"Period UOM": "Years",
"Cardinality Modified Flag": "Y",
"Product Line Id": "",
"Prod Cfg Model Id": "",
"Sub Obj VOD Num": "",
"Parent Promotion Item Integration Id": "",
"Domain Modified Flag": "N",
"Promotion Id": "88-1XLIBC",
"Lower Limit": "",
"Track As Agreement Flag": "N",
"Prorate Plan Name": "",
"Type": "Aggregate",
"Parent Promotion Item Id": "",
"Product Item Id": "",
"Grace Period UOM": "Days",
"Upper Limit": "",
"Disconnect Action": "Prompt",
"ISS Promotion Items for Import": [
{
"Period": "1",
"IsRecommendable Flag": "Y",
"Minimum Quantity": "",
"Item Name": "",
"Termination Charge": "",
"Id": "88-1XLIBJ",
"Prod Id": "88-1X5VJB",
"Product Line": "",
"Apply Component Charge Flag": "N",
"Class VodNum": "",
"Integration Id": "88-1XLIBJ",
"XA Modified Flag": "N",
"Root Promotion Item Id": "88-1XLIBJ",
"Product Name": "SDX Mustang 11",
"Is Constrained": "N",
"Prorate Plan Id": "",
"Description": "",
"Default Cardinality": "",
"Sub Obj Name": "",
"Grace Period": "",
"Default Quantity": "1",
"Sequence": "",
"Maximum Quantity": "",
"Period UOM": "Years",
"Cardinality Modified Flag": "N",
"Product Line Id": "",
"Prod Cfg Model Id": "88-1X5VJB",
"Sub Obj VOD Num": "",
"Parent Promotion Item Integration Id": "88-1XLIBH",
"Domain Modified Flag": "N",
"Promotion Id": "88-1XLIBC",
"Lower Limit": "",
"Track As Agreement Flag": "N",
"Prorate Plan Name": "",
"Type": "Aggregate Default",
"Parent Promotion Item Id": "88-1XLIBH",
```

```
"Product Item Id": "",
"Grace Period UOM": "Days",
"Upper Limit": "",
"Disconnect Action": "Modify"
}
]
}
]
},
"ListOfISS Promotion Upgrade To": {
"ISS Promotion Upgrade To": [
{
"Target Period UOM": "Years",
"Original Period": "1",
"Product Line": "",
"Class VodNum": "",
"Original Promotion Description": "",
"Target Promotion VOD Row Id": "88-1X6ET3",
"Prorate Plan Id": "",
"Penalty Amount Currency Code": "USD",
"Tgt End Date": "08/23/2021 10:00:00",
"Original Promotion Id": "88-1XLIBC",
"Original Period UOM": "Years",
"Tgt Start Date": "08/23/2021 10:00:00",
"Target Promotion Id": "88-1X6ET2",
"Penalty Amount Exchange Date": "10/12/2021",
"Tgt Released Flag": "Y",
"Original Grace Period UOM": "Days",
"Target Grace Period UOM": "Days",
"Product Line Id": "",
"Target Period": "1",
"Tgt First Version": "1",
"Class Name": "",
"Prod Cfg Model Id": "88-1XLIBC",
"Tgt Last Version": "0",
"Original Promotion Name": "SDX Basic",
"Commitment Start": "Original Start",
"Penalty Amount": "",
"Target Promotion Description": "",
"Target Promotion Name": "SDX BasicOptional1",
"Reason": "",
"Tgt Active Flag": "Y",
"Tgt Version": "1",
"Prorate Plan Name": "",
"Display Prorated Charge": "Y",
"Original Grace Period": "",
"Display Commitment": "Y",
"Target Promotion Configuration Model Id": "88-1X6ET2",
"Target Grace Period": "",
"Relationship Type": "Upgrade",
"Duration": "Original Duration"
},
{
"Target Period UOM": "Years",
"Original Period": "1",
"Product Line": "",
"Class VodNum": "",
"Original Promotion Description": "",
"Target Promotion VOD Row Id": "88-1X6EUQ",
"Prorate Plan Id": "",
"Penalty Amount Currency Code": "USD",
"Tgt End Date": "08/23/2021 10:00:00",
"Original Promotion Id": "88-1XLIBC",
"Original Period UOM": "Years",
"Tgt Start Date": "08/23/2021 10:00:00",
"Target Promotion Id": "88-1X6EUP",
```

```
"Penalty Amount Exchange Date": "10/12/2021",
"Tgt Released Flag": "Y",
"Original Grace Period UOM": "Days",
"Target Grace Period UOM": "Days",
"Product Line Id": "",
"Target Period": "1",
"Tgt First Version": "1",
"Class Name": "",
"Prod Cfg Model Id": "88-1XLIBC",
"Tgt Last Version": "0",
"Original Promotion Name": "SDX Basic",
"Commitment Start": "Original Start",
"Penalty Amount": "",
"Target Promotion Description": "",
"Target Promotion Name": "SDX Ineligible Prom 4",
"Reason": "",
"Tgt Active Flag": "Y",
"Tgt Version": "1",
"Prorate Plan Name": "",
"Display Prorated Charge": "Y",
"Original Grace Period": "",
"Display Commitment": "Y",
"Target Promotion Configuration Model Id": "88-1X6EUP",
"Target Grace Period": "",
"Relationship Type": "Upgrade",
"Duration": "Original Duration"
},
{
"Target Period UOM": "Years",
"Original Period": "1",
"Product Line": "",
"Class VodNum": "",
"Original Promotion Description": "",
"Target Promotion VOD Row Id": "88-1X6F47",
"Prorate Plan Id": "",
"Penalty Amount Currency Code": "USD",
"Tgt End Date": "09/23/2021 01:47:41",
"Original Promotion Id": "88-1XLIBC",
"Original Period UOM": "Years",
"Tgt Start Date": "09/23/2021 01:44:38",
"Target Promotion Id": "88-1X6F46",
"Penalty Amount Exchange Date": "10/12/2021",
"Tgt Released Flag": "Y",
"Original Grace Period UOM": "Days",
"Target Grace Period UOM": "Days",
"Product Line Id": "",
"Target Period": "1",
"Tgt First Version": "1",
"Class Name": "",
"Prod Cfg Model Id": "88-1XLIBC",
"Tgt Last Version": "0",
"Original Promotion Name": "SDX Basic",
"Commitment Start": "Original Start",
"Penalty Amount": "",
"Target Promotion Description": "",
"Target Promotion Name": "SDX BasicOptional2",
"Reason": "",
"Tgt Active Flag": "Y",
"Tgt Version": "1",
"Prorate Plan Name": "",
"Display Prorated Charge": "Y",
"Original Grace Period": "",
"Display Commitment": "Y",
"Target Promotion Configuration Model Id": "88-1X6F46",
"Target Grace Period": "",
"Relationship Type": "Upgrade",
```

```
"Duration": "Original Duration"
}
]
},
"ListOfProduct Eligibility BusComp": {
"Product Eligibility BusComp": [
{
"Product Class": "",
"Effective End Date": "",
"Product Id": "88-1XLIBC",
"Product Line Id": "",
"Country": "",
"External Integration Id": "",
"Matrix Rule Num": "296-116897025",
"Account Name": "",
"Type": "Not Available",
"Product Line": "",
"Post Code To": "",
"State": "AK",
"Exchange Prefix": "",
"Adjustment Group Id": "88-1X5Y23",
"Post Code From": "",
"Product": "SDX Basic",
"Effective Start Date": "",
"Product Class Id": "",
"Account Type Code": "",
"Account Id": "",
"Eligibility Status": "N",
"City": ""
}
]
}
}
]
}
},
"PromDomainProduct": {
"IntObjectFormat": "Siebel Hierarchical",
"MessageId": "",
"IntObjectName": "PDS Product Interface",
"MessageType": "Integration Object",
"ListOfPDS Product Interface": {
"Product": []
}
}
```


ISS Promotion Definition Loader QP

Use this REST API service to perform queries for existing product promotions using QueryPage method on the ISS Promotion WS integration object.

ISS Promotion Definition Loader QP

For a list of operations associated with this REST API service, see the following table.

Name	Description
QueryPage	Used to query for existing promotions using the QueryPage method.

Message Description: ISS Promotion Definition Loader QP - QueryPage

For a description of the most important fields in the request section, see the following table.

Node	Description	Type
PageSize	Indicates the maximum number of integration object instances to be returned. The default value is 10.	String
StartRowNum	Indicates the row in the result set for the QueryPage method to start retrieving a page of records. The default value is 0 (first page).	String
ViewMode	Default is All. You must apply Visibility mode to the business object. Valid values are: • Manager • Sales Rep • Personal • Organization • Sub-Organization • Group • Catalog • All **Note:** The ViewMode user property on the integration object has priority over the ViewMode method argument.	String
SiebelMessage	The input or the output integration object instance.	Integration Object

For a description of the most important fields in the response section, see the following table.

Node	Description	Type
ListOfISS Promotion WS	An instance of the integration object ISS Promotion WS.	Integration Object

Node	Description	Type
LastPage	Boolean indicating whether or not the last record in the query result set has been returned..	String

Swagger Definitions for ISS Promotion Definition Loader QP

To view the Swagger definitions in JSON format for this REST API service, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

```
...\ISSPromotionDefinitionLoaderQP
```

12 Purging Old Versions Of Product, Class, and Attribute

Purging Old Versions of Product, Class, and Attribute

This chapter describes how product administrators can purge or delete old versions of product, class, and attribute while retaining specified number of versions.

It includes following topics:

- *Query Versions of Product, Class, and Attribute*
- *Removing Old Versions Of Product, Class, and Attribute*

More information about versioned objects is available in *Siebel Order Management Guide*.

Query Versions of Product, Class, and Attribute

Method QueryPage method on SWI VOD Versions business service returns details about versioned objects: product, class, or attribute.

This method supports pagination. Before deleting versions, the caller can use this method to query available versions.

Input Arguments

Name	Value	Comments
Object Name	String	Name of product, class, or attribute
Object Number	String	The product Id if object type is "Product"
VOD Id	String	Class or Attribute Id
Object Type	String	Product Definition, Parent Class Definition, or Attribute Definition

Output Arguments

Name	Comments
Active Flg	Flag denoting the active version
Comments	User entered comments

Name	Comments
Object Name	Name of product, class, or attribute
Object Number	Object Number
Object Type	Product, Class, or Attribute
Parent Class Name	Name of the parent class
Parent Class VOD Id	Parent Class VOD Id
Released Flg	Released Flag
Required End Date	Version end date
Required Start Date	Version start date
VOD Id	VOD Id
Version	Version number

- **URI:** *https://SiebelURL.com/siebel/v1.0/service/SWI%20VOD%20Versions/QueryPage*
- **HTTP Method: POST**
- **Content-Type:** application/json
- **Authorization: BASIC**
- **Request body**

```
"body": {
 "PageSize": "10",
 "StartRowNum": "0",
 "SiebelMessage": {
 "MessageId": "",
 "IntObjectName": "SWI VOD Versions",
 "MessageType": "Integration Object",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfSWI VOD Versions": {
 "SWI VOD Versions BusComp": {
 "Object Name": "Wireless Service"
 }
 }
 }
}
```

Response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body**

```
{
 "LastPage": "true",
 "SiebelMessage": {
 "IntObjectFormat": "Siebel Hierarchical",
 "MessageId": "88-A8ADV",
 "IntObjectName": "SWI VOD Versions",
 "MessageType": "Integration Object",
 "ListOfSWI VOD Versions": {
 "SWI VOD Versions BusComp": [
 {
 "Version": "4",
```

```
"VOD Id": "88-3C4GYD",
"Comments": "",
"Required End Date": "",
"Object Name": "Wireless Service",
"Released Flg": "Y",
"Object Number": "88-3C4GY7",
"Required Start Date": "08/01/2050 00:12:00",
"Parent Class VOD Id": "88-3C4GX2",
"Active Flg": "Y",
"Object Type": "ISS_PROD_DEF",
"Parent Class Name": "Wireless Service Class"
},
{
"Version": "5",
"VOD Id": "88-3C4GYD",
"Comments": "",
"Required End Date": "08/01/2050 00:12:00",
"Object Name": "Wireless Service",
"Released Flg": "Y",
"Object Number": "88-3C4GY7",
"Required Start Date": "11/19/2024 03:09:00",
"Parent Class VOD Id": "88-3C4GX2",
"Active Flg": "Y",
"Object Type": "ISS_PROD_DEF",
"Parent Class Name": "Wireless Service Class"
},
{
"Version": "6",
"VOD Id": "88-3C4GYD",
"Comments": "",
"Required End Date": "",
"Object Name": "Wireless Service",
"Released Flg": "N",
"Object Number": "88-3C4GY7",
"Required Start Date": "",
"Parent Class VOD Id": "88-3C4GX2",
"Active Flg": "Y",
"Object Type": "ISS_PROD_DEF",
"Parent Class Name": "Wireless Service Class"
}
]
}
}
}
```


Removing Old Versions Of Product, Class, and Attribute

Siebel customers who are using Customer Order Management module for long time end up with lot of versions of Product, class, and attribute entities in the product catalog. Existing CleanupSingleObject method on ISS Authoring Import Export Service removes all versions except workspace Version. Customers want to retain recent versions rather than removing all versions.

Method RemoveOldReleasedVersions of Business Service SWI Cfg Object Broker removes (deletes or purges) versions of Product, class, and attribute while retaining specified number of versions.

Purging starts from the smallest version number. Purging stops at current active version or when future dated version is found.

Business service returns how many versions were removed, how many versions are remaining and what is the current active version.

Note: When the optional NoCommitFlag field is set to Yes, data is not committed to the database.

Input Arguments

Name	Value	Comments
ObjType	ISS_PROD_DEF / ISS_CLASS_DEF/ ISS_ATTR_DEF	Required. Object to delete. Product, class, or attribute
ObjNum	String	The VOD Id of the object to purge Either ObjNum or ObjName are required. If both are provided, ObjNum is used.
ObjName	String	Optional. See above. The VOD Name of the object to purge
NumVersToRetain	Integer	Required. Number of versions to retain after deletion
NoCommitFlag	Yes/No	Optional. Default is Yes. Yes: Changes are not committed to database. Returns what's the impact

Output Arguments

Name	Comments
VersRemained	Versions remaining
VersRemoved	Versions removed.
Committed	Flag (Y or N) denoting if the data is committed or not.
CurrentActiveVer	Current active version number
Comments	Returns "Success" if the number of versions deleted is the same as that specified in the input argument, otherwise, it will be "Purging stopped at version {n}, which is the current active version or a future version."
Versions	Versions: A property set that holds the version information, the versions are separated into two sections: Remains and Removed. Information includes: Start Date", "End Date", "Released Flag", "Version", "Active Flag", and "Comments"

- **URI:** *https://SiebelURL.com/siebel/v1.0/service/SWI%20Cfg%20Object%20Broker/RemoveOldReleasedVersions*
- **HTTP Method: POST**
- **Content-Type:** application/json

- **Authorization: BASIC**
- **Request body for Order:**

```
{
 "ObjNum":"88-3BNUEE",
 "ObjType":"ISS_PROD_DEF",
 "NumVersToRetain": "1",
 "NoCommitFlag": "Y"
}
```

Response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**

```
{
 "VersRemained": "3",
 "VersRemoved": "0",
 "Committed": "N",
 "CurrentActiveVer": "3",
 "Comments": "Purging stopped at version 3, which is the current active version or a future version.",
 "Versions": [
 {
 "Remains": [
 {
 "": [
 {
 "Start Date": "09/11/2024 12:15:35",
 "End Date": "09/17/2024 00:00:00",
 "Released Flag": "Y",
 "Version": "3",
 "Active Flag": "Y",
 "Comments": ""
 },
 {
 "Start Date": "09/30/2024 00:00:00",
 "End Date": "",
 "Released Flag": "Y",
 "Version": "4",
 "Active Flag": "Y",
 "Comments": ""
 },
 {
 "Start Date": "09/17/2024 00:00:00",
 "End Date": "09/30/2024 00:00:00",
 "Released Flag": "Y",
 "Version": "5",
 "Active Flag": "Y",
 "Comments": ""
 }
 ]
 }
 ],
 "Deleted": []
 }
 ]
}
```

13 Using Siebel REST API for Pricing Administration

Using Siebel REST API For Siebel Pricing Administration

This chapter describes Siebel Representational State Transfer (REST) Services for Siebel Pricing Administration entities and how to use them. It includes the following topics:

- *Overview of Siebel REST APIs in Pricing Administration*

More information is available in *Siebel CRM Web Services Reference* and *Siebel Order Management Guide* .

To view the Postman collection of REST APIs for Siebel Pricing Administration entities, navigate to the latest documentation library for Oracle Siebel CRM on Oracle Help Center (*https://docs.oracle.com/en/applications/siebel/index.html*), and click the Related Files link next to the link for *Siebel REST API Guide* to access a zip file. Extract the files in the zip file.

Note: This Postman collection requires use of the third-party API platform, Postman™. The data contained in this zip file is provided for knowledge transfer and testing only. You are allowed to update this data as it exists in its current state. For supported versions of third-party software, see the Certifications tab on My Oracle Support.

Overview of Siebel REST APIs in Pricing Administration

This topic provides a list of pricing administration REST APIs for Siebel Customer Order Management. You can use these REST API services to simplify integration with a centralized product catalog. This integration with pricing administration enables Oracle customers to create new product offerings faster, with a centrally-managed product catalog, and as a result reducing time-to-market.

To use pricing administration REST APIs, you need to access Siebel Business Objects, as described in *Using the Siebel REST API to Access Business Objects*. For a description of the business objects, and their respective base integration objects, see the following table.

Business Object	Integration Object
Price List	Base Price List
Attribute Adjustments	Base PSP Dynamic Matrix
Discount Matrices	Base Adjustment Group
Aggregate Discounts	Base Price Bundle Discount

Business Object	Integration Object
Aggregate Discount Sequences	Base Pricer Bundle Sequence
Contract Schedule	Base FS Service Schedule
Conditional Charge Plan	Base FS Conditional Charge
Volume Discount	Base Volume Discount

Note: More information on Siebel REST APIs is also available in *Siebel CRM Web Services Reference* and *Siebel Order Management Guide* .

14 Using Siebel REST API For Siebel Loyalty

Using Siebel REST API for Siebel Loyalty

This chapter describes Siebel Representational State Transfer (REST) Services for Siebel Loyalty users and how to use them. It includes the following topics:

- *Deleting Member Data in Siebel Loyalty*
- *Using Siebel Loyalty Business Components*

More information is also available in *Siebel Loyalty Administration Guide* .

Deleting Member Data in Siebel Loyalty

The data or information that is used to uniquely identify a loyalty member is called personally identifiable information (PII). In Siebel Loyalty, examples of PII are membership-related records, including member child objects like vouchers, transactions, cards, and so on. When you use the Siebel REST API to delete a Siebel loyalty user, the Siebel Loyalty application deletes all fields related to that member, and the deleted member can no longer be located in the Siebel Loyalty application.

The following request deletes a Siebel loyalty user:

- **URI: http://ServerName:port/siebel/v1.0/service/LOY Member Services/AsyncCascadeDelete**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
"body": {
"Member Number": "296-2186601"
}
}
```

Here are the response details for a successful request:

- **HTTP Code:** 200

  Content-Type: JSON
- **Response body:**

```
"Response": "Your request has been submitted for processing. Please check notifications for update."
```

Using Siebel Loyalty Business Components

Siebel CRM REST API services offers the following lightweight business components for Siebel Loyalty users:

- LWA Member

- LWA Transaction

During REST API calls, these business components exclude calculated fields, and seldom-used fields, resulting in quicker response times for API calls performed against loyalty members and transactions.

Processing Accrual Transactions

You use the business service method ProcessAccrualTransaction with the lightweight business component Base LWA Transaction to process an accrual transaction.

The ProcessAccrualTransaction business service method is described below.

- **Business Service.** LOY Accrual Service
- **Business Service Method Arguments:**
  - **Base LWA LOY Transaction.** Integration Object
  - **OrderNumber.** Required. The transaction order number.
  - **RequestMode.** Required. This processes or simulates the transaction. Valid values are Process, or Simulate.
  - **Total Points.** The total points of the accrual transaction.
  - **TotalLines.** Required. The total number of transactions.

This API internally calls the ProcessExternalOrderTxn business service method, which then processes an accrual transaction for the Transaction business component. You can query for a Siebel CRM child business component that returns its child links, that is, links to grandchildren, by sending a HTTP GET request to the resource's URI. For more information, see *Querying for a Siebel CRM Business Component To Return Child Links for Multiple Child Business Components*.

