Oracle® Retail Brand Compliance Management Cloud Service

Implementation Guide

Release 16.0

E84407-01

June 2017

Oracle Retail Brand Compliance Management Cloud Service Implementation Guide, Release 16.0

E84407-01

Primary Author: Bernadette Goodman

Contributing Author: Aidan Ratcliffe, Simon Tucker, Lloyd Colling

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via**™ licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex**™ licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR

Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

3 APIs Available in Brand Compliance

4 RESTful APIs

5 SOAP APIs

A Appendix: Secure Development Guide

Glossary

Send Us Your Comments

Oracle Retail Brand Compliance Management Cloud Service Implementation Guide, Release 16.0

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our web site at `http://www.oracle.com`.

Preface

This Implementation Guide describes the Application Programming Interfaces (APIs) available for this Oracle Retail Brand Compliance Management Cloud Service release.

Audience

This Implementation Guide is intended for the users of the Oracle Retail Brand Compliance Management Cloud Service application integration and implementation staff, and other parties intending to integrate with Brand Compliance Management Cloud Service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Retail Brand Compliance Management Cloud Service Release 16.0 documentation set:

- *Oracle Retail Brand Compliance Management Cloud Service Administration Guide*
- *Oracle Retail Brand Compliance Management Cloud Service Release Notes*
- *Oracle Retail Brand Compliance Management Cloud Service User Guide*

For information on the Oracle Retail Brand Compliance Management Cloud Service modules, see the following documents:

- *Oracle Retail Brand Compliance Management Cloud Service Product User Guide*
- *Oracle Retail Brand Compliance Management Cloud Service Project User Guide*
- *Oracle Retail Brand Compliance Management Cloud Service Reports User Guide*
- *Oracle Retail Brand Compliance Management Cloud Service Supplier User Guide*

Data Dictionary

For information on the content of the Oracle Retail Brand Compliance Management Cloud Service data records, see the associated Data Dictionary documents:

- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 5 - Product (CNF Specification)*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 6 - Product (FNF Specification)*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 7 - Product (BWS Specification)*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 8 - Product (Produce Specification)*
- *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 9 - Project*

These Data Dictionary documents are available through My Oracle Support:

Oracle Retail 16.0.x Data Models (Doc ID: 2200398.1)

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times **not** be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of an document with part number E123456-01.

If a more recent version of the document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following web site:

`http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html`

(Data Model documents are not available through Oracle Technology Network. You can obtain these documents through My Oracle Support.)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

Oracle Retail Brand Compliance Management Cloud Service (Brand Compliance) is an application designed to meet all aspects of sourcing, developing, and protecting retailer brands. The application provides solutions for product development, compliance, quality, and traceability. It is designed specifically for retail, food service, and manufacturing businesses to develop and protect their brands, manage their suppliers, and ensure full end-to-end product lifecycle management.

Brand Compliance includes a number of exposed Application Programming Interfaces (APIs) that can be called by external systems. This document details all of the exposed APIs and the information required in order to access them.

An API is a way of exchanging data between computer systems. The owners of an external system can interact with Brand Compliance by developing the necessary functionality to trigger the web service submissions, and handle the returned messages from Brand Compliance.

In order to access Brand Compliance using its APIs, the portal owner must grant the external system access. This is achieved by creating an External Systems account in the Admin area and assigning the unique access credentials in the form of a login ID and password.

Contents of this Guide

This implementation guide addresses the following topics:

- Chapter 2, "API Overview and Architecture": Overview of the Brand Compliance APIs and architecture used.
- Chapter 3, "APIs Available in Brand Compliance": Summary of the APIs, including the release in which the API is available and the type of API.
- Chapter 4, "RESTful APIs": Details of the available APIs that are based on the RESTful architecture.
- Chapter 5, "SOAP APIs": Details of the available APIs that are based on the SOAP architecture.
- Appendix A, "Appendix: Secure Development Guide": Description of some best practices for web service use.

Key Features of Brand Compliance Management Cloud Service

The application is composed of the following modules:

- Library enables the issue, receipt, and acceptance of policies, guidelines, and key working documents.
- Product supports the development of products and production specifications.
- Project supports the development of project briefs, plans, and workflow management.
- Supplier enables the identification, selection, and approval of suppliers.