The following request processes transactions for an order:

- **URI: http://ServerName:port/siebel/v1.0/service/LOY Accrual Service/ProcessAccrualTransaction?OrderNumber=1804&TotalLines=2&RequestMode=Process**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**

```
{
 "body": {
 "Base LWA LOY Transaction": {
 "IntObjectName": "Base LWA LOY Transaction",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfBase LWA LOY Transaction": {
 "LWA LOY Transaction": [
 {
 "Member Number": "296-183479719",
 "Transaction Date": "04/01/2022",
 "Order Number": "1804",
 "Partner Id": "88-15JV6",
 "Point Id": "88-19ZR5",
 "Amount": 10,
 "Points": 300,
 "Product Id": "88-1ARHO",
 "Product Name": "MSR Regular Accrual",
 "Program Id": "88-19ZQZ",
 "Qualifying Flag": "Y",
```

```
 "Transaction Sub Type": "Product",
 "Transaction Type": "Accrual"
 },

 {
 "Member Number": "296-183430719",
 "Transaction Date": "04/02/2022",
 "Order Number": "1804",
 "Partner Id": "88-15JV6",
 "Point Id": "88-19ZR5",
 "Amount": 1000,
 "Points": 350,
 "Product Id": "88-1ARHO",
 "Product Name": "MSR Regular Accrual",
 "Program Id": "88-19ZQZ",
 "Qualifying Flag": "Y",
 "Transaction Sub Type": "Product",
 "Transaction Type": "Accrual"
 }
 ]
 }
 }
 }
}
```


Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "Total Points": "650",
 "Base LWA LOY Transaction": {
 "IntObjectFormat": "Siebel Hierarchical",
 "IntObjectName": "Base LWA LOY Transaction",
 "MessageType": "Integration Object",
 "LWA LOY Transaction": [
 {
 "Status": "Processed",
 "Transaction Id": "88-318UDI",
 "Points": "350",
 "Transaction Number": "296-183490758",
 "Sub Status": "Success",
 "Processing Log": "Admin - Accruals\r\n Got Applied. 350 The Qualifying. Regular\r\n Issued Voucher
 296-183490768 Expiry Date: 04/01/2022 05:29:34\r\nNot satisfy the filter.\r\n 1-> Admin - Cancellation;
\r\n 1-> Admin - Gift;\r\n 1-> Admin - Gift Points;\r\n 1-> Admin - Loan;\r\n 1-> Admin - Lounge
 purchase;\r\n 1-> Admin - Membership Renewal;\r\n 1-> Admin - Point Reactivation;\r\n 1-> Admin -
 Points Purchase;\r\n 1-> Admin - Promotion Enrolment;\r\n 1-> Admin - Redemptions;\r\n 1-> Admin -
 Voucher;\r\n 1-> Admin - Voucher Reissue;\r\n 1-> Admin - Point Transfer;\r\n 1-> Batch Redemption;
\r\n 1-> CT Accruals;\r\n 1-> CT Debits;\r\n 1-> Program - Transfer Points;\r\n 1-> Program-Enrolment;
\r\n;Not within start/end dates.\r\n 1-> Special Points based on Product Line;\r\n;No rules were
 satisfied.\r\n 1-> Capture Base for Accrual Reward - Distance (PR-A1);\r\n 1-> Recurring Rewards on
 Primary Member Txn;\r\n;",
 "LOY Transaction Accrual Item": {
 "Accrualed Value": "350",
 "Promotion Name": "Admin - Accruals",
 "Point Block Name": "Indigo Points",
 "Point Sub Type": ""
 }
 },
 {
 "Status": "Processed",
 "Transaction Id": "88-318UDG",
```

```
 "Points": "300",
 "Transaction Number": "296-183490756",
 "Sub Status": "Success",
 "Processing Log": "Admin - Accruals\r\n Got Applied. 300 The Qualifying. Regular\r\n Issued Voucher
 296-183490772 Expiry Date: 04/01/2022 05:29:35\r\nNot satisfy the filter.\r\n 1-> Admin - Cancellation;
\r\n 1-> Admin - Gift;\r\n 1-> Admin - Gift Points;\r\n 1-> Admin - Loan;\r\n 1-> Admin - Lounge
 purchase;\r\n 1-> Admin - Membership Renewal;\r\n 1-> Admin - Point Reactivation;\r\n 1-> Admin -
 Points Purchase;\r\n 1-> Admin - Promotion Enrolment;\r\n 1-> Admin - Redemptions;\r\n 1-> Admin -
 Voucher;\r\n 1-> Admin - Voucher Reissue;\r\n 1-> Admin - Point Transfer;\r\n 1-> Batch Redemption;
\r\n 1-> CT Accruals;\r\n 1-> CT Debits;\r\n 1-> Program - Transfer Points;\r\n 1-> Program-Enrolment;
\r\n;Not within start/end dates.\r\n 1-> Special Points based on Product Line;\r\n;No rules were
 satisfied.\r\n 1-> Capture Base for Accrual Reward - Distance (PR-A1);\r\n 1-> Recurring Rewards on
 Primary Member Txn;\r\n;",
 "LOY Transaction Accrual Item": {
 "Accrualed Value": "300",
 "Promotion Name": "Admin - Accruals",
 "Point Block Name": "Indigo Points",
 "Point Sub Type": ""
 }
 }
 ]
 }
}
```


Processing Redemption Transactions

You use the business service method ProcessRedemptionTransaction with the lightweight business component Base LWA Transaction to process a redemption transaction. This business service method removes the dependency on processing the transaction order number, therefore improving response times.

This API creates and processes a single redemption transaction, however it also handles loans if points are required when a redemption falls short, and the member is eligible for a loan.

The integration object Base LWA LOY Transaction uses the U2 User Key to pass the date and time fields to assess the transaction date. It ensures that users can redeem the same product for a loyalty member at different times on the same day. If the user tries to redeem the same product for a member at the same time of the day then the following error message is displayed:

```
Duplicate Transaction exists for attributes Member Number, Product Name and Transaction Date.(SBL-
LTY-50015)'.
```

The user can update the Qty to 2 if the user is trying to redeem the same product for a member on the same date time.

The following request processes transactions for an order:

- **URI: http://ServerName:port/siebel/v1.0/service/LOY Redemption Web Service/ ProcessRedemptionTransaction**
- **HTTP Method:** POST
- **Content-Type:** application/json
- **Authorization:** Basic
- **Request body:**


```
{
 "body": {
 "Base LWA LOY Transaction": {
 "IntObjectName": "Base LWA LOY Transaction",
 "IntObjectFormat": "Siebel Hierarchical",
 "ListOfBase LWA LOY Transaction": {
 "LWA LOY Transaction": [
 {
```

```
 "Member Number": "296-183479719",
 "Transaction Date": "04/03/2022 05:30:50",
 "Partner Id": "88-15JV6",
 "Point Id": "88-19ZR5",
 "Amount": 1000,
 "Points": "350",
 "Product Id": "88-1ASW8",
 "Product Name": "MSR Regular Accrual",
 "Program Id": "88-19ZQZ",
 "Qualifying Flag": "Y",
 "Transaction Sub Type": "Product",
 "Transaction Type": "Redemption"
 }
 ]
 }
 }
 }
}
```


Here are the response details for a successful request:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "Base LWA LOY Transaction": {
 "IntObjectFormat": "Siebel Hierarchical",
 "IntObjectName": "Base LWA LOY Transaction",
 "MessageType": "Integration Object",
 "LWA LOY Transaction": {
 "Status": "Processed",
 "Transaction Id": "88-318UCF",
 "Points": "350",
 "Transaction Number": "296-183490719",
 "Sub Status": "Success",
 "Processing Log": "Admin - Redemptions\r\n Got Redeemed. 50 Regular\r\n Got Redeemed. 300 Regular\r
\nNot satisfy the filter.\r\n 1-> Admin - Accruals;\r\n 1-> Admin - Cancellation;\r\n 1-> Admin - Gift;
\r\n 1-> Admin - Gift Points;\r\n 1-> Admin - Loan;\r\n 1-> Admin - Lounge purchase;\r\n 1-> Admin -
 Membership Renewal;\r\n 1-> Admin - Point Reactivation;\r\n 1-> Admin - Points Purchase;\r\n 1-> Admin
 - Promotion Enrolment;\r\n 1-> Admin - Voucher;\r\n 1-> Admin - Voucher Reissue;\r\n 1-> Admin - Point
 Transfer;\r\n 1-> Batch Redemption;\r\n 1-> CT Accruals;\r\n 1-> CT Debits;\r\n 1-> Program - Transfer
 Points;\r\n 1-> Program-Enrolment;\r\n 1-> Recurring Rewards on Primary Member Txn;\r\n 1-> Special
 Points based on Product Line;\r\n;Not within start/end dates.\r\n 1-> Special Points based on Product
 Line;\r\n;",
 "LOY Transaction Redemption Item": [
 {
 "Point Type": "Regular",
 "Promotion Name": "Admin - Redemptions",
 "Redeemed Value": "50",
 "Accrual Item Id": "88-318S1A"
 },
 {
 "Point Type": "Regular",
 "Promotion Name": "Admin - Redemptions",
 "Redeemed Value": "300",
 "Accrual Item Id": "88-318JJQ"
 }
 ]
 }
 }
}
```


- Here are the response details for a successful request when a loan is issued to a loyalty member:

- **HTTP Code:** 200
- **Content-Type:** application/json
- **Response body:**


```
{
 "Base LWA LOY Transaction": {
 "IntObjectFormat": "Siebel Hierarchical",
 "IntObjectName": "Base LWA LOY Transaction",
 "MessageType": "Integration Object",
 "LWA LOY Transaction": [
 {
 "Status": "Processed",
 "Transaction Id": "88-318UD8",
 "Points": "100",
 "Transaction Number": "296-183490748",
 "Sub Status": "Success",
 "Processing Log": "Admin - Loan\r\n Got Applied. 100 Non Qualifying. Regular\r\nNot satisfy the
 filter.\r\n 1-> Admin - Accruals;\r\n 1-> Admin - Cancellation;\r\n 1-> Admin - Gift;\r\n 1->
 Admin - Gift Points;\r\n 1-> Admin - Lounge purchase;\r\n 1-> Admin - Membership Renewal;\r\n 1->
 Admin - Point Reactivation;\r\n 1-> Admin - Points Purchase;\r\n 1-> Admin - Promotion Enrolment;
\r\n 1-> Admin - Redemptions;\r\n 1-> Admin - Voucher;\r\n 1-> Admin - Voucher Reissue;\r\n 1->
 Admin - Point Transfer;\r\n 1-> Batch Redemption;\r\n 1-> CT Accruals;\r\n 1-> CT Debits;\r\n
 1-> Program - Transfer Points;\r\n 1-> Program-Enrolment;\r\n 1-> Recurring Rewards on Primary
 Member Txn;\r\n 1-> Special Points based on Product Line;\r\n;Not within start/end dates.\r\n 1->
 Special Points based on Product Line;\r\n;",
 "LOY Transaction Accrual Item": {
 "Point Type": "Regular",
 "Accrualed Value": "100",
 "Promotion Name": "Admin - Loan",
 "Point Block Name": "Indigo Points",
 "": {}
 }
 },
 {
 "Status": "Processed",
 "Transaction Id": "88-318UD3",
 "Points": "350",
 "Transaction Number": "296-183490743",
 "Sub Status": "Success",
 "Processing Log": "Admin - Redemptions\r\n Got Redeemed. 100 Regular\r\n Got Redeemed. 250
 Regular\r\nNot satisfy the filter.\r\n 1-> Admin - Accruals;\r\n 1-> Admin - Cancellation;\r
\n 1-> Admin - Gift;\r\n 1-> Admin - Gift Points;\r\n 1-> Admin - Loan;\r\n 1-> Admin - Lounge
 purchase;\r\n 1-> Admin - Membership Renewal;\r\n 1-> Admin - Point Reactivation;\r\n 1-> Admin
 - Points Purchase;\r\n 1-> Admin - Promotion Enrolment;\r\n 1-> Admin - Voucher;\r\n 1-> Admin -
 Voucher Reissue;\r\n 1-> Admin - Point Transfer;\r\n 1-> Batch Redemption;\r\n 1-> CT Accruals;\r
\n 1-> CT Debits;\r\n 1-> Program - Transfer Points;\r\n 1-> Program-Enrolment;\r\n 1-> Recurring
 Rewards on Primary Member Txn;\r\n 1-> Special Points based on Product Line;\r\n;Not within
 start/end dates.\r\n 1-> Special Points based on Product Line;\r\n;",
 "LOY Transaction Redemption Item": [
 {
 "Point Type": "Regular",
 "Promotion Name": "Admin - Redemptions",
 "Redeemed Value": "100",
 "Accrual Item Id": "88-318UDC"
 },
 {
 "Point Type": "Regular",
 "Promotion Name": "Admin - Redemptions",
 "Redeemed Value": "250",
 "Accrual Item Id": "88-318UDC"
 }
 ]
 }
 ]
```

```
        }
    }
```

15 Using Siebel REST Outbound Services

Using Siebel REST Outbound Services

This chapter provides an overview of how Siebel CRM supports outbound REST communications for on-premises or cloud applications. It includes the following topics:

About REST Outbound

Siebel CRM supports outbound REST communications for on-premises or cloud applications, allowing Siebel applications to interact with other on-premises or cloud applications that provide REST APIs.

> **Note:** Siebel CRM does not support REST outbound calls in XML format. It only supports REST outbound calls in JSON format.

Siebel outbound REST functionality is built alongside SOAP Outbound functionality, and allows Siebel CRM to call any external endpoint that provides a valid OpenAPI or SWAGGER compliant schema in JSON format.

Siebel outbound REST functionality leverages Siebel Web Tools to import the REST contract and to create the required metadata, such as business services, integration objects, and data such as Web service definitions. For more information about Siebel Web Tools, see *Using Siebel Tools* .

Considerations When Using REST Outbound in Siebel

From Siebel CRM 22.9 Update release, the following limitations apply to REST outbound:

- Siebel REST API only supports OpenAPI 2.0 and OpenAPI 3.0 compatible JSON specifications.
- Files must be in JSON format only.
- You can import JSON files using the Web Service Wizard in Web tools.
- The Siebel Web Tools OM makes a JBS call to the applicationcontainer_internal tomcat while importing the JSON specification which requires setting up the 64-bit Java subsystem with the CONTAINERURL parameter pointing to applicationcontainer_internal tomcat.
- You can download the JSON specification files to your local machine, and save them in any folder. You can also save these files to any directory to which Web Tools has access.
- You can also use the URL pointing to the JSON specification to import into Siebel, which requires setting up the proxy configuration for applicationcontainer_internal tomcat and importing certificates from the host server of the URL into the Trust Store of applicationcontainer_internal tomcat to enable the applicationcontainer_internal tomcat to connect to the host server and download the JSON specification to import.
- When using Siebel REST API, you can use JSON files that refer to other JSON files for the definition which would be auto downloaded, provided, the proxy configuration is set for applicationcontainer_internal tomcat and certificates from the host server are imported into the Trust Store of applicationcontainer_internal tomcat.
- Siebel REST outbound supports security of type basic authentication, and Siebel REST Outbound Request/Response Filter Services can be used to support other security authentication mechanisms to inject the required security parameters into the payload.
- Siebel REST outbound is backward compatible, which means the JSON specification that was imported and being used after 20.8 Update release would continue to work seamlessly with this Update release.

Roadmap for Creating REST Outbound Services in Siebel CRM

This roadmap covers the essential tasks that you must perform to import a JSON into Web Tools, and to create an outbound REST service and repository artefacts.

When these steps are completed, the newly created business service can be used like any other Siebel business service to call external endpoints.

1. Create a 64-bit Java subsystem if it doesn't exist already.

   For more information about Java subsystems, see "Requirements for Implementing a Java Business Service" in *Transports and Interfaces: Siebel Enterprise Application Integration* .
2. (Optional) If you need to increase logs for debugging or diagnostic purposes, update your REST outbound log level value.

   For more information, see *Setting the Log Level for REST Outbound Services*.
3. Create a new outbound REST Service based on an OpenAPI compliant JSON file.

For more information, see *Creating an Outbound REST Service Based on an OpenAPI Compliant JSON File*.

4. In a Siebel Application, open the new REST outbound service in the Outbound REST Services view and verify that you have correctly imported artefacts such as service methods to the Outbound REST Services. You must also verify the endpoint URL in the Address column of the Service Params Applet.
   For more information, see *Creating an Outbound REST Service Based on an OpenAPI Compliant JSON File*.
   In OpenAPI 3.0 specification, it is possible to provide different and multiple Host Servers under Server Node, or in Path under Server Node or at operations Node (such as get, put, post, delete, patch) under Server. Only one Host Server which would be most appropriate to that endpoint URL would be persisted in the Address column for use. If a request needs to be sent to a different Host Server, then the Address column can be updated accordingly, or the Filter service can be used to inject and modify the payload for updating the host at runtime.
5. If necessary, set up the proxy for external endpoint access and import the certificates of the external endpoint into the applicationcontainer_internal tomcat that was configured in the 64-bit Java subsystem.
   For more information, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .
6. If you need to use a security type other than basic authentication, then setup the Request and Response Filter services as required, to inject the security parameters into the payload.
   For more information, see *Overview of the REST Outbound Filter Service*.

When these steps are completed, the newly created business service can be used like any other Siebel business service to call external endpoints.

Setting the Log Level for REST Outbound Services

When using Siebel outbound REST functionality, you may need to change the log level for debugging or diagnostic purposes. The default value is ERROR, and other available values are TRACE, DEBUG, INFO, WARN or FATAL. If you decide to set the log level for REST outbound services, you need to do so in each of your Siebel servers. REST outbound related logs are located in the following file: `ses/applicationcontainer_internal/logs/restoutbound.log`

> **Note:** Outbound services created before the 20.8 release continue to use the pre-20.8 release REST outbound framework. If you want to use the new 20.8 framework, then you need to re-import the OpenAPI Specification (formerly the Swagger Specification) in the 20.8 release of Siebel tools. Before the 20.8 release, the REST outbound files were located in the following file: `ses/applicationcontainer_internal/logs/siebejbs.log`. The 22.9 release supports both the 2.0 and 3.0 OpenAPI Specification, and the OpenAPI Specification which was imported in 20.8 release or after 20.8 release would work seamlessly even after 22.9 update. But, if there is an update to already imported OpenAPI Specification using the 20.8 release, then the OpenAPI Specification (JSON) needs to be reimported after the 22.9 update.

To set the log level for REST outbound services

1. Open the following file: `ses\applicationcontainer_internal\webapps\siebel\WEB-INF\log4j2-siebel.xml`
2. Locate the entry: `REST_OUT_LOGGER`
3. Change the log level as required.
   The following values are valid: ERROR, TRACE, DEBUG, INFO, WARN or FATAL. For example:

```
<logger name="REST_OUT_LOGGER" level="INFO" additivity="false">
```

```
<AppenderRef ref="REST_OUT_LOGGER"/>
```

Configuring Web Tools to Call a Java Business Service

Before you begin: Ensure you have created a 64-bit Java subsystem. For more information about Java subsystems, see *Requirements for Implementing a Java Business Service* in *Transports and Interfaces: Siebel Enterprise Application Integration* .

In order to process the imported OpenAPI JSON file in Web Tools, you must configure the 64-bit Java subsystem to invoke certain Java business services.

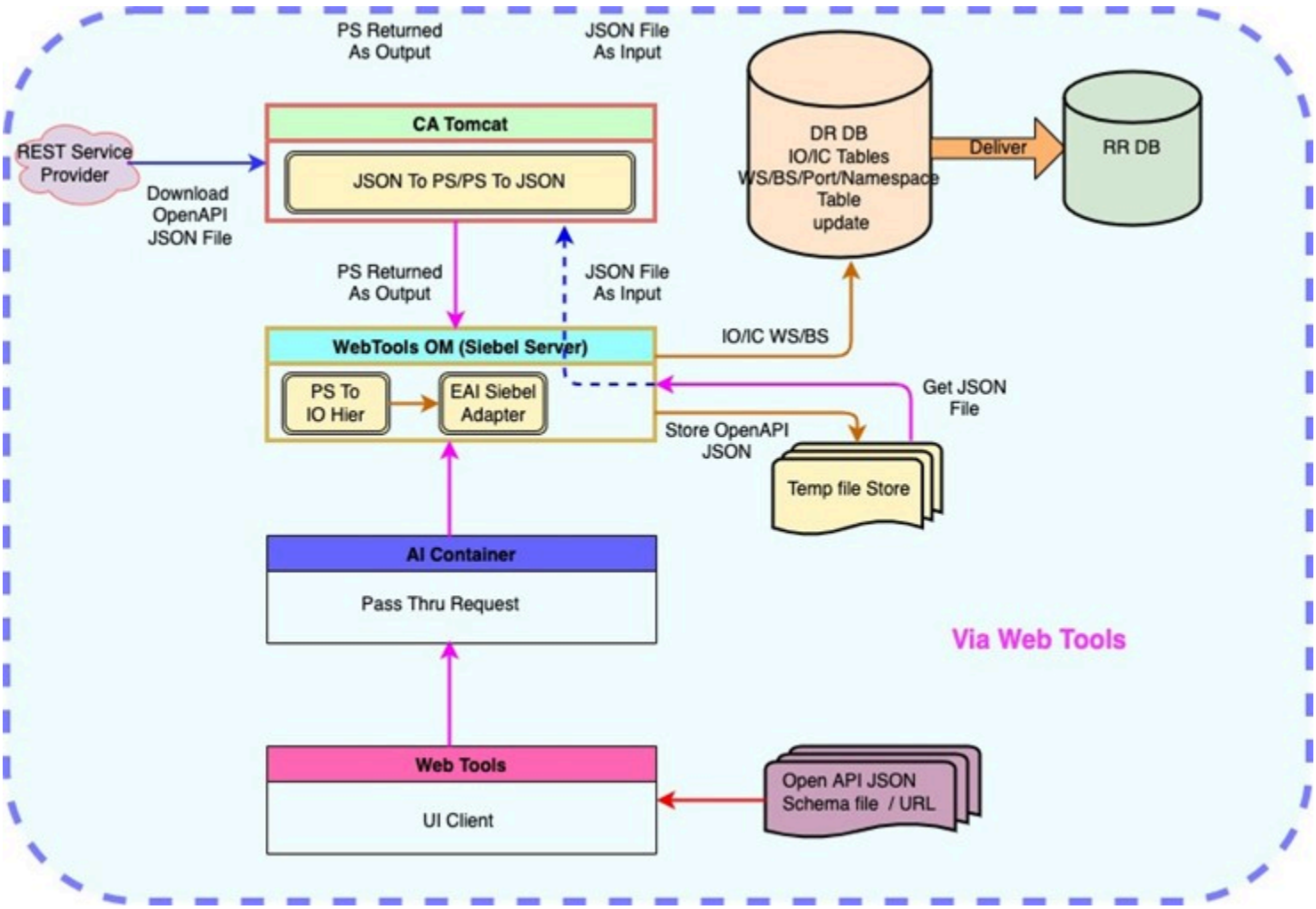
To configure Java subsystem in the server's 64-bit Java subsystem

1. Open srvrmgr.
2. Update the CONTAINERURL param for the named subsystem JAVA64 using the following command in srvrmgr:

   `change param CONTAINERURL=http://localhost:<Config Agent Port>/siebel/jbs` for named subsystem.

Creating an Outbound REST Service Based on an OpenAPI Compliant JSON File

This topic describes how to create an outbound REST service. You do this by importing an OpenAPI compliant JSON file or URL, which describes an external REST API endpoint. The following figure shows how the JSON file or URL import creates an outbound proxy business service, a Web Service, and other artefacts.

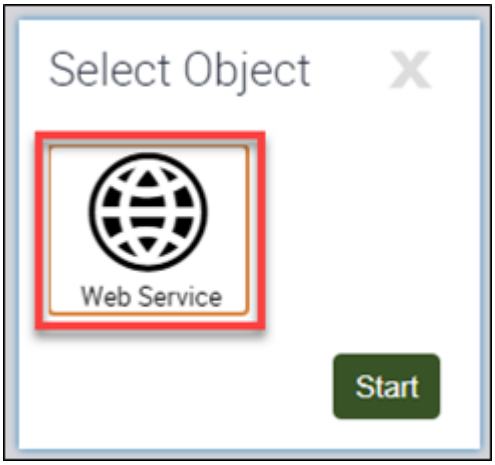


To create an outbound REST service in Web Tools

1. In Web Tools, create and open a new workspace.

2. Start the New Objects Wizard.

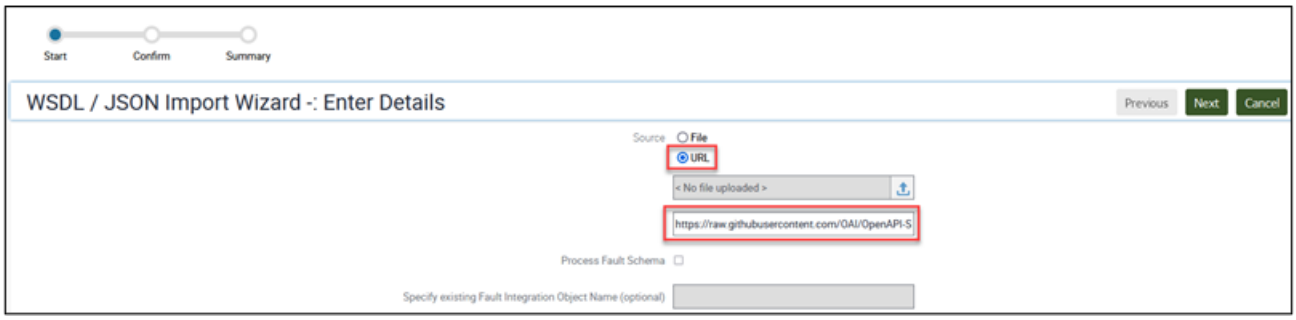
3. Select Web Service and click Start.



The WSDL/JSON Import Wizard appears.

4. Choose File or URL.

   If you choose URL:

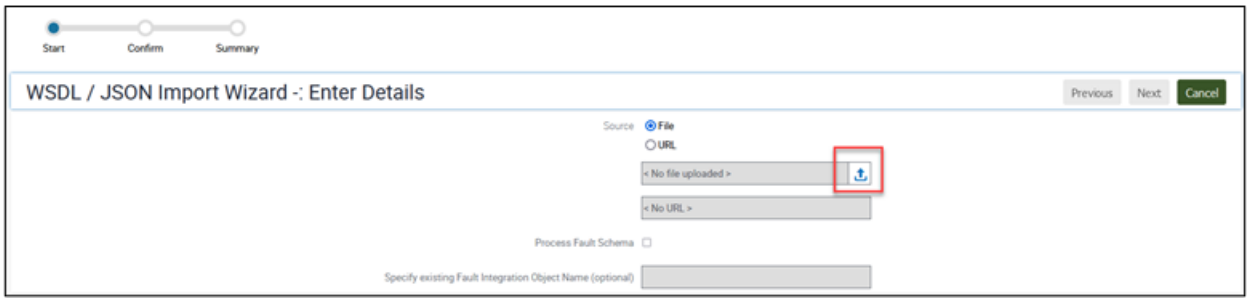


   a. Selecting the URL radio button enables the URL input text box.
   b. Enter the URL from which you want to download a JSON schema file.

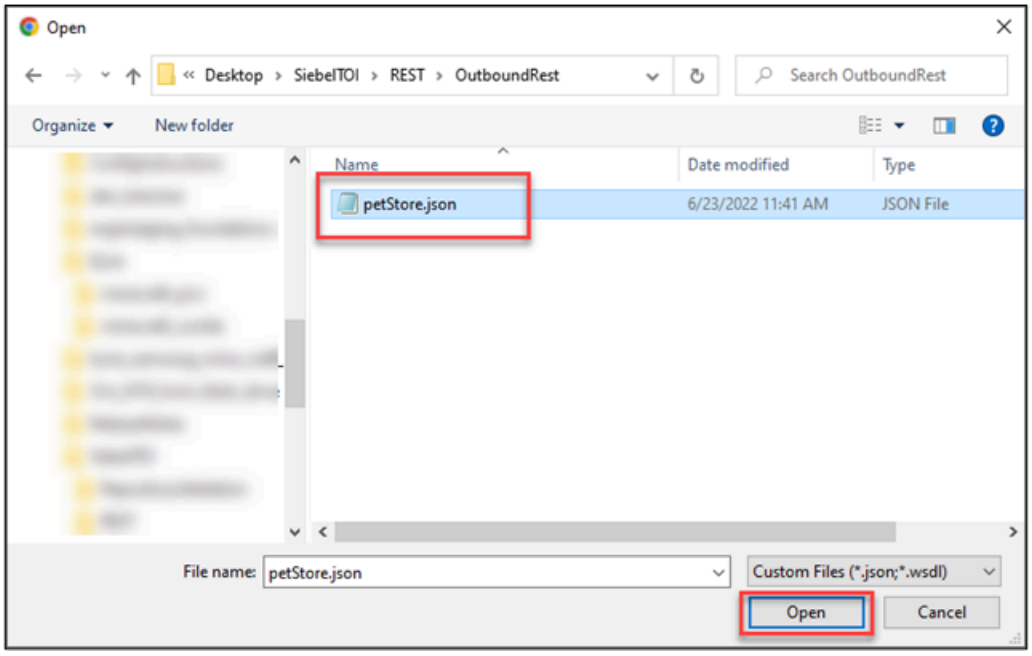
      > **Note:** If the endpoint must be trusted, you must import the root certificate into the `ses\applicationcontainer_internal` server's trust store.

   If you choose File:

   a. Click the upload icon and choose the file to use in the import.

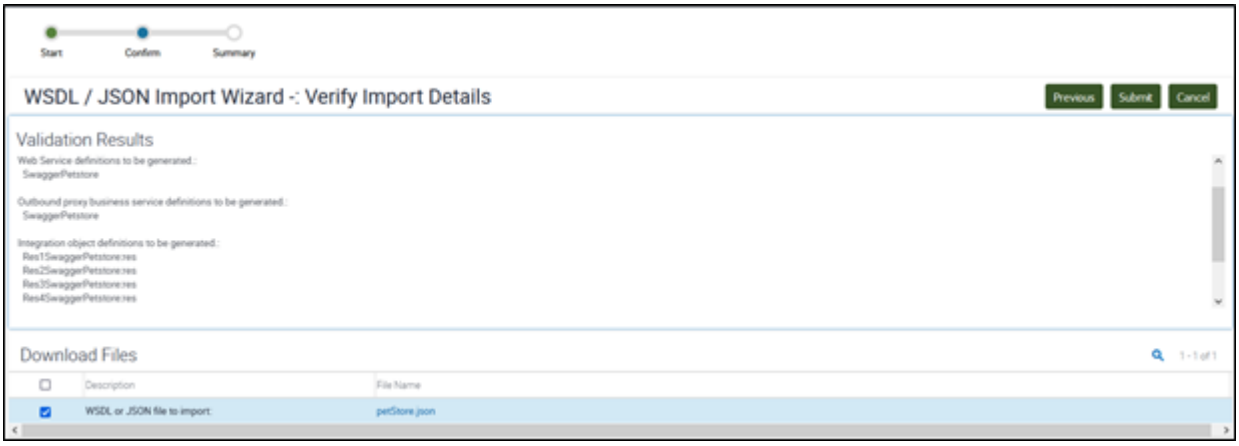


   b. Choose the file and click Open.

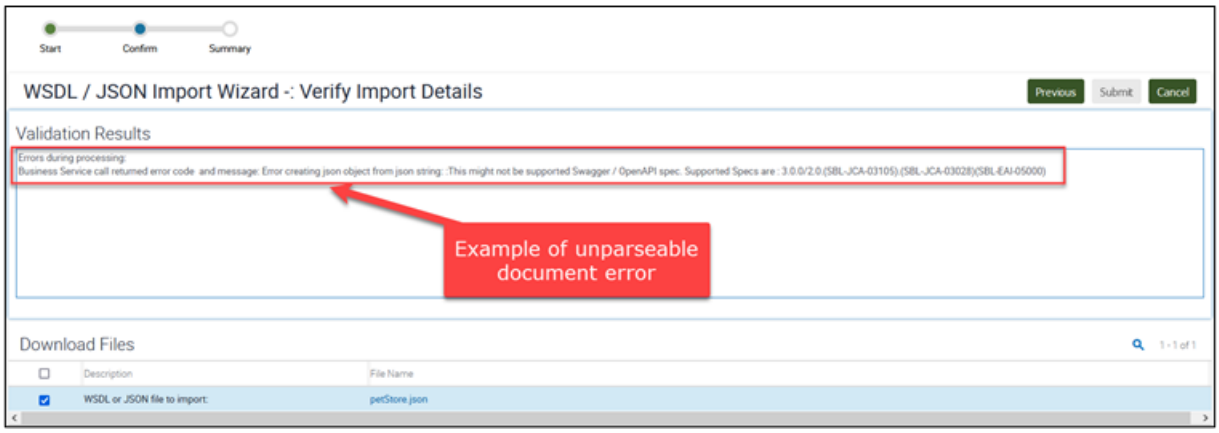

Open
Desktop
SiebelTOI
REST
OutboundRest
Search OutboundRest
Organize
New folder
Name
Date modified
Type
petStore.json
6/23/2022 11:41 AM
JSON File
File name:
petStore.json
Custom Files (*.json;*.wsdl)
Open
Cancel

5. Click Next.

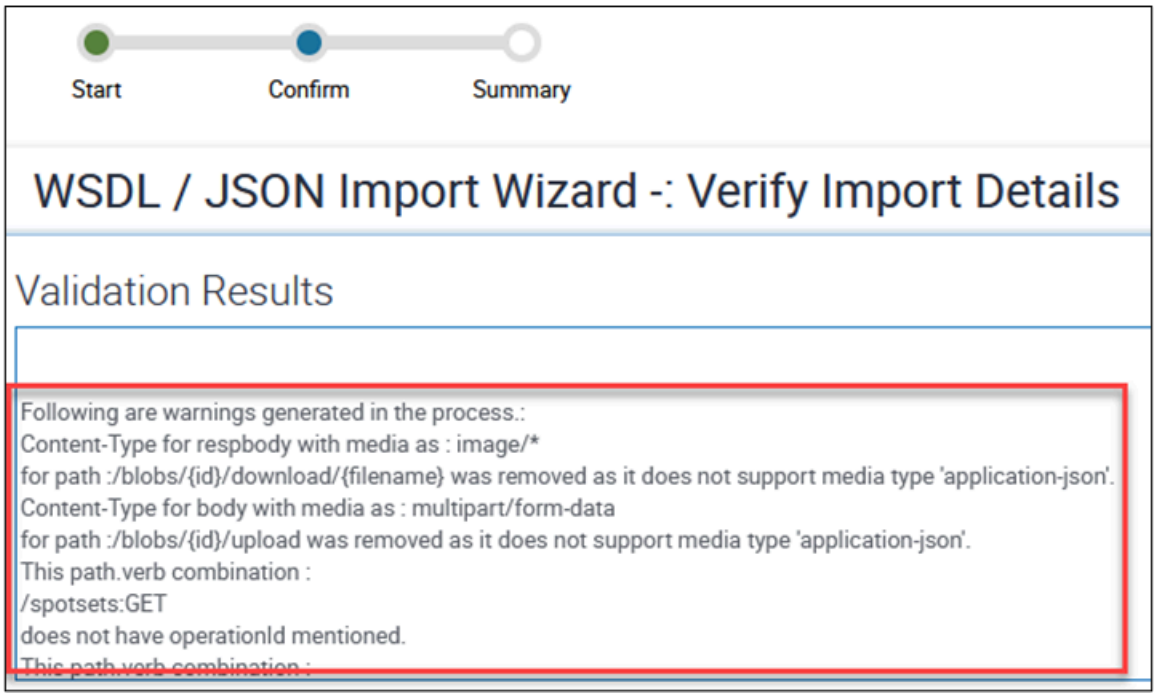
The file contents are validated. The validation results are displayed after completion.



If your file is corrupt and cannot be parsed, processing errors are displayed.



Sometimes, it could be a logical (not syntactical) schema error.



The Validation Results describe:

- Objects to be created
- Warnings from parsing document
- Errors from parsing document
  - By-passed paths: not processed
  - Unresolved server address: substituted with "hostname:port"
  - Failures: why the errors occurred and what you can do to resolve them

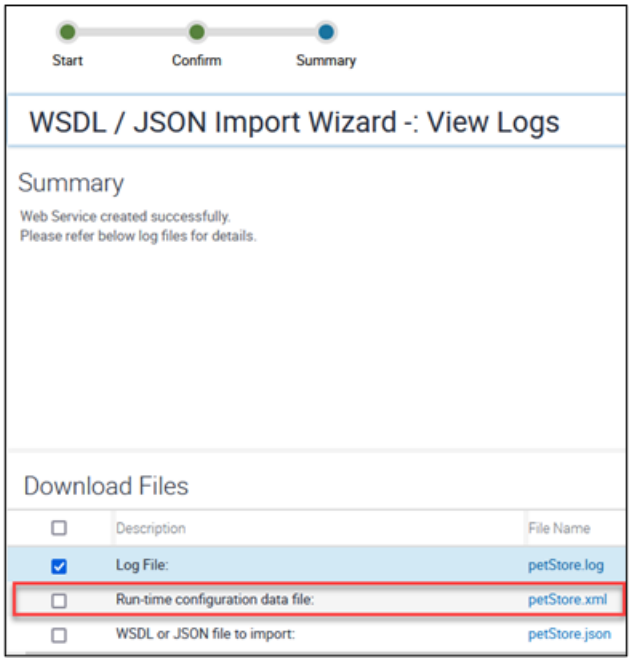
Note: When large JSON files are imported, Web Tools may sometimes timeout while the `SWToolsObjMgr` keeps processing. Due to the large JSON file, there might be thousands of records being inserted into the repository, which will take time. Sometimes the Web Tools UI might become unresponsive. This issue will be resolved in a future release.

When the JSON schema has been parsed, the Admin Data for the user interface is created and inserted into the database.

6. Click the Submit button.

   This creates the repository artifacts and populates the Admin Data. An XML file is also generated. In the future, if the Admin data gets deleted or changed, you can use this XML file to import artefacts into the Outbound REST Services UI in Siebel.

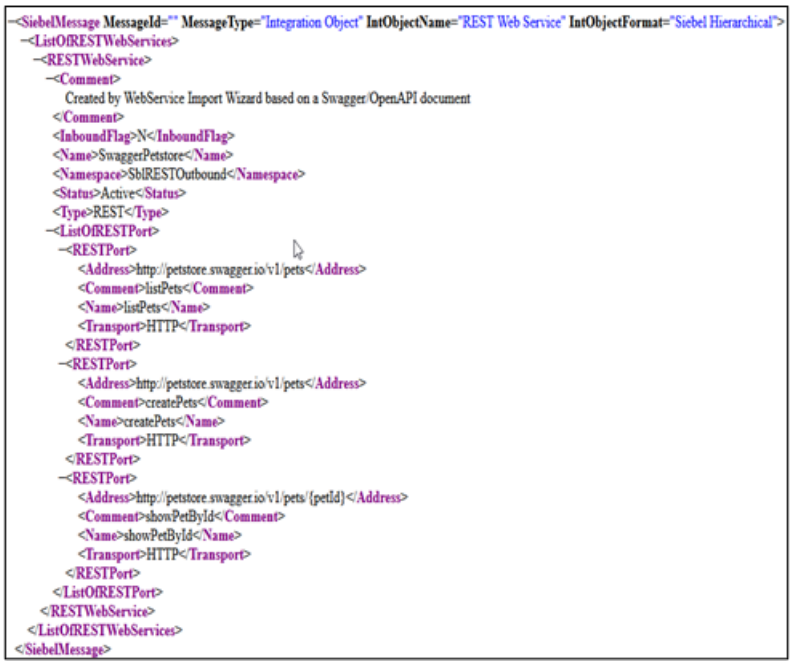
   > **Note:** If you decide not to deliver the workspace with the newly created repository objects, you must manually delete the new outbound REST service in the Outbound REST Services UI in Siebel. This prevents stale service records remaining in the database and helps you avoid naming conflicts during future imports.



   Here is a list of downloaded files:

   - Log file for the import process
   - XML file containing the Admin Data for the Outbound REST Service
   - JSON file imported

   Here is a sample generated XML file.