2 API Overview and Architecture

The APIs exposed by Brand Compliance are implemented in one of two formats, SOAP web services or RESTful web services. In both cases, requests and responses are normally in an XML format.

The APIs are already built and available for clients to build services for their external applications to utilize. The external system makes a call to Brand Compliance using HTTPS. If the external system is validated (that is, it is enabled in the Admin area of Brand Compliance and the login user ID and password match), the call is accepted.

Calls will typically allow for the retrieval, creation, update, and in some cases deletion, of data through the API. Each call is logged in the Web Service Log in Brand Compliance, with the request and response messages being recorded as appropriate.

A call may contain a number of parameters or filters to determine the records to be returned or updated, or to identify where to create a new record, such as to filter Product Specification of a certain type, status and so on.

SOAP and RESTful APIs

The Brand Compliance APIs are web services in both SOAP and RESTful architectural styles. In general, the SOAP services are developed for specific requirements or are historical services. Unless there is a definite requirement for a SOAP implementation of a service, any new services or new versions of services are implemented using the RESTful architectural style.

Request and Response XML

In general, APIs exposed by Brand Compliance as web services use XML messages for both requests and responses. Exceptions to this are RESTful services where the use of an XML request is replaced by the use of URI parameters in a GET call. The APIs do not support other message formats, such as JSON.

The XML data schemas are written to allow for both forwards and backwards compatibility. All elements within the XML schemas are marked as optional. The element names are ordered so that new fields are added after existing elements and, within each sequence of elements, the last element is always an *any* element so that fields added in future versions are ignored by clients that are using an old version of the schema. For example:

```
<xs:complexType name="fpsMicroTestLocaleDataFullDTO">
    <xs:sequence>
        <xs:element form="qualified" minOccurs="0" name="description"
type="xs:string"/>
        <xs:element form="qualified" minOccurs="0" name="locale"
```

```
type="xs:string"/>
        <xs:element form="qualified" minOccurs="0" name="id" type="xs:long"/>
        <xs:element form="qualified" minOccurs="0" name="createdOn"
type="xs:string"/>
        <xs:element form="qualified" minOccurs="0" name="updatedOn"
type="xs:string"/>
        <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
```

The element names in the XML Schemas are defined by the core system and do not change in specific implementations. If an implementation has additional fields that have been added to the system (the implementation is non-core), those fields may be exposed after the core fields (before the xs:any element) and will incur a maintenance overhead in later versions to cope with any additional core fields added after the custom fields.

Data types in the XML Schemas are the standard datatypes for XSDs, WSDLs, and WADLs. Values should be entered as described on the following web site:

http://www.w3.org/TR/xmlschema-2/

The following table lists the common data types used in the Brand Compliance schemas:

Data Types

Type	Example
xs:string	A normal string
xs:string	<![CDATA[A String with <br/> html data]]>
xs:long	9223372036854775807
xs:int	2147483647
xs:short	-32768
xs:boolean	true or false
xs:boolean	1 or 0
xs:date	2015-01-30
xs:dateTime	2015-01-30T23:59:59

Versioning

Due to the format of the XML schemas as described above, most revisions of the web services should be forwards and backwards compatible to allow clients of those web services to be upgraded and developed without binding the lifecycle of the client services to the upgrade cycle of Brand Compliance.

Where the necessary changes to a service cannot be made compatible with previous versions, a new endpoint will be created to expose the changed service. If possible, the previous version of the service will remain available at the original endpoint to enable existing and legacy clients to continue to work, but will be noted as deprecated in documentation to encourage developers of clients to the services to avoid using it.

Logging

All calls to APIs exposed as web services are logged within the application and can be viewed by logging into Brand Compliance as a Power User, navigating to Company > Admin> Notifications, and choosing the Web Service Log option. For each call, the name of the web service, date and time of the call, status (in progress/completed/failed), duration of the call (not available for In Progress calls), and a list of error messages if the call failed are shown.

The request and response XMLs are attached to each web service log entry. If a request does not contain XML, no attachment is created.

Accessing the APIs

This section describes how to access the APIs.

Availability

If enabled within the portal, an API is accessible while the portal is online. The support for a portal's availability is subject to the client's service level agreement.

Updates to the APIs are applied as per changes and fixes to the application. Each release involves a scheduled period of downtime. Details of the change and impact are covered in the supporting Release Notes.

Note: After a Brand Compliance upgrade release, it may be necessary to change the URLs that the external system uses to call the APIs.

Deployment Model

The APIs, made available by Brand Compliance, are all web services implemented using either a SOAP or RESTful architecture.

The web services are exposed over HTTPS and require HTTP Basic authentication in order to access the service requested.

Authentication

In order to authenticate with Brand Compliance when accessing an API deployed as a web service, a user name and password must be supplied for HTTP basic authentication. Within Brand Compliance, the External System record defines the user names and passwords that have access to the web services.

To see the list of existing External System records, log in to Brand Compliance as a Power User, navigate to Company > Admin > Roles and Permissions, and select the External Systems option.

To create a new External System:

1. Select the Actions > New action to open up a tab for creating an External System.
2. Enter a Login ID (User Name) and select the Actions > Save option.
3. To set the password to be used by the External System, select the Actions > Change Password option.
4. Select the Actions > Save action. The new user name and password are active and available for use.

The passwords used for External Systems are validated against the same criteria as the passwords for interactive users of Brand Compliance. It is likely that a minimum length, mixed case, and digits will be required for the password, depending on the setup of the system.

Authorization

At present, all APIs are available to all authenticated requesters.

Listing Available APIs

In all installations of Brand Compliance, a full list of the deployed web services can be found at /services.

For example, for a system where the application is deployed to URL https://www.example.com/brandCompliance, you can retrieve a list of all services from https://www.example.com/brandCompliance/services.

Note: In order to access the /services page, the user must first be logged in/authenticated with the Brand Compliance portal.

When downloading the WSDL or WADL files, if prompted for further credentials, enter the Login ID and Password of the External System account.

Security Considerations

This section provides information on aspects of security you need to consider.

Password Management

Unlike interactive logins for users, the use of web services does not permit warnings to be included in the returned messages. Also, there is no guarantee that there is someone handling the call in order to interpret any warnings about password expiry. Therefore, Brand Compliance does not currently apply the password lifetime rules to External Systems.

Note: It is recommended that the passwords used for accessing APIs deployed as web services are changed regularly and that passwords are not reused or changed in such a way that the next value is predictable, such as, do not change from Password11 to Password12.

The passwords used to access the Brand Compliance APIs should be stored securely to prevent potential attackers from being able to masquerade as an authorized system.

Rich Text Data

In Brand Compliance, certain fields allow the display of HTML-formatted data. Such data is cleansed before display in Brand Compliance to ensure that any attempts at Cross-Site Scripting (XSS) attacks are mitigated, but this cleansing is not applied within the APIs. Therefore, it is advisable to cleanse any HTML-formatted data before it is displayed to users.

Excluded Fields

Many of the APIs published by Brand Compliance provide access to manage all of the records of a given type. With certain records, the fields accessible through the API are a subset of the fields within Brand Compliance. Any updates leave those fields unchanged. An example of this is the password-related fields on the User record, none of which are exposed in the APIs.

Secured Connections

As all of the APIs are exposed as web services over HTTPS, it is necessary for callers to communicate over SSL/TLS using the ciphers that the Brand Compliance servers accept. To ensure optimal security, it is advised that standard certificate checks are in place and not disabled. In the case of Apache CXF, the Client TLS Parameter disableCNCheck should be left as false.

Change History

Updates made to Brand Compliance records through the APIs are shown in the record's Change History log with user shown as System.

URL Encoding

Certain characters within URLs are reserved for a special meaning. In order to avoid calls that contain these characters being blocked by the API's security, the characters must be encoded and passed as plain text.

The reserved characters include ";", "/", "?", ":", "@", and "=", all of which could be present in the parameters passed to Brand Compliance APIs - such as within email addresses or within the names of users, products, or suppliers.

UTF-8 URL encoding should therefore be applied to all calls to the Brand Compliance APIs. The APIs will automatically decode the values on receipt.

For example, an un-encoded call of:

```
…/services/rest/user/byKey/john.smith@example.com
```

should be encoded as:

```
…/services/rest/user/byKey/john.smith%40example.com
```

If a call comprises multiple values, each must be encoded individually.