Here is a list of log files, which you can use for troubleshooting, in case of failures:

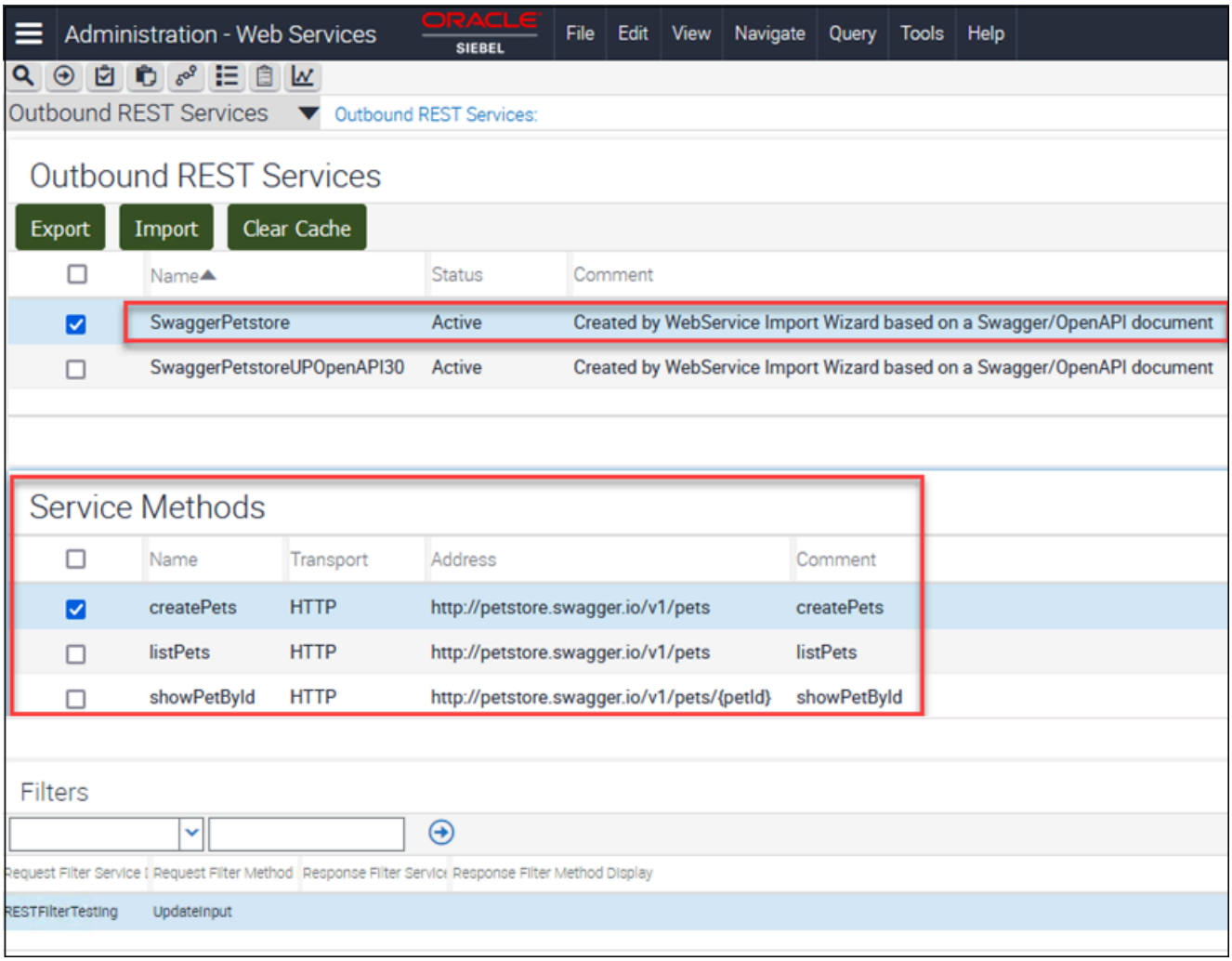
- For Java related issues, see `ses/applicationcontainer_internal/logs/restoutbound.log`
- For Java Heap Size error, which occurs if you import a large JSON file, see `ses/applicationcontainer_internal/logs/siebeljbs.log`

  If you get this error, you need to increase the Java Heap Size for ses/applicationcontainer_internal tomcat.
- For debug logs, see *Setting the Log Level for REST Outbound Services*.
- For Siebel OM related errors, see `SWToolsMgr_enu.logs`
- For information about Siebel Server log files, see *Siebel System Monitoring and Diagnostics Guide* .

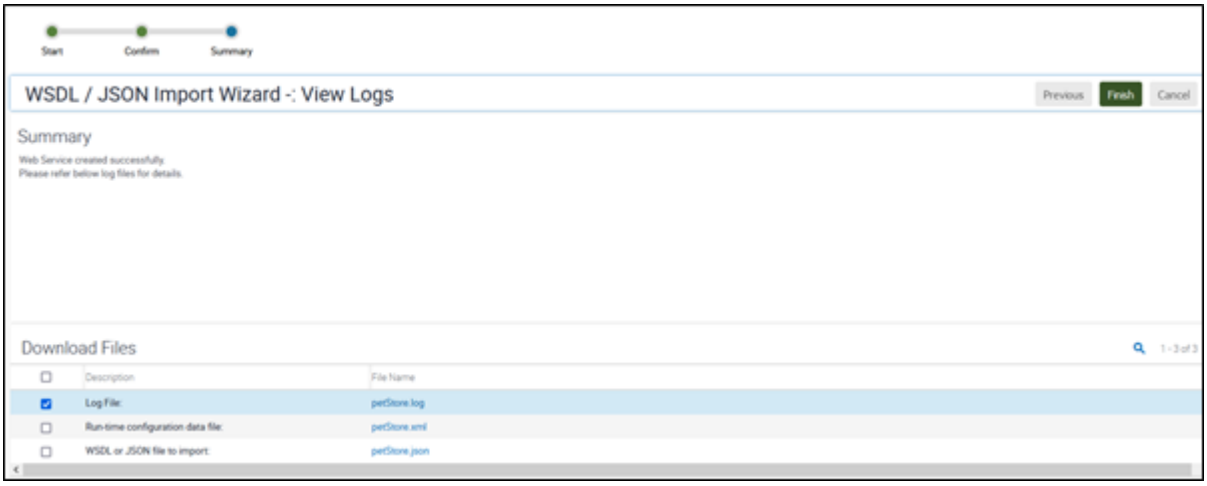
7. To view the new outbound REST service, do the following:

   a. Open an application, such as Call Center.
   b. Navigate to Administration - Web Services.
   c. Navigate to Outbound REST Services.

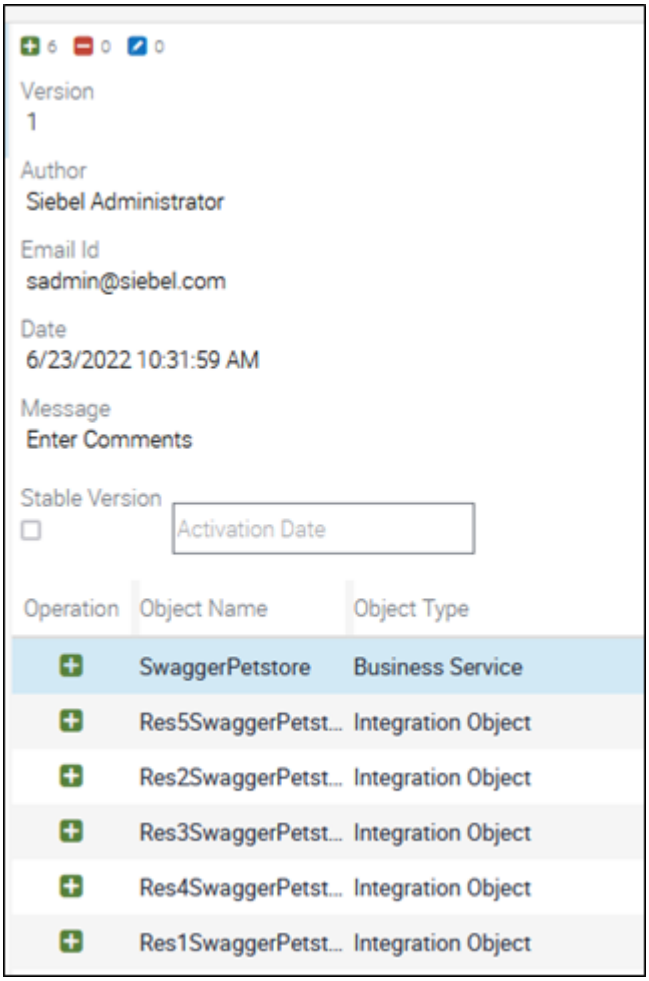
The following Outbound REST Services screen shows the names for the Service, the Endpoint URLs, and operations. Here, the input for Filter Service can be provided against each operation. For more information, see *Overview of the REST Outbound Filter Service*.



8. Click Finish to close the Web Service Wizard.



9. The created objects are listed in the Developer Workspace. You need to deliver the workspace to use the newly created artefacts at runtime.



About the Repository Objects Generated by the JSON Import

The import of OpenAPI Specification generates two objects in the Siebel repository: a proxy outbound REST business service with methods and arguments, and if necessary, a set of integration objects.

Use this JSON file extract as an example in the following topics:

- *About the REST Outbound Service Created by the JSON Import*
- *Overview of the REST Outbound Proxy Business Service*
- *Overview of Generated Repository Integration Objects*
- *Overview of Custom Headers in REST Outbound Proxy Business Service Methods*
- *Adding Custom Headers in REST Outbound Proxy Business Service Methods*

Note: As of this release, the definitions having keywords such as "allOf", "anyOf", "oneOf" are not supported yet. To import JSON having such keywords, you need to resolve the definitions manually and import the modified JSON. For better understanding on how to resolve these definitions manually, refer to *OpenAPI 3.0 specification*. Any schema which has these key words/terms would need to substitute/include the properties/objects/items from $ref or the schema provided under them as per these rules:

- `oneOf` – validates the value against exactly one of the subschemas
- `allOf` – validates the value against all the subschemas
- `anyOf` – validates the value against any (one or more) of the subschemas

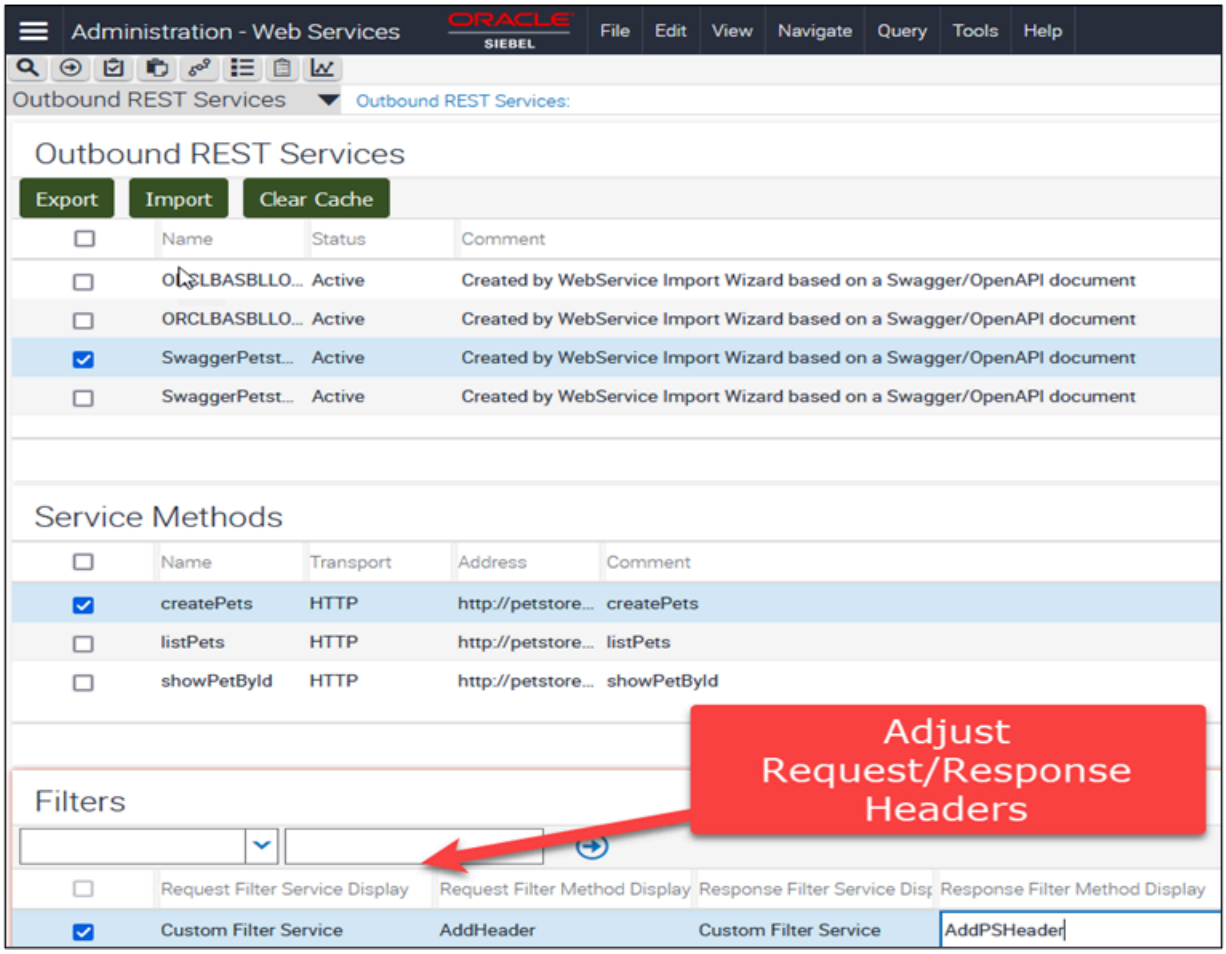
You can also use existing Siebel Integration Objects, instead of those created by the Web Services Import Wizard, when creating a Siebel-to-Siebel outbound REST integration, as explained in the following topic:

- *Using Existing Integration Objects in Outbound REST Services*

About the REST Outbound Service Created by the JSON Import

When the WSDL/JSON wizard finishes, it creates the new outbound REST service, as shown in the following figure. The new outbound REST service has service methods with URLs which shall be invoked for this service. You can update the endpoint URL in the Outbound REST Services screen if needed.

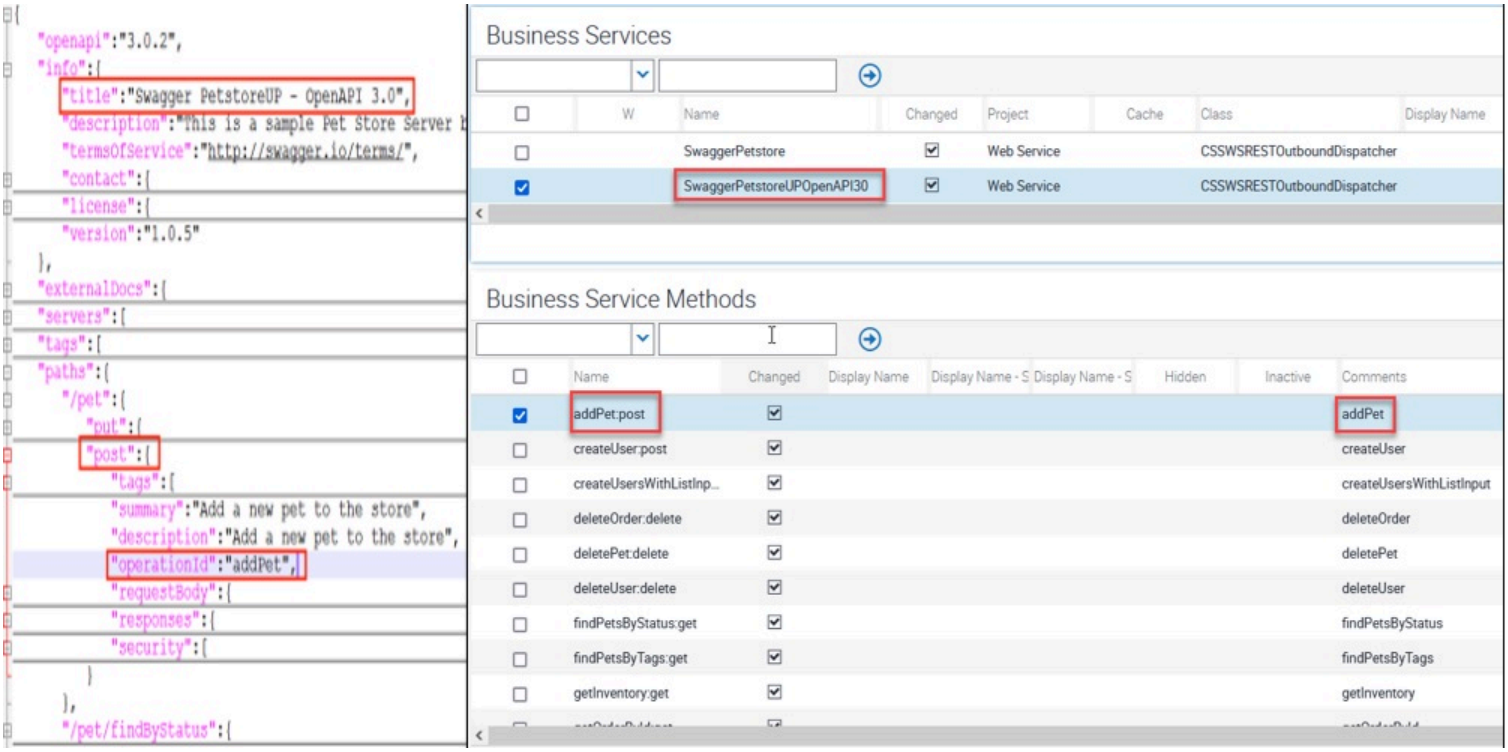
For more information on the business service and the repository artefacts in this REST outbound service, see *About the Repository Objects Generated by the JSON Import*.



Note: If you decide not to deliver the workspace with the newly created repository objects, you must manually delete the new outbound REST service. This prevents stale service records from remaining in the database and it avoids naming conflicts during future imports. For more information about outbound Web services, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

Overview of the REST Outbound Proxy Business Service

The following figure shows the Business Services pane with the business service: `SwaggerPetstoreUPOpenAPI30`. It also shows the Business Service Methods pane, and the method name: `addPet:post`.



Business Services

The first object generated by the WSDL/JSON Import Wizard is a REST outbound proxy business service belonging to the `CSSWSRESTOutboundDispatcher` class. This service acts as a client-side implementation of the REST API endpoint, and it includes the business methods and arguments that correspond to the operations, or `OperationId` fields, defined in the imported JSON document.

The business service name is derived from the `title` value in the JSON document. In this example, the business service is called: `SwaggerPetstoreUPOpenAPI30` . The names of the generated repository artefacts derive from the imported OpenAPI JSON file. A number is appended to each business service, business service method, business method arguments, integration objects and integration components depending on how often you have imported a specific JSON file into Web Tools.

If the names can't be formatted into a standard Siebel naming standard, for example, if the name is too long or if it contains special characters, then the WSDL/JSON Import Wizard automatically generates these names. The original names are available in the Comments field, under the Business Service.

Business Methods

Each operation or `OperationId` value listed in the JSON file becomes a business service method under the generated REST outbound proxy business service. Business methods are named as follows:

- The verb under the path is appended to the `OperationId` value. In our sample JSON file, the OperationId is `AddPet` and `post` is the verb of the path. The business service method is called: `addPet:post`
- If the JSON file has no `OperationId` values, then the JSON file import removes all special characters from the path name and combines it with the required verb to create the business method name.

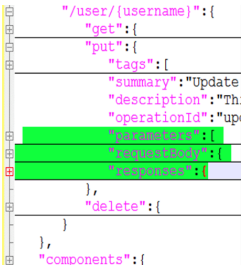
- If the path name is longer than the Siebel standard length, then the business method is named as follows: `Mth<serial number><business service>:<verb>` For example: `Mth10SwaggerPetstoreUPOpenAPI30:get`

  Note: The serial number is between 1 and 999. It starts at 1, and is incremented by 1 for every method created.

Business Arguments

The JSON Import Wizard generates method arguments in addition to the proxy business service in the repository. In 3.0 JSON Specification, Request is moved out of the `parameters` and provided under `requestBody` in the JSON file . The `parameters` listed under the `path` node in the JSON file become proxy business method arguments. The method argument name derives from the `name` property only for the parameter. For `requestBody` and `responses` in the JSON file, there are no names provided in the JSON specification. For `requestBody` the name is `body:body` and for Responses the name has prefix Res to response code and suffixed with `:res`. For example the name for response code 200 would be like `Res200:res`. When you provide the input values for business service method arguments, you can provide these in any order. JSON does not require inputs to be in a specific order.

The following figure shows a file with 3.0 JSON specification:



Parameters listed in JSON files can be any of the following:

- **Simple parameters.** Integers, strings, boolean, and decimal number become simple method arguments with corresponding Siebel types.