Key Features of the API Process

This section summarizes the key aspects of the API process:

- The external system makes a call to Brand Compliance using HTTPS. The external system must be registered in Brand Compliance, and the user ID and password must match.
- Calls to RESTful services use the GET, POST, PUT, and DELETE methods to retrieve, create, update, and delete records respectively. Calls to SOAP services use a predefined set of methods to perform the operations.
- Each call is logged in the Web Service Log in Brand Compliance. The request and response XML messages are attached to the logs.
- A predefined set of parameters is available to apply filters to the call, to retrieve a specific record or set of records.

- Data is returned as XML, as defined by the associated WADL or WSDL schema. Only elements that contain data are returned; empty elements are omitted.
- For RESTful APIs, an initial GET call to the service returns the unique internal key ID of the returned records in either the recordId or the id element. The record ID is then used in a subsequent call using a URI to retrieve the record's full XML data set, or to perform an update operation.

  Update functions require the full set of fields and their values to be passed, as the full contents of the record are replaced. If all fields are not present in the external system, the update process would need to first retrieve the current contents of the record from Brand Compliance before applying the changes to that data set, and submitting the full data set as the update.
- There could be several initiators for the data exchange, including the following:
  - Human initiation, where an individual makes the request, such as by clicking a button to run the request.
  - End user action, where the completion of an activity in the external system triggers the exchange.
  - Scheduled request (or polling), where the external system sends or checks for data on a regular basis. For example, it may be looking for newly created Product Records or Specifications with a change in status.

  For security purposes, the APIs rely on the external system polling, retrieving or submitting calls to the APIs. Brand Compliance does not push data externally.
- The nature of the APIs may require multiple requests to be made, for example:
  - Request a list of Product Records with their IDs.
  - Request the timestamp reference for each Product Record.
  - Identify any Product Records changed or created within a given time scale.
  - Request the full details for that Product Record.

These types of requests can be automated. The external system is altered to enable the operations to take place in sequence according to a pre-defined algorithm.

Certain modifications can also be made to the external system to enable the storage of data returned from Brand Compliance. For example, if the external system requests a record to be created in Brand Compliance, a confirmation response is sent, containing the record ID. This ID may then be stored in the external system for future reference.

3

APIs Available in Brand Compliance

The APIs deployed in a particular implementation of Brand Compliance can be found at /services: https://<*hostname*>/<*context*>/services.

Individual Services

The following table lists the individual services available in Brand Compliance, indicating at which version each was introduced and, where appropriate, at which version it was deprecated.

Brand Compliance API Services

Service Name and Functions	Description	Type	From Version	To Version
User Details				
userDetails	Manages single sign-on between the main Brand Compliance application and the Reports module. This is superseded by the SSO Restful Service.	SOAP	1.0	15.0
Product Service				
createProduct	Creates a new Product Record.	SOAP	1.8	18.0
updateProduct	Updates existing Product Records.	SOAP	1.8	18.0
submitProduct	Allows the update and creation of Product Records. If the record already exists, it is updated; otherwise, a new record is created.	SOAP	1.8	18.0
Retrieve Product Specifications				
getProductSpecification V1	Retrieves a paged list of Product Specification records filtered according to the parameters defined in the request.	SOAP	1.8	18.0
Project Activities				
Activity Management	Allows updates to Project Activities from an external service. Used in Artwork integration. This service has been superseded by the Artwork Rest Service.	SOAP	1.8	1.10
UserRestService				

Service Name and Functions	Description	Type	From Version	To Version
List of Values	Retrieves a list of retailer or supplier users in a paged list.	REST	1.9	NA
Retrieve Record by ID	Retrieves a single user's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	1.9	NA
Retrieve Record by Business Key	Retrieves a single user's details. There are no request parameters, but the URL contains the login id parameter that determines the record to retrieve.	REST	1.9	NA
Check Record Modification Timestamp	Retrieves the last modification time for a user. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	1.9	NA
Create Record	Creates a new retailer or supplier user.	REST	1.9	NA
Update Record	Updates an existing retailer or supplier user.	REST	1.9	NA
SupplierRestService				
List of Values	Retrieves a list of suppliers in a paged list.	REST	1.9	NA
Retrieve Record by ID	Retrieves a single supplier’s details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	1.9	NA
Retrieve Record by Business Key	Retrieves a single supplier's details. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	1.9	NA
Check Record Modification Timestamp	Retrieves the last modification time for a supplier. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	1.9	NA
Create Record	Creates a new supplier.	REST	1.9	NA
Update Record	Updates an existing supplier .	REST	1.9	NA
SiteRestService				
List of Values	Retrieves a list of sites in a paged list.	REST	1.9	NA
Retrieve Record by ID	Retrieves a single site’s details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	1.9	NA
Retrieve Record by Business Key	Retrieves a single site's details. There are no request parameters, but the URL contains the supplier and site code parameters that determine the record to retrieve.	REST	1.9	NA