- **Complex parameters.**
  - Complex parameters can be with $ref references to definitions under components or without references in the JSON file having type as object or array, and can become integration objects.
  - If there are simple or complex parameters as arrays, or complex parameters without a $ref reference, then these parameters also become integration objects.
  - They are named by appending the new business service to the automatically generated integration object name.
  - Complex parameters can be derived from JSON file under `parameters`, `requestBody`, or `responses` in the JSON file.
  - If the complex parameters are derived from JSON file under `parameters`, then the naming convention for the integration objects would be like `Arg<serial number><business service>:<in>`

    For example: `Arg4SwaggerPetstoreUPOpenAPI30:query`

    In this example, the `<in>` value specifies where this parameter is defined in the outbound call, such as `query`, `header`, `path`, `formData` and `cookie`.
  - If the complex parameters is derived from JSON file under `requestBody`, then the naming convention for the integration objects would be like `Req<serial number><business service>:body`

    For example `Req2SwaggerPetstoreUPOpenAPI30:body`
  - If the complex parameters are derived from JSON file under `responses`, then the naming convention for the integration objects would be like `Res<serial number><business service>:res`

    For example `Res2SwaggerPetstoreUPOpenAPI30:res`

  The following suffixes are used with business service method arguments:

  - **:path.** This argument substitutes the path parameter for the appropriate value.

    For example, in the following URI:

    ```
    /store/order/{orderId}
    ```

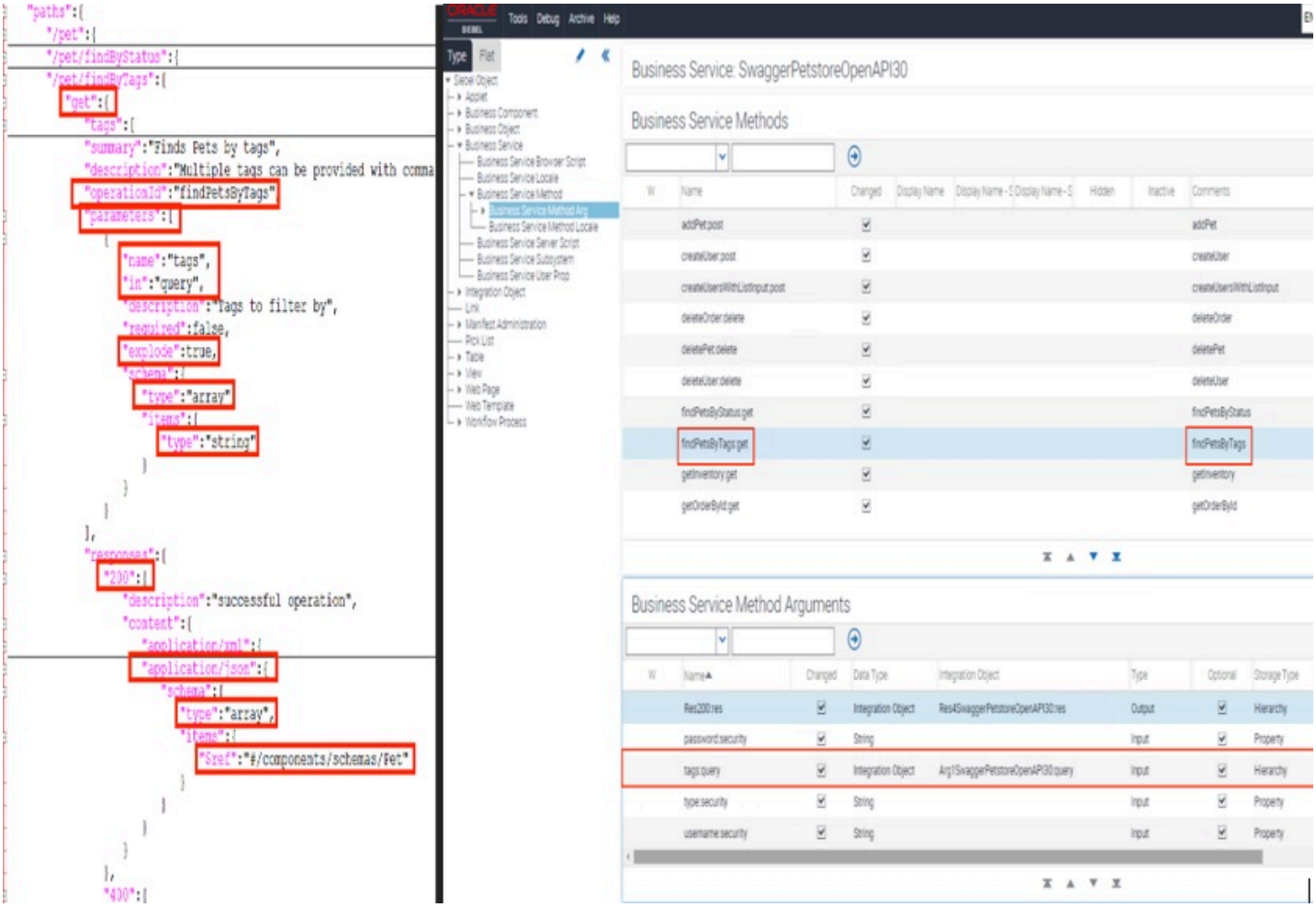
    In the business service method argument, the parameter `orderId` becomes `orderId:path`. The value which the user provides at runtime replaces this path parameter. For example, if you provide the value 123456 for `orderId:path`, then at runtime this is substituted for `/store/order/123456`.
  - **:query.** This argument is passed as a query parameter in the URL.
  - **:header.** This argument is passed as an header parameter in the request.
  - **:security.** This argument is used as a security parameter where required in the request.
  - **:body.** This argument is used as the request body.
  - **:res.** This argument is used to map the response received from the host server.
- **Security parameters.** These parameters are appended with the suffix `security`, for example: `username:security`. As of this release, there are two security parameters retained from earlier 20.8 release. They are `username:security` and `password:security`. These two parameters are retained for backward compatibility with 20.8 release. These two parameters can aid in basic authentication only. For other security schemes/ authentication requirement, use the Filter service. For more information, see *Overview of the REST Outbound Filter Service*.

Note: From Siebel CRM 20.8 Update release, only basic authentication was supported, so you could only pass the `username:security` and `password:security` arguments at runtime. From Siebel CRM 22.9 Update release, other authentication mechanisms are supported by way of Filter Service and Filter methods using Request and Response Filter in the Application Administration > Web Services > Outbound REST Services UI.

Note: Outgoing REST requests using Basic Authentication must contain the following three parameters to avoid an error from an external system which may be a variation of "authorization needed".

Parameter Name	Value
type:security	Basic
username:security	The user name for the external system
password:security	The user's password

The following figure shows the Business Service Method Arguments pane. This pane lists the method argument generated from a complex parameter: `Arg1SwaggerPetstoreUPOpenAPI30`. It also lists the method arguments security parameters `username:security` and `password:security` for backward compatibility with 20.8 upgrade for basic authentication usage.



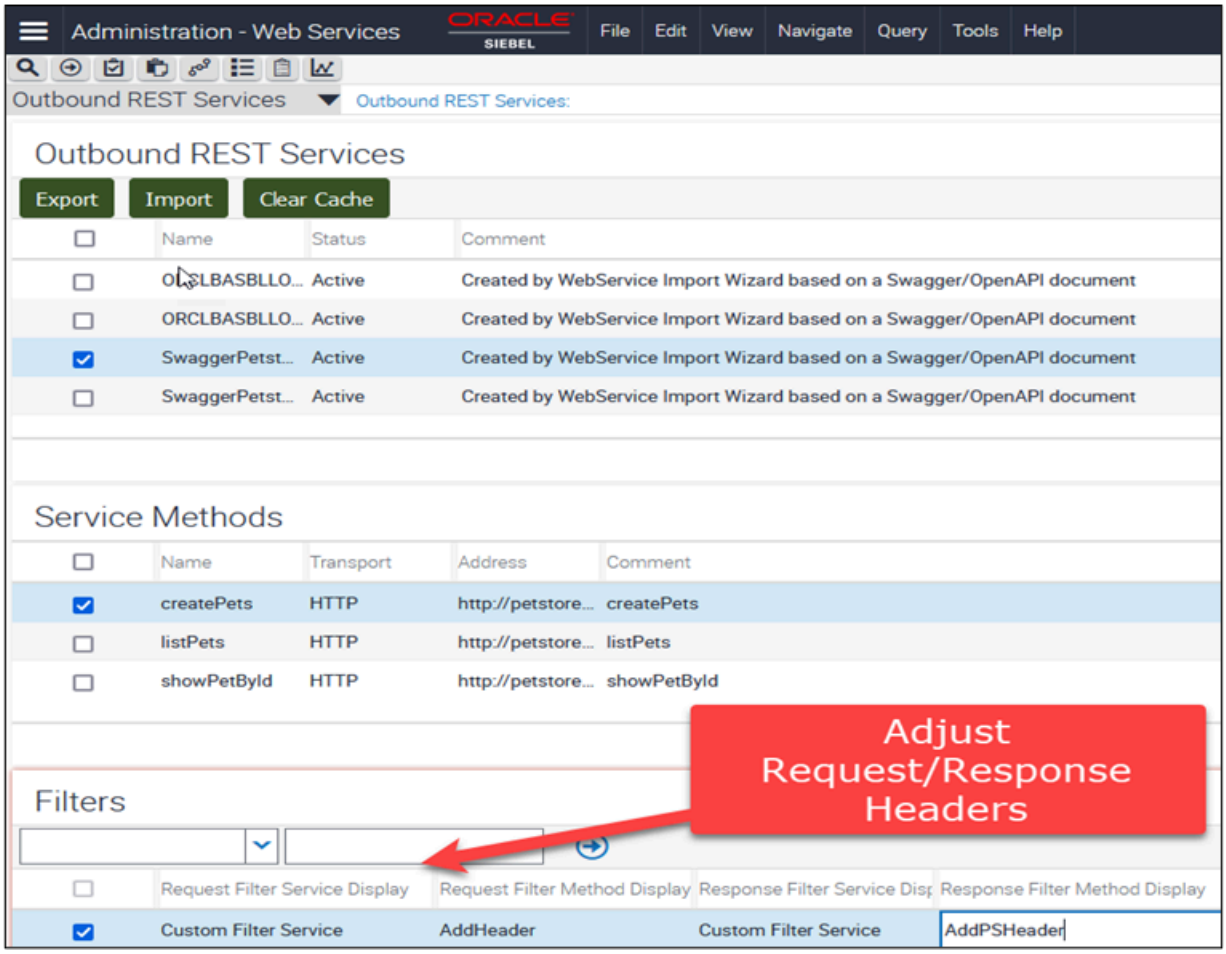
In the above figure, the method name is generated from `operationId` if provided in the JSON file like `<operationId>:<operation>`. For example, in the above figure, the name `findPetsByTags` is the operationId and `get` is the operation verb, therefore the method name generated is `findPetsByTags:get` after preprocessing the `operationId` string for any special characters and reducing the length within 75 chars. If these conditions are not satisfied, then a method name like `Mth<serial number><business service>:<verb>` is generated. For example: `Mth10SwaggerPetstoreUPOpenAPI30:get`.

In the above example, for business service method argument, name is taken from the `name` property under `parameters` which is `tags` and the `in` property which is `query`. Hence the Business Service Method Argument name is generated as `tags:query`. As the parameter is of type `array`, an Integration Object is generated to represent the data for this parameter at runtime. The Integration Object name is generated like `Arg<serial number><business service>:<in>` which then becomes `Arg1SwaggerPetstoreUPOpenAPI30:query` in the above example. The `optional` column is checked for argument `Arg1SwaggerPetstoreUPOpenAPI30:query`, as the JSON file specifies `required` value as `false`. So, at runtime, when invoking this proxy Business Service, providing this argument is optional.

> **Note:** If the JSON file does not specify the data types clearly, then it would be defaulted to String / DTYPE_TEXT while creating the Business Service Arguments / Integration Fields. For number data type in JSON file, if format is not specified, then it will be defaulted to double. For integer data type in JSON file, if the format is not specified then it will be defaulted to integer (int32).

Overview of the REST Outbound Filter Service

In **Application Administration** > **Web Services** > **Outbound REST Services**, you can provide the Request and Response Filter inputs using any Business Service and Business Service Methods as shown in the following figure, provided that the Business Service and its methods can modify the payload and add necessary header or security parameters.



The REST Outbound Filter Service can be used to do the following:

- Add additional host addresses. The Filter service can be used to update the host in the payload at runtime.
- Use any other Business Service to configure the Filters for Request/Response payload customization and to add any custom headers.
- Use any other Business Service to configure the Filters for injecting the security-related parameters other than basic authentication, such as OAuth, API-Key and other authentication mechanisms into the Request/Response payload.
- The Payload can be intercepted, after validation at the Siebel layer before it is sent out using the Filter service and the payload has the below structure and format, where there is a top-level property set and inside it there are separate child property set, such as:
  - Top level Propertset
    - url Propertset

    - query Propertset
    - body Propertset
    - security Propertset
    - misc Propertset
    - header Propertset

- The user can inject any property into the property set of url, query, body, and header and at the appropriate place. Let's say for example the user wants to add some header (say OAuth header or token), then using the Filter service the user can intercept the payload and inject the header parameter into the header property set using the Filter Business Service server script. Likewise, the user can add any parameter to any of the above listed child property set to tune the payload if required. The `misc` property set mentioned above is for internal usage of Siebel. The `security` property set is used only for basic authentication as of now.
- This injection of property or tuning of the payload is solely left to the user and is no more validated for correctness after filter service invocation, hence the user needs to be cautious in adding the properties at the right place into the right child property set of the payload PropertySet.
- User should only alter or add properties and should not alter the hierarchy of the property set i.e. different types child property set under top level property set.
- Below is a sample property set to show how intercepted property set looks at the filter service.

```
> Type =
- > Child property set #1 at level 1:
- > Value =
- > Type = url
- > httpMethod = POST
- > url = https://oicpmapdev-oicpm.integration.ocp.oraclecloud.com:443/ic/api/integration/v1/flows/rest/
ORCL-BA-SBL_LOY_MEM_SYNC_TO_CT/1.0/296-187967036
- > Child property set #2 at level 1:
- > Value =
- > Type = query
- > query =
- > Child property set #3 at level 1:
- > Value =
- > Type = body
- - > Child property set #1 at level 2:
- - > Value =
- - > Type = IO1ORCLBASBLLOYMEMSYNCTOCT1
- - - > Child property set #1 at level 3:
- - - > Value =
- - - > Type = RootIC1
- - - - > Child property set #1 at level 4:
- - - - > Value =
- - - - > Type = ListOfSiebelMessage__SKIP__WRAPPER
- - - - - > Child property set #1 at level 5:
- - - - - > Value =
- - - - - > Type = SiebelMessage
- - - - - > string_IntObjectFormat = Siebel Hierarchical
- - - - - > string_IntObjectName = Base LOY Member CT
- - - - - - > Child property set #1 at level 6:
- - - - - - > Value =
- - - - - - > Type = ListOfLOY_0x737063_Member_0x737063_CT__SKIP__WRAPPER
- - - - - - > Child property set #1 at level 7:
- - - - - - > Value =
- - - - - - > Type = LOY_0x737063_Member_0x737063_CT
- - - - - - > string_Status = Active
- - - - - - > string_Last_Name = int
- - - - - - > string_First_Name = memberss
- - - - - - > string_Country =
- - - - - - > string_Id = 88-33WSAK
- - - - - - > string_Gender =
- - - - - - > string_Last_Update = 10/31/2022 09:26:51
- - - - - - > string_Email_Address =
```

```
- - - - - - - > string_Middle_Name =
- - - - - - - > string_Birth_Date = 10/04/1998
- - - - - - - > string_Con_Last_Update = 10/27/2022 11:24:07
- - - - - - - > string_City =
- - - - - - - > string_Postal_Code =
- - - - - - - > string_Member_Number = 296-187967036
- > Child property set #4 at level 1:
- > Value =
- > Type = security
- > password = XXXXXX
- > type = basic
- > username = +++++++
- > Child property set #5 at level 1:
- > Value =
- > Type = misc
- > serviceMethod = memnumpost1
- > Child property set #6 at level 1:
- > Value =
- > Type = header
```

- Adding security headers to REST Outbond request:

  Security headers can be added to outbound payload in two ways:

  - Adding script in PreInvokeMethod of the proxy business service:

    To add security headers into the pay load, see the "Configuring OAuth Support for Siebel REST Outbound Connections" topic in the Security Guide.
  - Using filter service to inject the parameters into the payload/response using custom Filter Service:
    i. Create a new business service 'RESTFilterTesting' with UpdateInput and UpdateOutput methods.
    ii. Create a business service Service_PreInvokeMethod script with the below code:

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 var j;
 var token_type = "tokentype";
 var access_token = "1234567890";
 if(MethodName == "UpdateInput" && Inputs != null)
 {
 for (var i = 0; i < Inputs.GetChildCount(); i++)
 {
 j = Inputs.GetChild(i);
 if(j.GetType() == "header")
 {
 j.SetProperty("Authorization", token_type + " " + access_token);
 break;
 }
 }
 }
 return (CancelOperation);
}
```

    iii. Configure Administration – Web Service -> Outbound REST Services screen with the above for Request and Response Filter under Filters for the REST Service -> Service Methods that you need to inject these above parameters.
    iv. You need to add appropriate script if required to intercept the response like script below and configuring the method (for example UpdateOutput) at the Administration – Web Service -> Outbound REST Services screen with the above for Filter business service under Filters for the REST Service -> Service Methods.

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 if(MethodName == "UpdateOutput" && Outputs!= null)
```

```
 {
 // write your script code here to update the response
 }
 return (CancelOperation);
}
```

Below is the sample updated Request done by the Filter business service script above:

```
> Value =
> Type =
- > Child property set #1 at level 1:
- > Value =
- > Type = url
- > httpMethod = POST
- > url = https://https://petstore3.swagger.io/api/v3/pet
- > Child property set #2 at level 1:
- > Value =
- > Type = query
- > query =
- > Child property set #3 at level 1:
- > Value =
- > Type = body
- - > Child property set #1 at level 2:
- - > Value =
- - > Type = Req2SwaggerPetstoreOpenAPI30:body
- - - > Child property set #1 at level 3:
- - - > Value =
- - - > Type = ReqBody__SKIP__WRAPPER
- - - - > Child property set #1 at level 4:
- - - - > Value =
- - - - > Type = pet
- - - - > long_id = 512312459
- - - - > string_status = pending
- - - - > string_name = kk5name123
- - - - - > Child property set #1 at level 5:
- - - - - > Value =
- - - - - > Type = category
- - - - - > long_id = 11
- - - - - > string_name = categorykk5name
- - - - - > Child property set #2 at level 5:
- - - - - > Value =
- - - - - > Type = ListOftags__Array
- - - - - - > Child property set #1 at level 6:
- - - - - - > Value =
- - - - - - > Type = tags
- - - - - - > long_id = 5
- - - - - - > string_name = tagskk15name
- - - - - - > Child property set #2 at level 6:
- - - - - - > Value =
- - - - - - > Type = tags
- - - - - - > long_id = 6
- - - - - - > string_name = tagskk25name
- - - - - > Child property set #3 at level 5:
- - - - - > Value =
- - - - - > Type = ListOfphotoUrls__Primitive
- - - - - - > Child property set #1 at level 6:
- - - - - - > Value =
- - - - - - > Type = photoUrls
- - - - - - > string_elem__value = 55
- - - - - - > Child property set #2 at level 6:
- - - - - - > Value =
- - - - - - > Type = photoUrls
- - - - - - > string_elem__value = 66
- > Child property set #4 at level 1:
- > Value =
```

```
- > Type = security
- > Child property set #5 at level 1:
- > Value =
- > Type = misc
- > serviceMethod = addPet
- > Child property set #6 at level 1:
- > Value =
- > Type = header
- > Authorization = tokentype 1234567890
- > Content-Type = application/json
```

Note: For Siebel Update 21.8 to 22.8 user must follow the first approach of adding script to PreInvokeMethod of the proxy business service as described in the Siebel Security Guide -> Using OAuth with Siebel REST -> Configuring OAuth Support for Siebel REST Outbound Connections. After 22.9 to add security header, user can either follow the first approach or follow filter service approach of 22.9 guide. The difference is that, till 22.8 the steps are to add a Proxy Business Service Method Argument (Name: "Authorization:header") and then add a script in the proxy business service's Service_PreInvokeMethod event handler to assign value to this parameter. Post Siebel Update 22.9, this can be achieved by using the Custom Filter service step for injecting the "Authorization" or any other parameter as described in this topic.