Service Name and Functions	Description	Type	From Version	To Version
Check Record Modification Timestamp	Retrieves the last modification time for a site. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	1.9	NA
Create Record	Creates a new site.	REST	1.9	NA
Update Record	Updates an existing site.	REST	1.9	NA
ContactRestService				
List of Values	Retrieves a list of contacts in a paged list.	REST	15.0	NA
Retrieve Record by ID	Retrieves a single contact's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	15.0	NA
Retrieve Supplier Contact Record by Business Key	Retrieves a single supplier contact's details. There are no request parameters, but the URL contains the supplier code, name, and email address parameters that determine the record to retrieve.	REST	15.0	NA
Retrieve Site Contact Record by Business Key	Retrieves a single site contact's details. There are no request parameters, but the URL contains the supplier and site codes, name, and email address parameters that determine the record to retrieve.	REST	15.0	NA
Check Record Modification Timestamp	Retrieves the last modification time for a contact. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	15.0	NA
Create Record	Creates a new contact.	REST	15.0	NA
Update Record	Updates an existing contact.	REST	15.0	NA
ProductRecordRestService				
List of Values	Retrieves a list of Product Records in a paged list.	REST	1.9	NA
Retrieve Record by ID	Retrieves a single Product Record's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	1.9	NA
Retrieve Record by Business Key	Retrieves a single Product Record's details. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	1.9	NA
Check Record Modification Timestamp	Retrieves the last modification time for a Product Record. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	1.9	NA
Create Record	Creates a new Product Record.	REST	1.9	NA

Service Name and Functions	Description	Type	From Version	To Version
Update Record	Updates an existing Product Record.	REST	1.9	NA
ProductSpecificationRestService				
List of Values	Retrieves a list of Product Specifications in a paged list.	REST	1.9	NA
Retrieve Record by ID	Retrieves a single Product Specification's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	1.9	NA
Retrieve Record by Business Key	Retrieves a single Product Specification's details. There are no request parameters, but the URL contains the specification number and version parameters that determine the record to retrieve.	REST	1.9	NA
Retrieve List with Advanced Filtering	Retrieves a list of Product Specifications in a paged list using advanced filtering. Parameters are passed to filter the returned list by values such as specification status, specification type, country where sold, and so on.	REST	1.9	NA
Check Record Modification Timestamp	Retrieves the last modification time for a Product Specification. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	1.9	NA
Create Record	Creates a new Product Specification.	REST	16.0	NA
Update Record	Updates an existing Product Specification.	REST	16.0	NA
BusinessCategoryService				
List of Values	Retrieves a list of Business Categories in a paged list.	REST	16.0	NA
Retrieve Record	Retrieves a single Business Category (a path of all levels). There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	16.0	NA
Check Record Modification Timestamp	Retrieves the last modification time for a Business Category. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	16.0	NA
Create Record	Creates a new Business Category.	REST	16.0	NA
Update Record	Updates an existing Business Category.	REST	16.0	NA
Delete Record	Deletes an existing Business Category.	REST	16.0	NA
AuditRestService				
List of Values (Audit)	Retrieves a list of Audits or Visits in a paged list.	REST	16.0	NA