Overview of Generated Repository Integration Objects

If necessary, the WSDL/JSON Import Wizard can also generate integration objects. Each complex object, simple array object, or array type object defined in the JSON file converts into a Siebel integration object, composed of a parent or root integration component, and child integration components.

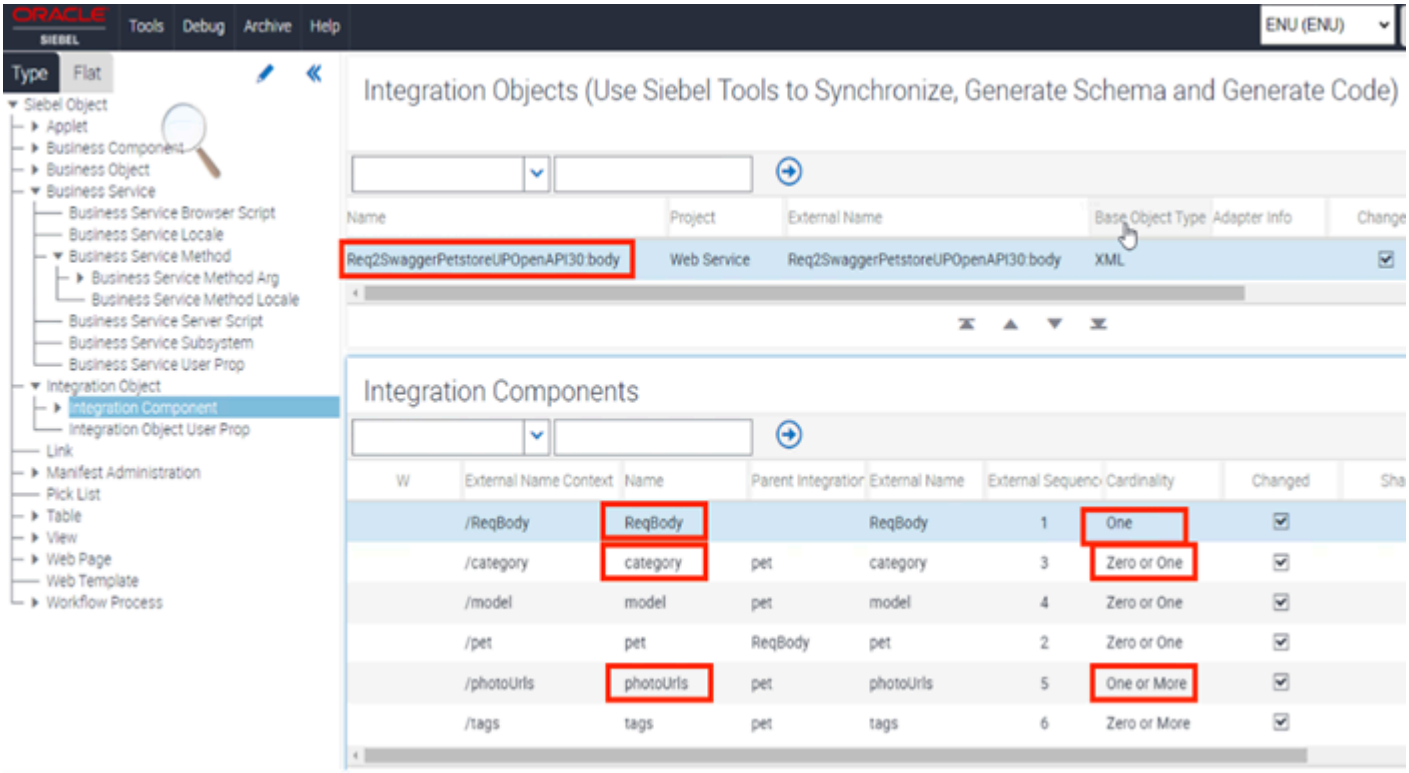
About Integration Components

The following figure shows the Integration Objects pane. This lists the new integration object `Req2SwaggerPetstoreUPOpenAPI30:body`. The Integration Components pane shows the root integration component called `ReqBody` and its associated integration components:

- Root Integration component called `ReqBody`
- A Child Integration component to `ReqBody` called pet is generated. This is the actual object mapping in the JSON file.

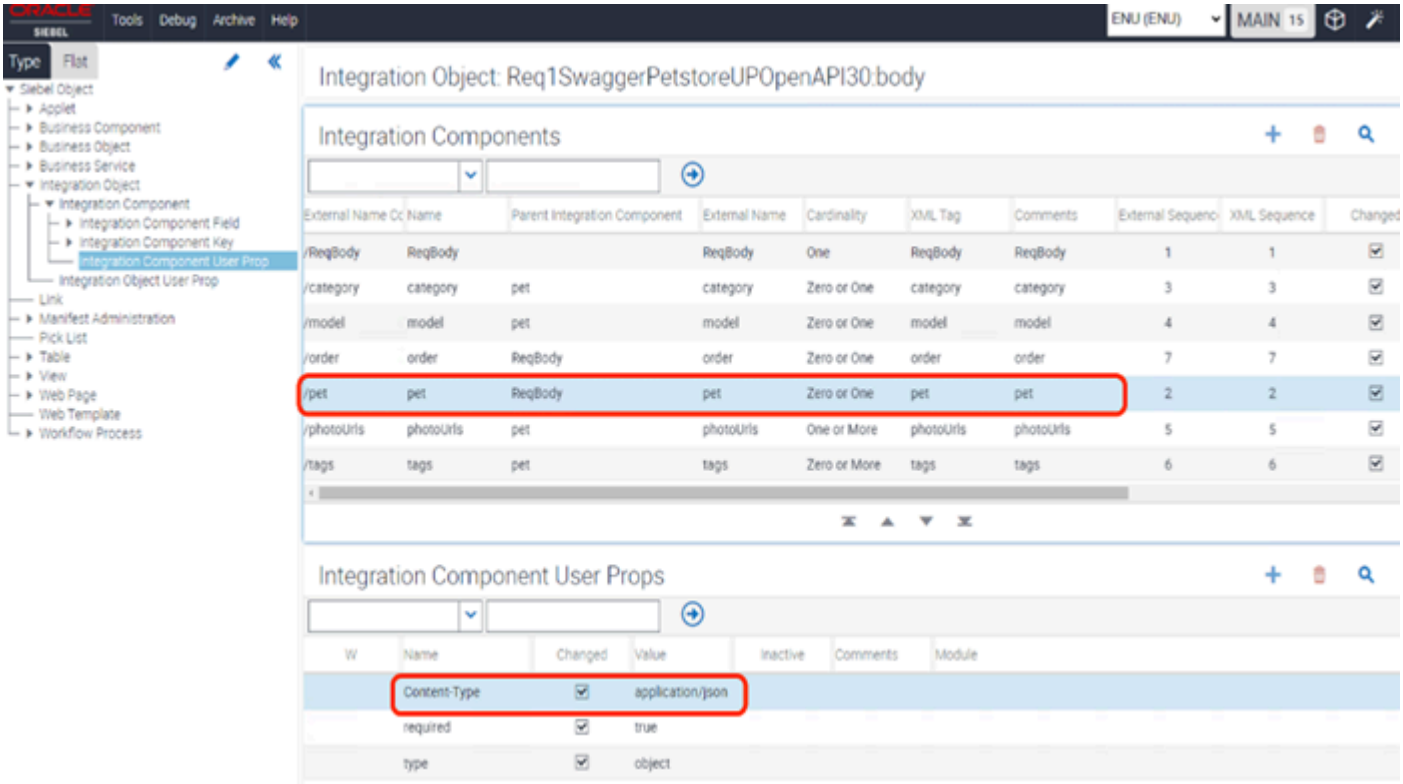
  There will also be a few Integration Component User Property elements generated for each Integration Component, such as format, required, Content-Type, type, isarray which are used for validation and runtime functionality.
- Child objects called `category`
- An array called `photoUrls`
- An array called `tags`

Arrays of simple and complex objects within a complex object are represented as integration components, which are located within the root integration component of the integration object. For example the array called `photoUrls` is a string array, and is represented as an integration component under `Req2SwaggerPetstoreUPOpenAPI30:body`.



After performing the import, the complex object **`ReqBody/Pet`**, has the following properties:

- **`id`**
- **`name`**
- **`status`**



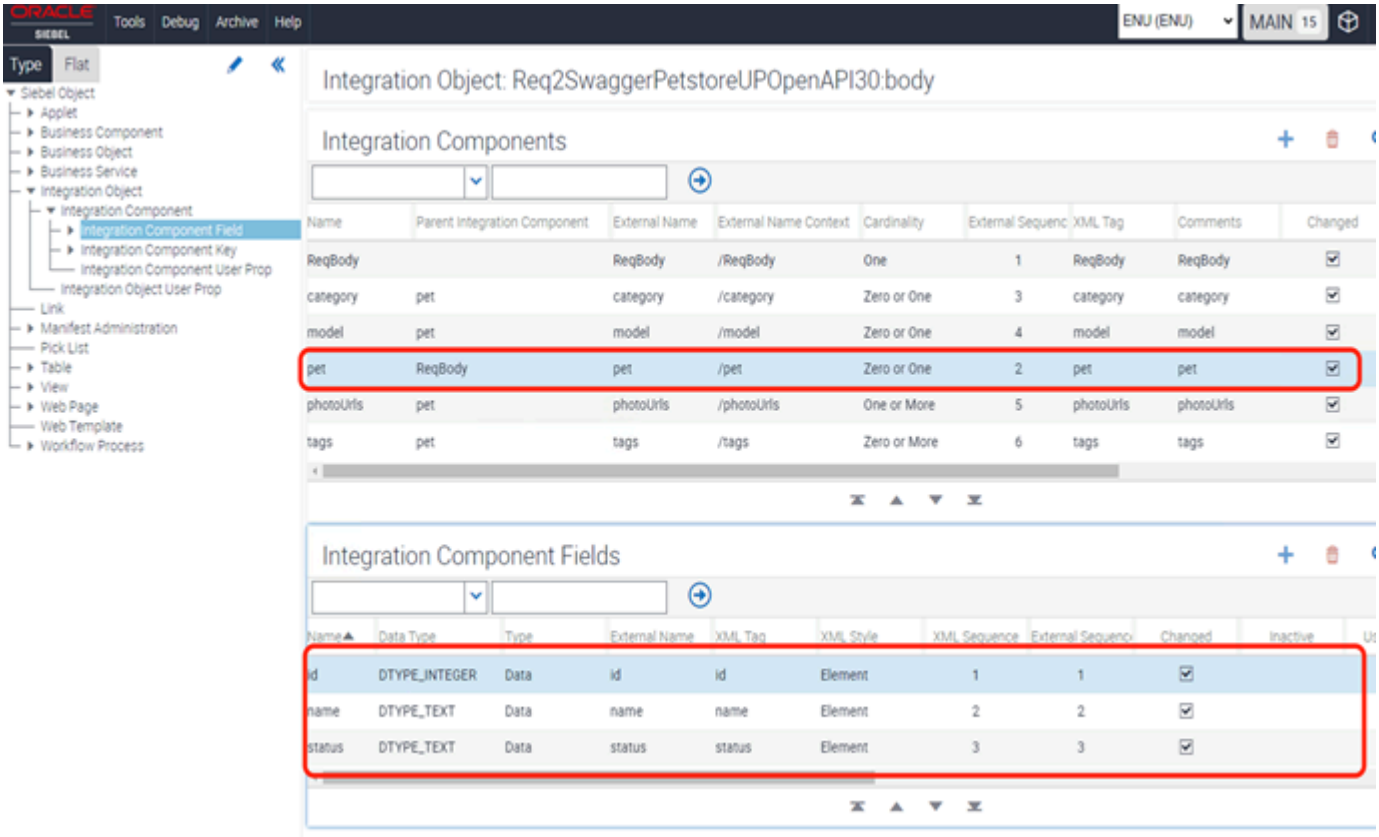
In the above example, the integration object **`Req1SwaggerPetstoreUPOpenAPI30:body`** has the root integration component **`ReqBody`** which represents the wrapper to hold the actual request body for different **`content-type`** under **`content`** in the JSON file for operations such as put, post, patch and delete. If the data represented is different for each **`content-type`** inside **`content`** under the **`requestBody`** in the JSON file, then there would be multiple integration components created to represent each **`content-type`**. The **`content-type`** represented under **`content`** is identified from the user property created for the integration component named as **`Content-Type`** and the value as the actual content-type given in the JSON file, for example **`application/json`** (in the above example). As of this release, only application/json, application/*, and */* **`content-type`** are supported. Hence, repository object, only for those content types, is generated during import, and any unsupported content types are ignored.

The naming convention for the root integration component is as follows: **`ReqBody`**. In this instance, the integration object is called **`Req1SwaggerPetstoreUPOpenAPI30:body`**, therefore the value of the integration object serial number is 1. The root integration component **`ReqBody`** is a placeholder, and it has no significance in the input provided at runtime. In addition, the integration object **`Req1SwaggerPetstoreUPOpenAPI30:body`** has the root integration component **`ReqBody`** with the integration component fields:

- **`id`**
- **`name`**
- **`status`**

> **Note:** If the JSON file does not specify the data types clearly, then it would be defaulted to String / DTYPE_TEXT while creating the Business Service Arguments / Integration Fields. For number data type in JSON file, if format is not specified, then it will be defaulted to double. For integer data type in JSON file, if the format is not specified, then it will be defaulted to integer (int32).

The following figure shows the newly created integration component fields for the integration object **`Req2SwaggerPetstoreUPOpenAPI30:body`**.



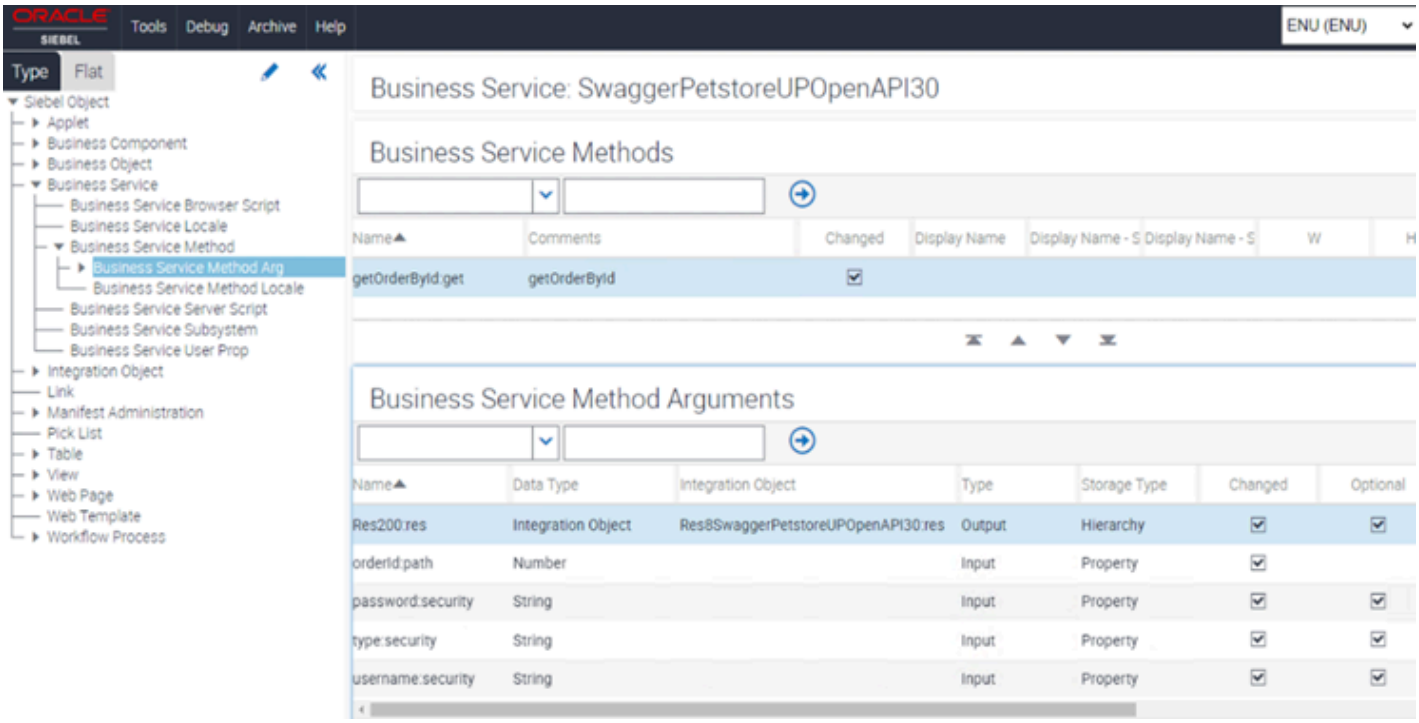
About Integration Component Response Arguments

Response arguments are prepared according to the specification of the responses under each path in the JSON file and corresponding operation or verb (such as get, put, post, delete, and patch) combination in the OpenAPI file, and the response structure is defined for each **`response`** with a valid schema.

In the **`Swagger Petstore 3.0`** example, the business service method **`getOrderById:get`** has a defined response schema for responses when the return code is 200. For the **`Swagger Petstore 3.0`** example, the import creates the following items:

- An output argument called: **`Res200:res`**

  The following figure indicates an output argument called: **`Res8SwaggerPetstoreUPOpenAPI30:res`**, and its data type is Integration Object.



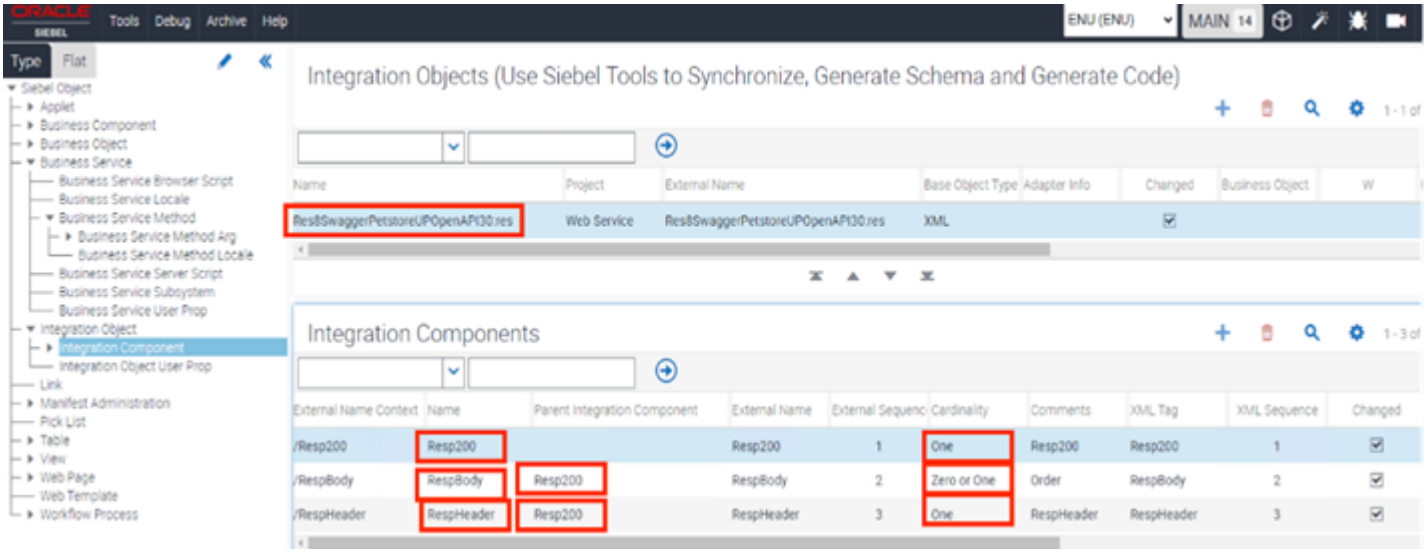
- A response integration object.

  This has a root integration component for the response code, in this example, **`Res200`**.

- A child integration component defining the response body as **`RespBody`**.