Service Name and Functions	Description	Type	From Version	To Version
Retrieve Record by ID (Audit)	Retrieves a single Audit or Visit (and its Non Conformances/Issues). There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	16.0	NA
Retrieve Record by Business Key (Audit)	Retrieves a single Audit or Visit (and its Non Conformances/Issues). There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	16.0	NA
Check Record Modification Timestamp (Audit)	Retrieves the last modification time for an Audit or Visit. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	16.0	NA
Create Record (Audit)	Creates a new Audit or Visit.	REST	16.0	NA
Update Record (Audit)	Updates an existing Audit or Visit.	REST	16.0	NA
List of Values (Checklist)	Retrieves a list of Checklists in a paged list.	REST	16.0	NA
Retrieve Record by ID (Checklist)	Retrieves a single Checklist. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	16.0	NA
Retrieve Record by Business Key (Checklist)	Retrieves a single Checklist. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	16.0	NA
Create Record (Checklist)	Creates a new Checklist.	REST	16.0	NA
Update Record (Checklist)	Updates an existing Checklist.	REST	16.0	NA
Delete Record (Checklist)	Deletes an existing Checklist.	REST	16.0	NA
TaskRestService				
List of Values	Retrieves a list of current tasks for the user in the given language.	REST	1.9	NA
UrgentItemsRestService				
Number of Urgent Items	Retrieves the number of urgent items pending for a given user.	REST	1.9	NA
ArtworkRestService				
Started Activities	Retrieves a list of Project Activities that are flagged as Artwork activities, and their status has changed to Started within the specified date range.	REST	1.10	NA
Update Record	Updates the sub-status of existing Artwork Project Activities.	REST	1.10	NA
SSO	Single sign-on integration between Brand Compliance and the external Artwork system.	REST	1.10	NA

Services Grouped by Area

The following sections summarize the service features available by area of Brand Compliance functionality.

User and Supplier APIs

The following service features are applicable to the functionality available for user and supplier APIs:

Allows Supplier, Site, Contact, and User records to be created, updated, and extracted.
Does not allow deletion of records.
Does not import or export file attachments.
Does not include Scorecards, Audits & Visits, or Alerts.
A system log of service transactions is kept (not reportable).

Product APIs

The following service features are applicable to the functionality available for product APIs:

Allows Product Records to be created, updated, and extracted.
Barcodes transmitted by the data feed are stored on the Product Record and automatically transferred to the Product Specification when linked (using normal specification editor procedures).
A Product Record must be activated before it can be linked to a Product Specification.
Does not import file attachments.
Not intended for bulk upload of Product Records.
Does not include Pack Copy files or Surveillance test results.
If the full content of the Product Record is not included in the data feed, subsequent manual data entry may be required.
Does not apply to Produce Product Record types as they are created through a different process in Brand Compliance.
The SOAP version only applies to the single variant mode of operating the Product module, that is, does not handle multiple variants on the same Product Record.
A system log of service transactions is kept (not reportable).

Specification Outbound API

The following service features are applicable to the functionality available for the specification outbound API:

Transmit Product Specification data in XML form for Product Specifications that meet the filters used in the service call from the external system.
All specification types, including Produce and Counter Tickets, are supported.
Does not export file attachments.
Does not export Deleted Product Specifications.

A system log of service transactions is kept (not reportable).

Specification Inbound API

The following service features are applicable to the functionality available for the specification inbound API:

Allows Product Specification records to be created, updated, and extracted.
All specification types, including Produce and Counter Tickets, are supported.
Does not allow deletion of records.
Does not import or export file attachments.
Specifications can be created at any status; validation is applied when the user manually edits and progresses the specification through the Brand Compliance application UI.
A system log of service transactions is kept (not reportable).

Audits API

The following service features are applicable to the functionality available for the audits API:

Allows Audit and Visit records to be created, updated, and extracted.
Includes the associated Issues/Non Conformances and Checklists.
Does not allow deletion of records.
Does not import or export file attachments.
A system log of service transactions is kept (not reportable).

Artwork Project Activities

The following service features are applicable to the functionality available for the Artwork Project Activities API:

Enables external Artwork systems to update the sub-status of specific Project Activities that relate to Artwork. The external system is not enabled to complete the activity in Brand Compliance; this must be done manually.
Activity templates in the Project module of Brand Compliance have an option to integrate that activity with an Artwork system.
Project Activities in Brand Compliance, which have been set to integrate with Artwork and which have a status of Started, are available to the external system. Pack Copy data is not sent using an API, it is automatically sent in email to the designer.
The start and end dates of the Activity are not sent to the Artwork system. The Artwork process is usually well defined and it is usual for a project to be set up in the Artwork system at this point, using the Artwork system's own templates and based on the date that the message is received.