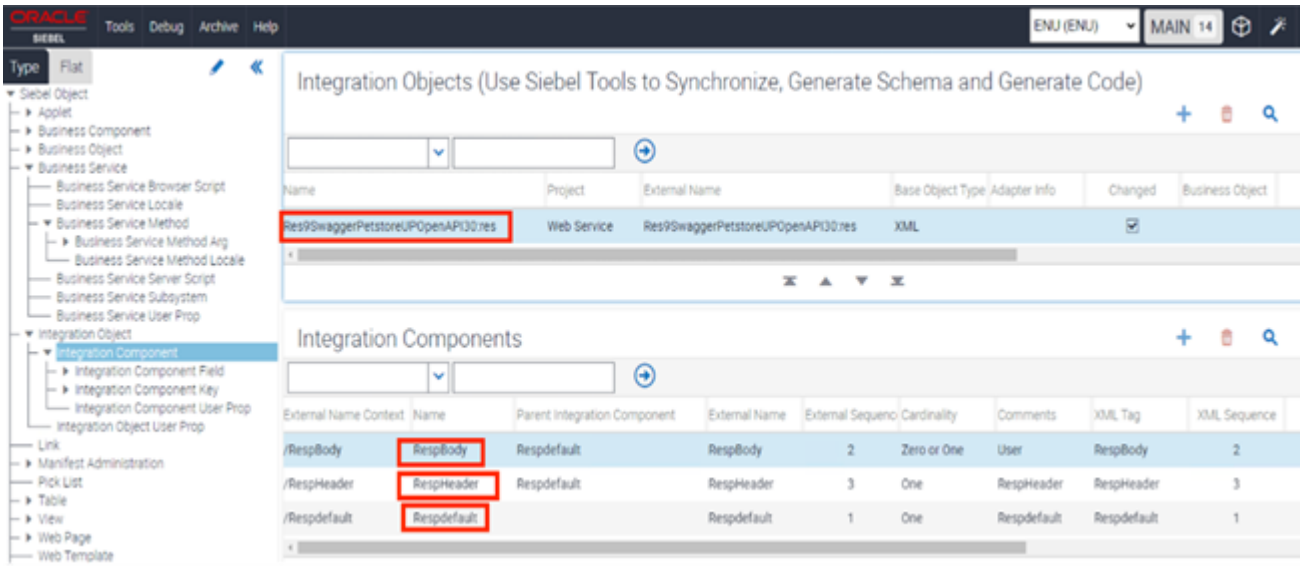
  This is located under the root integration component.

- A child integration component indicating the response header as **RespHeader**.



The following figure shows the response integration object called **Res9SwaggerPetstoreUPOpenAPI30:res** and its Root integration components **Respdefault**, and child integration component as **RespBody** and **RespHeader**. The response in this example (as shown below as **RespBody**) is **user**, and is displayed in the Comments field. It is the actual response received from the outbound call.

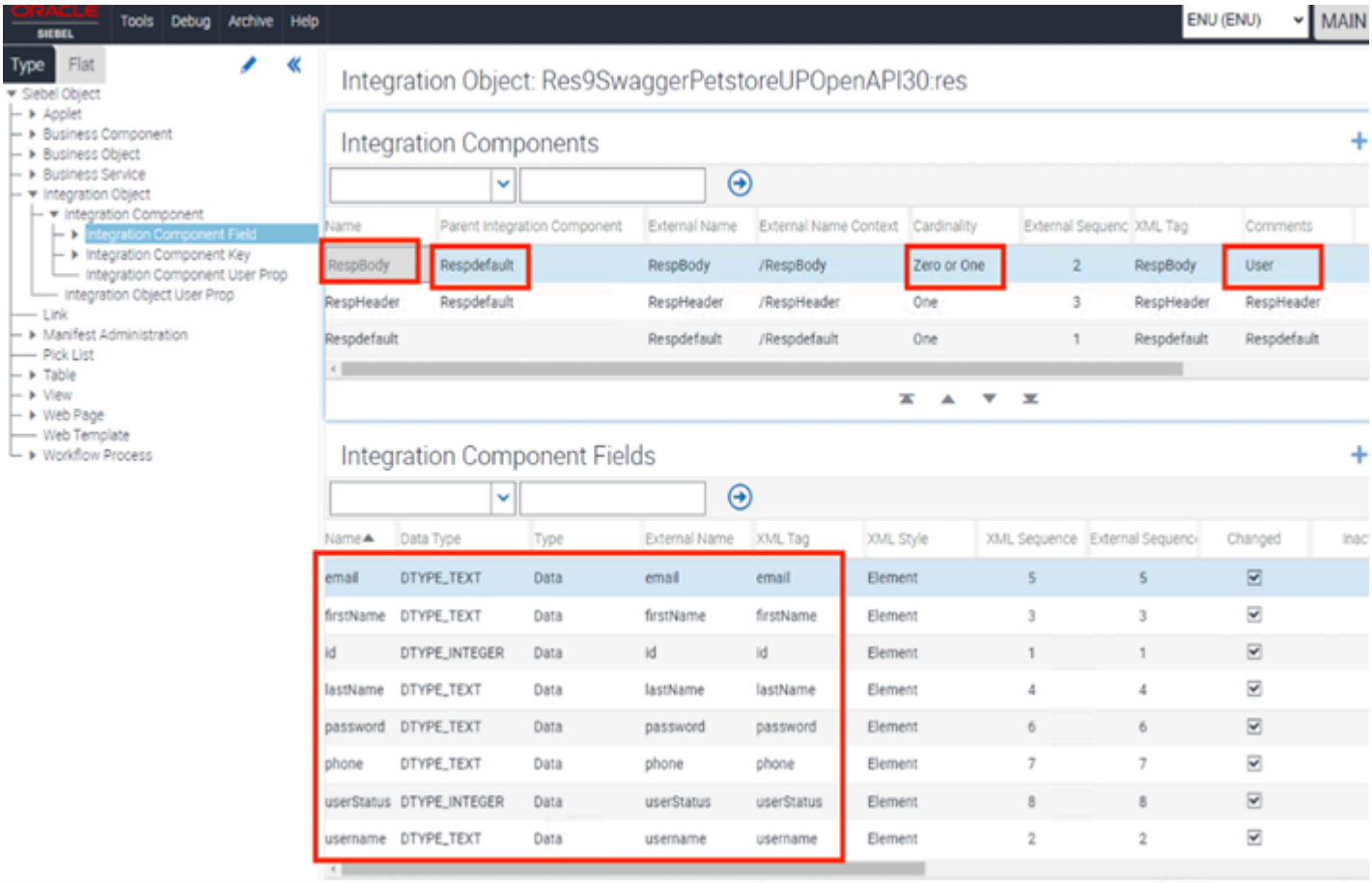
- If **user** is a non-array type, then its content is placed under **RespBody**.
- If **user** is an array of users, then each user is wrapped under **RespBody**. All **RespBody** integration components are then placed under **ListOfRespBody**, and **ListOfRespBody** is placed under **Respdefault**.



The response body integration component contains fields as specified in the OpenAPI definition. The following figure shows these integration component fields:

- email

- firstName
- id
- lastName
- password
- phone
- userStatus
- username



Note: If the JSON specification has no schema referenced under response code, then no response integration object is created. Also, at runtime, the JSON response is generically converted to a property set, and is returned to the caller.

Using the SiebelHierInput Parameter in a Request

If you want to allow additional integration objects or integration component wrappers in your request body, you can use the SiebelHierInput parameter. This is used in scenarios where you want to pass the output of other Siebel business services as an input to the Siebel REST Outbound service. In these cases there can be additional wrappers of type integration object or integration component provided in request body. Another possibility is a request that contains a single object type, provided as an array type, with the prefix ListOf. Alternatively, undefined integration components are provided in the integration object along with other valid defined integration component types. In these scenarios, you can set the SiebelHierInput parameter in the input, and specify the values Y, yes, or TRUE to intelligently interpret the request, as per the repository definition.

The XML example below shows an example of an input property set with additional and undefined elements, using the SiebelHierInput parameter. The following considerations apply:

- `<Invalid_Wrapper>`. This is an additional wrapper.
- `<Undefined_IO_or_wrapper>`. Another type of additional undefined wrapper.
- `<ListOfIO6SwaggerPetstore2>`. An integration object which is defined in the repository.
- `<undefined>`. This is another undefined wrapper used around a valid integration component category. `<undefined_type>` is skipped, and `<category>` is picked for processing.
- `<ListOftags>`. The `<ListOf>` wrapper is undefined because `tags` is an object not an array. The first object located immediately after it is processed by the Outbound service.

```
<?xml version="1.0" encoding="UTF-8"?>
<?Siebel-Property-Set EscapeNames="true"?>
<PropertySet password_clnsecurity="password" username_clnsecurity="username" type_clnsecurity="basic"
 SiebelHierInput="true">
 <Invalid_Wrapper>
 <Undefined_IO_or_wrapper>
 </Undefined_IO_or_wrapper>
 <ListOfIO6SwaggerPetstore2>
 <RootIC6 status="available" name="dog" id="1">
 <undefined_type>
 <category name="canine" id="2"></category>
 </undefined_type>

 <ListOftags>
 <tags name="tage1" id="2"></tags>
 <tags name="tage2" id="3"></tags>
 </ListOftags>

 <ListOfphotoUrls>
 <photoUrls elem__value="url1"></photoUrls>
 <photoUrls elem__value="url2"></photoUrls>
 </ListOfphotoUrls>
 </RootIC6>
 </ListOfIO6SwaggerPetstore2>
 </Invalid_Wrapper>
</PropertySet>
```

Based on this example, a sample JSON is as follows:

```
{
 "category":{
 "name":"canine",
 "id":2
 },
 "tags":{
 "name":"tag1",
 "id":2
 },
 "photoUrls":[
 "url1",
 "url2"
 ]
}
```

The second XML example below shows an example of an input property set with additional and undefined elements such as SiebelMessage, ListofWrappers, and uses the SiebelHierInput parameter. The following considerations apply:

- `<SiebelMessage>`. This is an additional wrapper.
- `<ListOfIO1TestPushCustomers>`. An integration object.

- `<ListOfPerson>`. The `<Listof>` wrapper is undefined because `Person` is an object not an array. The first object located immediately after it is processed by the Outbound service.
- `<ListOfName>`. The `<Listof>` wrapper is undefined because `UserName` is an object not array. The first object located immediately after it is processed by the Outbound service.

```
<?xml version="1.0" encoding="UTF-8"?>
<?Siebel-Property-Set EscapeNames="true"?>
<PropertySet
 SiebelHierInput="y"
 password_clnsecurity="123"
 username_clnsecurity="asd"
 type_clnsecurity="basic">
 <SiebelMessage
 MessageId="88-4JVO0"
 MessageType="Integration Object"
 IntObjectName="IO1TestPushCustomers"
 IntObjectFormat="Siebel Hierarchical">
 <ListOfIO1TestPushCustomers>
 <RootIC1>
 <ListOfPerson>
 <Person
 PersonId=""
 bookingId=""
 AddressId="">
 <ListOfUserName>
 <UserName
 GivenName=""
 Surname="">
 </UserName>
 </ListOfUserName>
 </Person>
 </ListOfPerson>
 </RootIC1>
 </ListOfIO1TestPushCustomers>
 </SiebelMessage>
</PropertySet>
```

Using the disableURLEncoding Parameter in a Request

If you want to disable default outbound URL encoding, you can use the disableURLEncoding parameter. This is used in scenarios where you want to provide a full or partially encoded outbound URL to make an outbound call without the REST Outbound framework encoding this URL. You can avoid double encoding by setting the value of the disableURLEncoding parameter to true.

Note: This disables default encoding of all elements of the URL including path and query parameters.

Consider the following XML example:

```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set
EscapeNames="true"?><PropertySet
 childlinks_clnquery="ADD2021%2F12"
 account_undkey_clnpath="ADD2021%2F12"
 disableURLEncoding="true"
 password_clnsecurity="mypassword"
 username_clnsecurity="myusername"
 type_clnsecurity="basic"
></PropertySet>
```

The path and query parameters in this example contain the user-encoded / (slash), this is displayed as: `%2F`

```
childlinks_clnquery="ADD2021%2F12"
```

```
account_undkey_clnpath="ADD2021%2F12"
```

In this example, the disableURLEncoding parameter is set to true. This disables the default URL encoding and avoids double encoding of already encoded characters. This XML example hits the target URL, however the URL is not encoded by the REST Outbound framework. Instead, all elements of the URL are passed unchanged:

```
http://ServerName:port/siebel/v1.0/data/Account/Account/ADD2021%2F12/?childlinks=ADD2021%2F12
```

Overview of Custom Headers in REST Outbound Proxy Business Service Methods

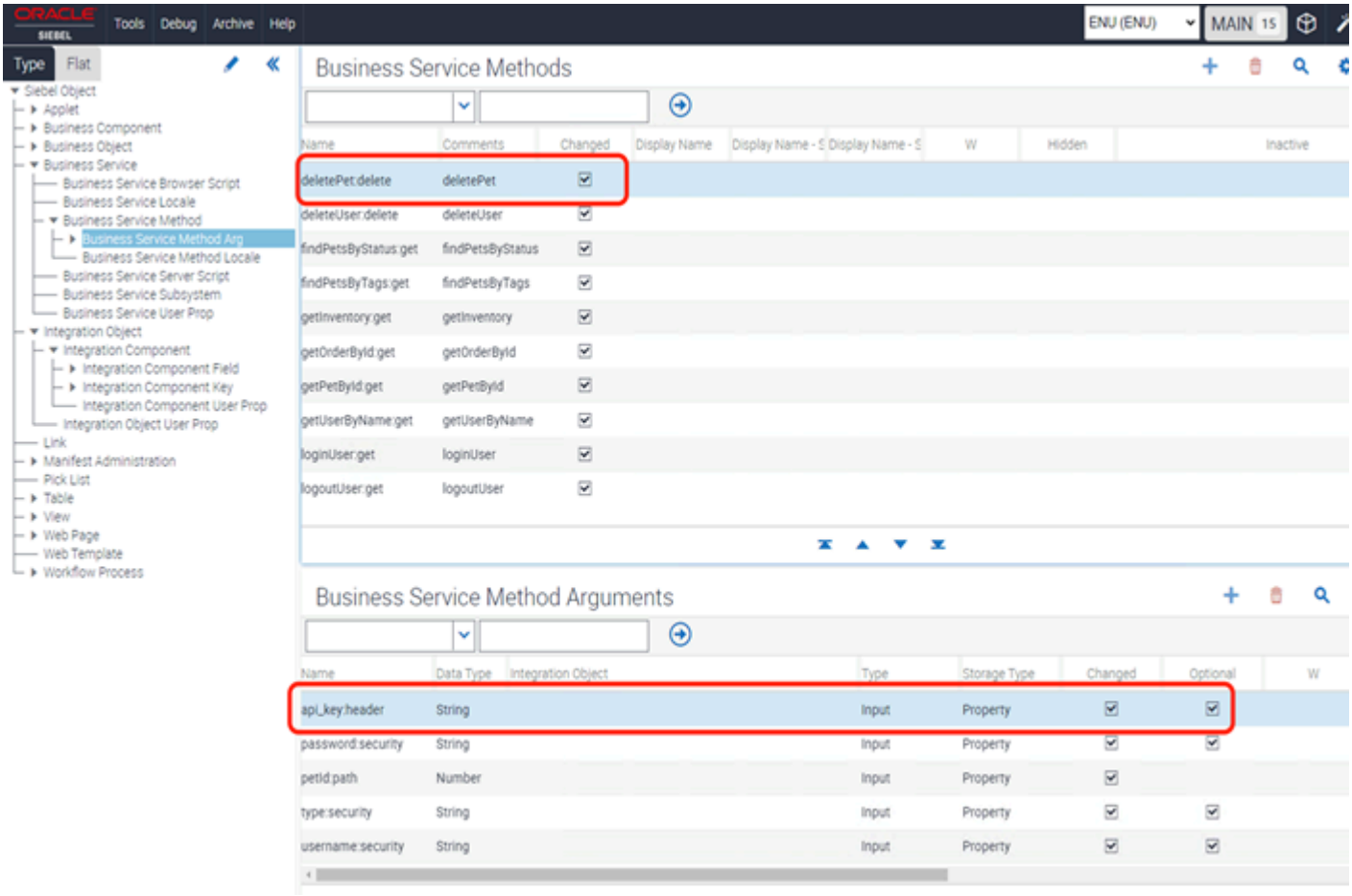
When you import the OpenAPI specification in the JSON format, the import creates proxy business service method arguments for all the parameters, including header parameters specified in the OpenAPI JSON. However, there may be some scenarios where the header parameters are not defined in the JSON specification, or where customers want to add some custom headers that are not part of JSON specification. In these scenarios, headers or custom headers can be added as method arguments. For more information, see *Adding Custom Headers in REST Outbound Proxy Business Service Methods*.

Business service method arguments take the following string format: `parameter-name:header`. The suffix `:header` specifies that this parameter goes into the header section when the REST Outbound http call is invoked.

Custom headers prefixed with `HDR.` , `HDR_`, or `siebel_transport_header` can be sent directly without requiring a corresponding entry in the repository definition. The following XML example shows the `HDR.` , `HDR_`, and `siebel_transport_header` headers used in an input `<PropertySet>`. The headers `api_key:header` and `customheader:header` remain in the repository, however `HDR_headerName1`, `HDR.headerName2` and `siebel_transport_header:headerName3` are not part of the repository definition. During runtime all these headers are processed and become part of the request header.

```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set EscapeNames="true"?>
<PropertySet
 type_clnsecurity="basic"
 username_clnsecurity="username"
 password_clnsecurity="password"
 api_undkey_clnheader="abc"
 HDR_undheaderName1="HDR_value"
 HDR.headerName2="HDR.value"
 siebel_undtransport_undheader_clnheaderName3="siebeltransporheaderval"
 customheader_clnheader="headervalue"
>
</PropertySet>
```

The following figure shows an example of an existing parameter, a business service method argument called `api_key`, with the word header appended, making it `api_key:header`.



You can append other parameters to REST HTTP requests, for example:

- Path parameters, for example: `api_key:path`

  This requires the address to have a variable named with parameter name. For example, `https://somehost:80/sampleuri/{api_key}`
- Input parameters, for example: `api_key:body`
- Header parameters, for example: `api_key:header`

Adding Custom Headers in REST Outbound Proxy Business Service Methods

There may be some scenarios where the header parameters are not defined in the JSON specification, or where customers want to add some custom headers that are not part of JSON specification. In these scenarios, headers or custom headers can be added as method arguments. For more information, see *Overview of Custom Headers in REST Outbound Proxy Business Service Methods*.

To add a new custom header parameter:

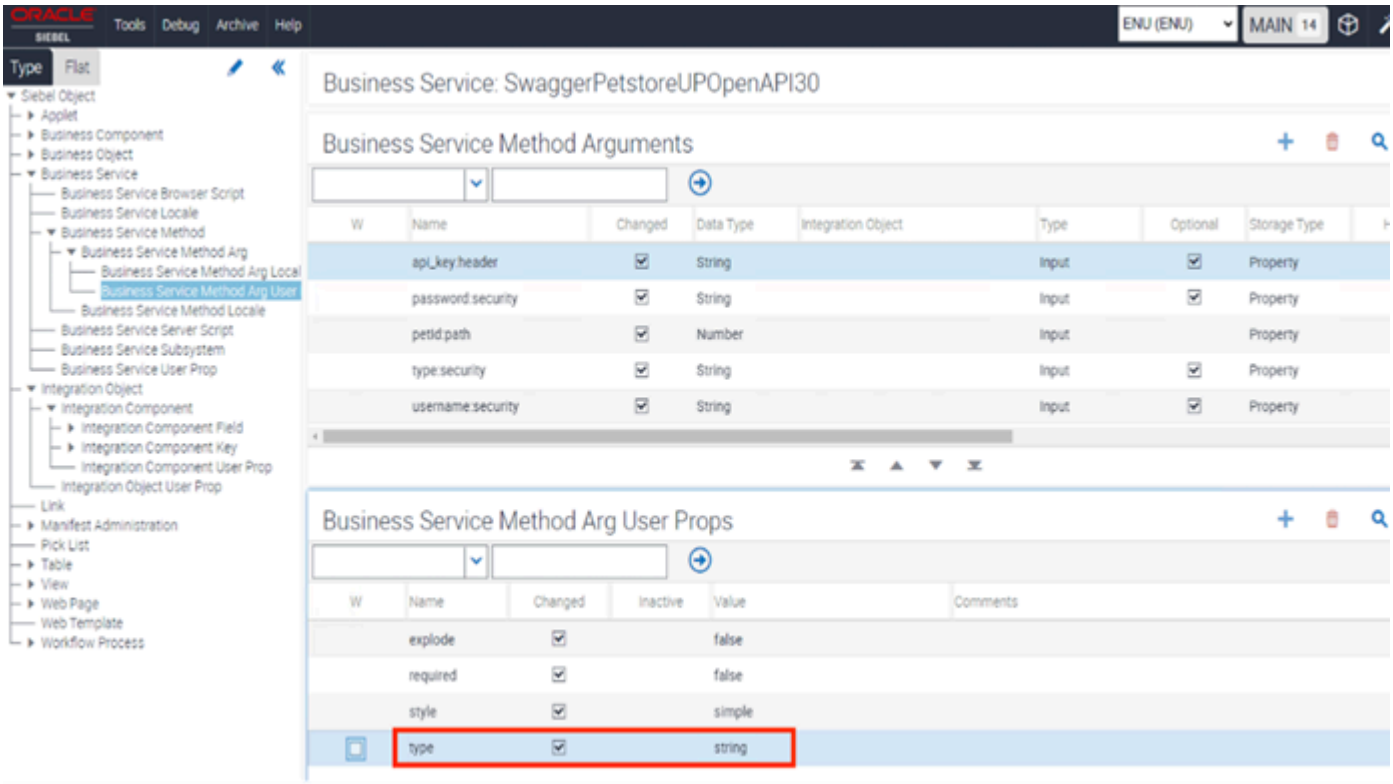
1. Create a new Workspace.