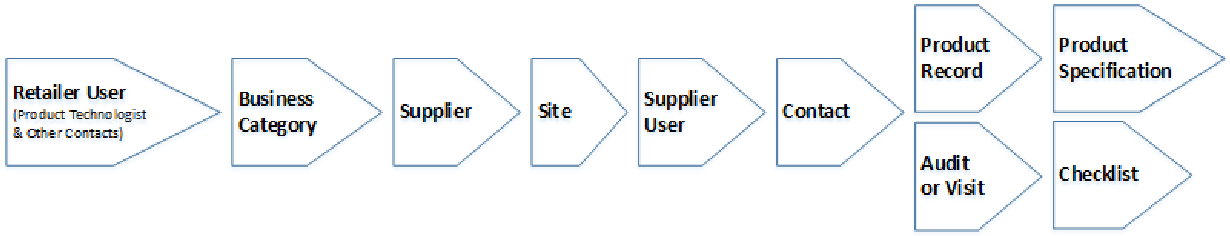
The Artwork API is secured and initiated per other Brand Compliance APIs. The following table summarizes the features introduced by each Brand Compliance release:

Artwork Integration Features by Release

Release	Feature
1.0	Activity Management SOAP API available to send a message when an activity status is set to Started. Message can be returned to set the Activity's sub-status accordingly in Brand Compliance.
1.5/1.6	Single Sign-on (SSO) Phase I: ▪ The Artwork system's URI is stored in the Brand Compliance Admin Area. An Artwork option is added to the Brand Compliance Navigator bar. It is enabled if the user has the Artwork User authority profile. ▪ Users must be created manually in both systems. SSO is achieved using a SOAP call.
1.8	Single Sign-on (SSO) Phase II: ▪ Users are managed automatically between Brand Compliance and the Artwork system through a REST API to pass user and role information; so no need to create manually in both systems. ▪ It is not necessary to log into both applications. The user must first log in to Brand Compliance, and when the user clicks the Artwork link, a custom SSO takes places. This involves the exchange of a secure token, and a callback to Brand Compliance from the external system to retrieve the user's information.
1.10	The Activity Management SOAP API is deprecated. It is superseded by the equivalent Artwork Rest Service.
15.0	Security Improvements: ▪ Messages are no longer automatically sent out from Brand Compliance, but now rely on a call from the Artwork external system to check if an activity has started. Also relies on a call to Brand Compliance to check for new users.

Dependencies

Following is the typical sequence of records to be created in order to create all of the above services:



The following table indicates the dependencies of each type of record that can be created using the APIs, that is, what configuration or other records must be in place for the creation to succeed:

Record Creation Dependencies

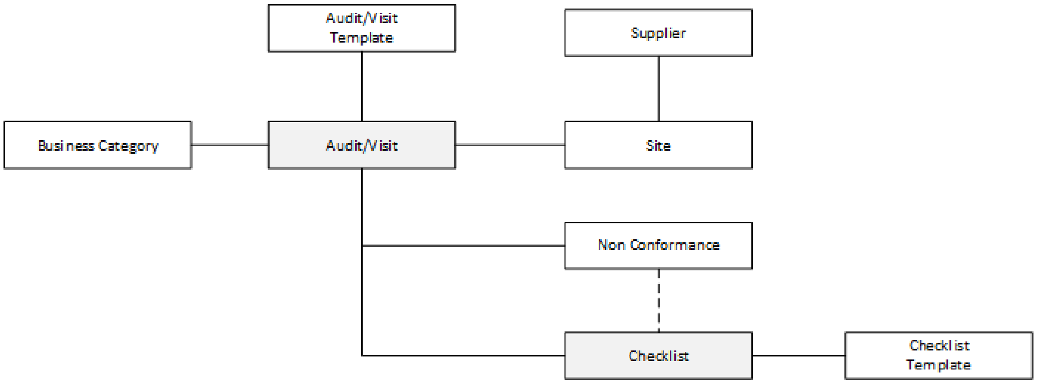
Record Type	Configuration	Other Record Types	Comments
User	**Company** (mandatory) **Languages** (mandatory) **User Roles** (mandatory) Authority Profiles Time Zones Users (Managers and Delegates) Areas	Supplier (if a supplier user)	When creating a user, it is associated to either the retailer or a supplier. If it is a supplier user, the ID of the Supplier record must be provided.
Supplier	**Billing Codes** (mandatory) **Business Units** (mandatory) **Supplier Types** (mandatory) Countries Supplier Top Grades Currencies Invoicing Systems Supplier Statuses Areas	NA	When creating a Supplier, omit the Site and Supplier Contact elements, and omit the Billing Details ID. Create Sites and Contacts after creating the Supplier.
Site	**Product Technologists** (mandatory) **Business Categories** (mandatory) **Site Types** (mandatory) Business Units Currencies Invoicing Systems Countries Audit Reasons Reference Types Reference Statuses Site Top Grades Site Statuses Risk Levels Other Production Details	Supplier User (Product Technologist)	When creating a Site, the ID of the Supplier record must be provided. Omit the Scorecards and Audit elements, and omit the Billing Details ID and Reference ID. If assigning a Business Category, the ID of the Business Category record must be provided; if assigning a Product Technologist, the IDs of the User record must be provided. Create Contacts after creating the Site.