2. In your list of pre-existing proxy business services, select a required proxy business service, select required method, and then locate your method arguments.
3. Add a new argument record giving it an appropriate name to the new header argument.
4. Append the suffix `:header` to the parameter name.

   For example, if the parameter name is `myFirstHeader`, then the complete parameter name is: `myFirstHeader:header`.
5. Update the values in the Type column to specify whether your header is an input header or an output header.
6. (Optional) Update the values in the Storage Type to Property.
7. To the new method argument, add a use property with the following details:
   - **Name.** Type
   - **Data type.** This can be string, integer, or boolean and must be entered in lowercase.

     > **Note:** Data type is required when assigning this parameter value to the outgoing JSON payload. The value is usually string for headers.

   The following figure shows an example of a Business Service Method Argument User Property called `type`. It has the value `string`, this indicates that the `api_key:header` is of type string.



> **Note:** There is no direct mapping between a Siebel data type and a JSON data type, therefore Siebel REST API uses this user property to specify what JSON data type to use when sending the HTTP request.

8. Save the new argument and its user property, and deliver the workspace.

   You can now pass values to this new method argument at runtime. This argument is converted into a header parameter during the outbound REST endpoint call.

Using Existing Integration Objects in Outbound REST Services

You can integrate Siebel modules with REST outbound module for Siebel REST requests of type Business Service, where the Integration Object that needs to be interfaced is the same. You can use existing Siebel Integration Objects, instead of those created by the Web Services Import Wizard, when creating a Siebel-to-Siebel outbound REST integration. This removes the necessity of creating data mapping for integrations, as it uses existing Siebel Integration Objects. You can also use this method of integration for business services that have simple properties defined at method argument level or have a combination of simple properties and Integration Objects.

To use existing Integration Objects in Outbound REST services, you must:

1. Generate the input schema using an OpenAPI (swagger) description for the business service with the `MatchRequestFormat` flag enabled. For example: `https://www.mysite.com/siebel/v1.0/service/PDS Order/QueryByExample/describe?matchrequestformat=y`
2. Import it in Web Tools to create the repository artefacts, such as proxy Business Service and Integration Objects.
3. Replace the wizard-generated Integration Objects with the Integration Objects that are already available in Siebel repository in the appropriate Business Service method argument.
4. Replace the business service method argument name with the actual type of the Siebel hierarchy.

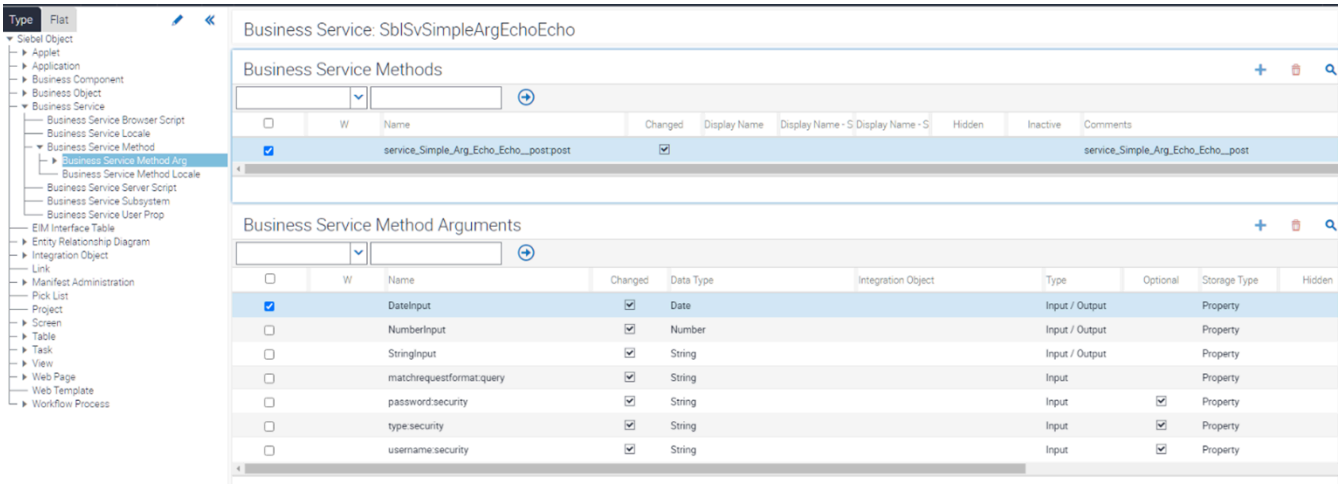
Types of JSONs Supported

This feature supports simple JSONs and complex JSONs of Siebel hierarchy type. For detailed information see:

- *Simple JSON Example*
- *Complex JSON Example*

Simple JSON Example

In this example, Simple Arg Echo business service is imported in Web Tools in REST outbound wizard. It generates the proxy Business Service and the Integration Objects. You need to replace the tool generated Integration Objects with simple arguments available in the original business service, such as DateInput, NumberInput, StringInput, and specify the appropriate data type and input/output type. Once the workspace is delivered and the proxy Business Service is triggered, the implementation relies on the argument names, which do not have the colon sign (:), to recognize and align the input and output property set processing as per the Siebel standards. This helps the implementation to directly consume the property set generated from another Siebel module as input to the REST Outbound service or use the result generated from the REST Outbound for further processing of another Siebel module without having to create any data mapping.



Handling Data Types

Business Service argument data types are used to derive the data type of JSON value in the request. String and Date data types are converted to String in JSON. Number is retained as Number in JSON data type.

Example of Input/Output property set for proxy Business Service of type "Simple Arg Echo" for Echo

Property Set Request Example:

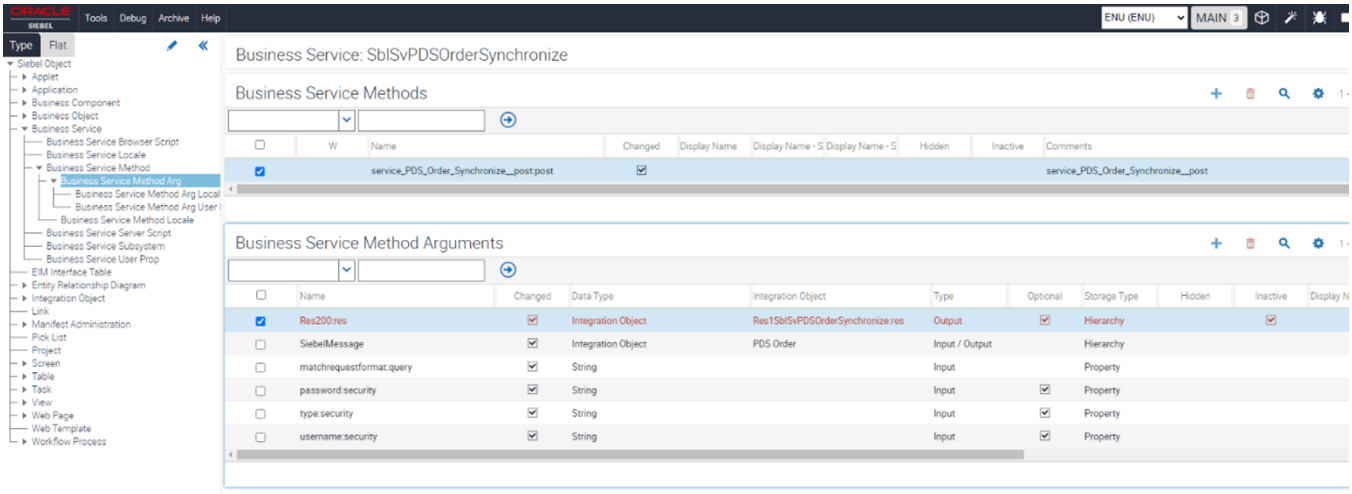
```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set EscapeNames="true"?><PropertySet
 matchrequestformat_clnquery="Y" password_clnsecurity="sadmin"
 username_clnsecurity="sadmin"
 type_clnsecurity="basic"
 StringInput="test"
 NumberInput="1"
 DateInput="09/03/2014 23:03:38"
></PropertySet>
```

Property Set Response Example:

```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set EscapeNames="true"?>
<PropertySet
 DateInput="09/03/2014 23:03:38"
 StatusCode="200"
 NumberInput="1"
 StringInput="test"
></PropertySet>
```

Complex JSON Example

In this example, PDS Order BS Synchronize method is imported in Web Tools in REST outbound wizard. It generates the proxy Business Service and Integration Objects. You need to replace the tool generated Integration Objects with the original PDS Order Integration Object and specify the appropriate data type and input/output type. Business Service method argument name should be the Siebel hierarchy type needed in the input or output. Note that the argument name doesn't have colon sign (:) here, which is used as an indicator to process it as a Siebel Integration Object.



Handling Data Types

For the Integration Object provided in the input, Integration Component field data type is used to derive the data type of JSON value in the request. Here's the conversion to compute the JSON value datatype from the Siebel data type.

- TYPE_BOOL → Boolean
- DTYPE_INTEGER → Integer
- DTYPE_NUMBER → Number
- DTYPE_ID / DATE / TEXT / [all other types] → String

Support for Special Parameters

Apart from normal property sets, this implementation supports additional parameters at Integration Object level or at the ListOf(array) level for features like pagination. This implementation also supports hierarchical Integration Components which are there in PDS Order Integration Object and others.

Example of Input/Output property set for proxy Business Service of type "PDS Order" Synchronize method

Property Set Request Example:

```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set EscapeNames="true"?><PropertySet
 matchrequestformat_clnquery="Y" password_clnsecurity="sadmin"
 username_clnsecurity="sadmin"
 type_clnsecurity="basic">
 <SiebelMessage
 MessageId="88-4XL9E"
 MessageType="Integration Object"
 IntObjectName="PDS Order"
 IntObjectFormat="Siebel Hierarchical">
 <ListOfPDS_spcOrder>
 <Header
 Account_spcLocation="SHIP_FROM"
 Comments=""
 Id="88-3TSBB"
 Revision="1"
 Service_spcAccount_spcId="88-3TS9P"
 ..............................
 Compound_spcProduct_spcNumber="88-3TSBB"
 Price_spcList_spcId="88-17KYJ">
 <ListOfLine_spcItem>
 <Line_spcItem
 Related_spcAsset_spcIntegration_spcId=""
```

```
Eligibility_spcReason=""
Extended_spcQuantity_spcRequested="1"
Need_spcRefresh="Y"
.........................>
<Line_spcItem
Related_spcAsset_spcIntegration_spcId=""
Eligibility_spcReason=""
Extended_spcQuantity_spcRequested="1"
Need_spcRefresh="Y"
.........................>
</Line_spcItem>
<Line_spcItem
Related_spcAsset_spcIntegration_spcId=""
Eligibility_spcReason=""
Extended_spcQuantity_spcRequested="1"
Need_spcRefresh="Y"
.........................>
<ListOfXA>
<XA
Parent_spcId="88-3TSEV"
Value="Microsoft"
Cfg_spcState_spcCode="Engine Picked Item"
Unit_spcof_spcMeasure=""
Id="88-3TSEW"
.........................>
</XA>
<XA
Parent_spcId="88-3TSEV"
Value=""
Cfg_spcState_spcCode="Engine Picked Item"
Unit_spcof_spcMeasure=""
Id="88-3TSEX"
.........................>
</XA>
</ListOfXA>
</Line_spcItem>
</Line_spcItem>
</ListOfLine_spcItem>
</Header>
</ListOfPDS_spcOrder>
</SiebelMessage>
</PropertySet>
```

Property Set Response Example:

```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set EscapeNames="true"?><PropertySet
StatusCode="200">
<SiebelMessage
MessageId="88-4XL9E"
MessageType="Integration Object"
IntObjectName="PDS Order"
IntObjectFormat="Siebel Hierarchical">
<ListOfPDS_spcOrder>
<Header
Account_spcLocation="SHIP_FROM"
Revision="1"
Id="88-3TSBB"
Status="Pending"
...............>
<ListOfLine_spcItem>
<Line_spcItem
Eligibility_spcReason=""
Extended_spcQuantity_spcRequested="1"
Need_spcRefresh="Y"
Discount_spcPercent=""
.................>
```

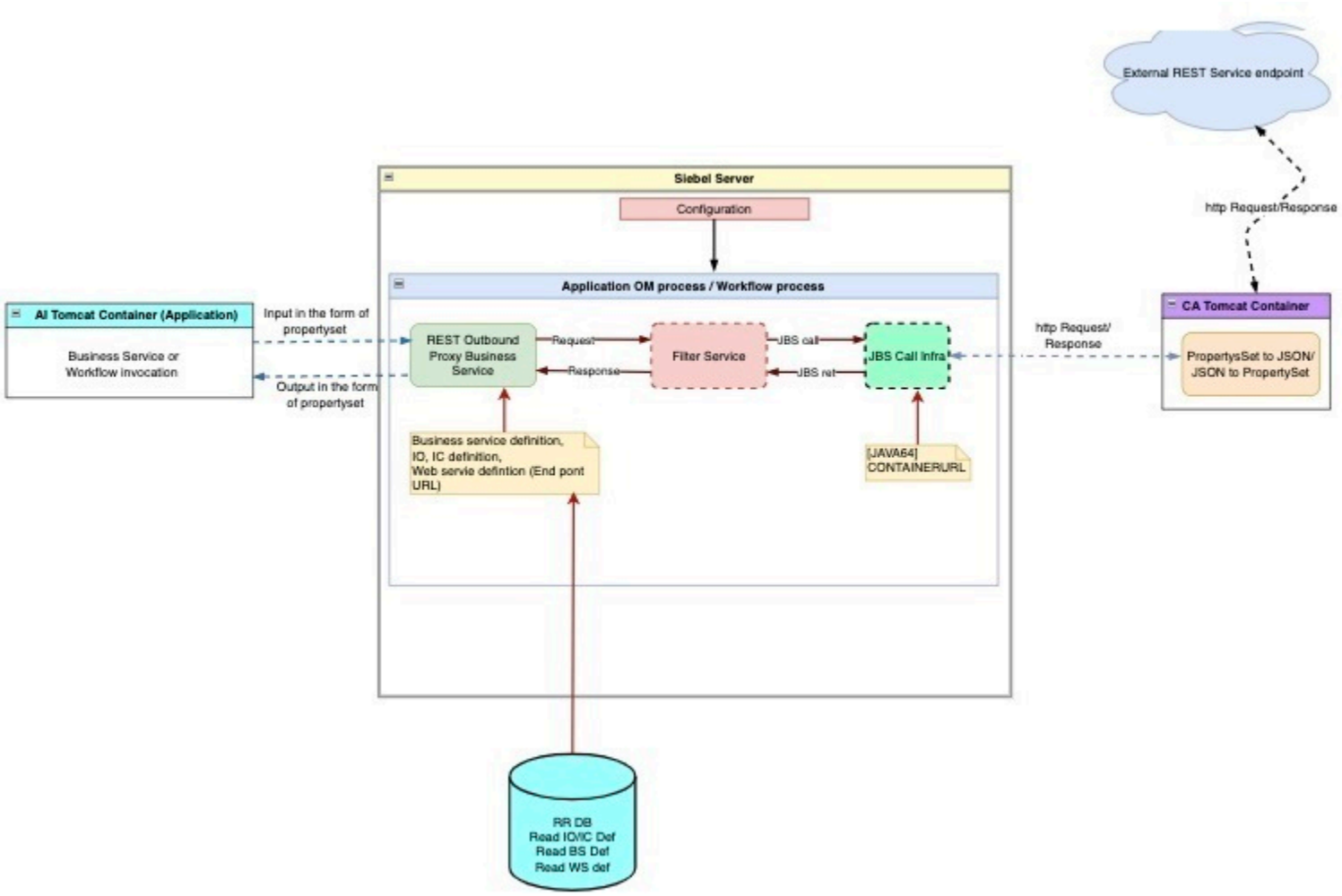
```
<Line_spcItem
Extended_spcQuantity_spcRequested="1"
Eligibility_spcReason=""
Need_spcRefresh="Y"
Current_spcPrice=""
.....................>
</Line_spcItem>
<Line_spcItem
Extended_spcQuantity_spcRequested="1"
Eligibility_spcReason=""
Need_spcRefresh="Y"
Current_spcPrice=""
Discount_spcPercent=""
.........................>
<ListOfXA>
<XA
Parent_spcId="88-3TSEV"
Value="OM_FULL"
Unit_spcof_spcMeasure=""
Id="88-3TSF4"
.........................>
</XA>
<XA
Parent_spcId="88-3TSEV"
Value="1"
Unit_spcof_spcMeasure=""
Id="88-3TSF3"
.......................>
</XA>
</ListOfXA>
</Line_spcItem>
</Line_spcItem>
</ListOfLine_spcItem>
</Header>
</ListOfPDS_spcOrder>
</SiebelMessage>
</PropertySet>
```

HTTP Status Response

The response generated has the HTTP status of the REST outbound request. It is available in property set with property name: `StatusCode`.

Invoking the External REST Service Endpoint Using Proxy Business Service at Runtime

This topic describes the steps involved when invoking an external REST service endpoint as a business service from a Siebel application or workflow.



1. **Obtain the OpenAPI compliant JSON file.** This step requires that you obtain the external JSON file or URL, and import it into Web Tools using the WSDL/JSON wizard. This creates the proxy business service, integration objects and Web service.

   For more information, see *Creating an Outbound REST Service Based on an OpenAPI Compliant JSON File*.
2. **Prepare the input property set.** This step requires that you prepare your input property set for input to the proxy business service. This depends on the definition of the proxy business service in Web Tools. Inputs can be simple, such as strings, or integers, or they can be complex types such as integration objects.
3. **Set up the Filter Service (Request Filter and Response Filter).** In this step, you can use the Filter Service to:

   - Customize the Request/Response headers
   - Add security parameters for different authentication mechanisms
   - Specify a different Filter Business Service for each Service Method
   - Specify a different Filter Business Service for Request and Response

   You can make these changes in Application Administration > Web Services > Outbound REST Services UI. Following is a bare bones example of two methods: UpdateInput and UpdateOutput:

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
```

```
 if(MethodName == "UpdateInput")
 {
 Inputs.SetProperty("REQFILTER","Request Filter working");
 }

 if(MethodName == "UpdateOutput")
 {
 Outputs.SetProperty("RESPFILTER","Response Filter working");
 }
 return (CancelOperation);
}
```

4. **Application or script or workflow invokes the proxy business service.** In this step, the proxy business service is invoked just like any other business service and does the following:
   - It uses the Business Service definition to validate the inputs.
   - It uses the Web Service definition to obtain the endpoint URL and other details.
5. **Proxy business service calls the Config Agent (CA) Tomcat.** In this step, the proxy business service makes a Java Business Service call to the Config Agent (CA) Tomcat attached to the Siebel server. This executes the Java Business Service. The address of the CA Tomcat is obtained from the 64-bit Java subsystem attached to the Object Manager, passing the property set input obtained from the user.

   The Java Business Service does the following:

   a. It converts the property set to JSON payload.
   b. It makes an HTTP call to external end point URL, obtaining the response, and then converting the JSON response back to a property set, and finally returning to the Proxy Business Service
6. **Proxy business service converts the property set to output.** In this step, on receiving the property set response from Java Business Service, the proxy business service converts it back to the output arguments according to its method's out parameters.

Note: If the external endpoint requires the caller to trust the endpoint, then you must import the CA certificate of the endpoint to the trust store of the CA tomcat. If the call to external endpoint requires access to any proxy, then you must add the proxy settings to the CA tomcat service. For more information on installing certificate files, see *Siebel Security Guide* .