Record Type	Configuration	Other Record Types	Comments
Contact	**Contact Roles** (mandatory)	User Supplier Site (if a Site contact)	When creating a Contact, the IDs of the associated User and Person records must be provided. The Person record holds an individual's personal contact details; the Contact record holds the individual's contact details in relation to that role (which may be different per role); the User record represents the individual's user account. A contact is associated to a supplier and/or a site; the IDs of the Supplier and Site records must be provided as appropriate. Omit the Site element if not a site-level contact.
Product	**RecordSpecification Types** (mandatory) **Product Record Statuses** (mandatory) **Product Technologist** (mandatory) **Other Contacts** (mandatory) Test Disciplines Specification Formats Business Categories	Supplier (if Active) Site (if linked to a site) User	A Product Record may be linked to a supplier/site; the IDs of the Supplier and Site records must be provided as appropriate. Omit the Discipline Detail, Product Covered, Barcode and Shipping Case Code IDs when creating. If assigning a Business Category, the ID of the Business Category record must be provided; if assigning a Product Technologist or Other Contact, the IDs of the User record must be provided. Product Technologist and Other Contacts are mandatory, however will be defaulted to TBC on creation if not provided, however, the dummy TBC User record must already exist. The status must be Active in order to be linked to a specification; for the status to be Active the Product Record must be linked to a supplier.
Product Specification	Specification Glossaries (some are mandatory)	Supplier (if linked to PR) Site (if linked to PR) Product Record (if linked to PR Contact (if linked to PR) User (if linked to PR) Business Category (if linked to PR)	A Specification may be linked to a Product Record by providing the ID of a Product Record which it is at Active status. If linking to a Product Record, the Supplier, Site, and Product Record (and associated Users, Contacts and Business Categories) must be created first. Some links to Specification Glossaries are mandatory.
Business Category	Business Categories (parent levels) Specification Types	NA	When creating a Business Category, if the category is a lower-level category, the ID of its parent Business Category record must be provided.

Record Type	Configuration	Other Record Types	Comments
Audit/Visit	**Audit/Visit Template** (mandatory) **Audit/Visit Status** (mandatory) **Product Technologist** (mandatory) **People Present** (mandatory) Supplier Contacts Business Categories Non Conformance: Assigned to Completed by Approved by	Supplier Site User	An Audit or Visit is linked to a supplier/site; the IDs of the Supplier and Site records must be provided. If assigning a Business Category, the ID of the Business Category record must be provided; if assigning a Product Technologist or People Present, the IDs of the User record must be provided. Non Conformances may include links to User records for the users who the issue is assigned to, was completed by, or is approved by. If the person is not a user of the application, a corresponding text field can be used to record the name. Also, see notes below.
Checklists	**Audit/Visit** (mandatory) **Checklist Question** (mandatory)	Audit/Visit Checklist Template	A Checklist is linked to an Audit/Visit, the IDs of the Audit or Visit record must be provided. The Checklist's questionnaire questions and available answers are derived from the Checklist Templates assigned to the Audit/Visit, the IDs of the Template records must be provided.

Notes on the Creation and Maintenance of Audits/Visits and Checklists

- The following diagram shows the main links between Audits/Visits, their Checklists, and other entities:

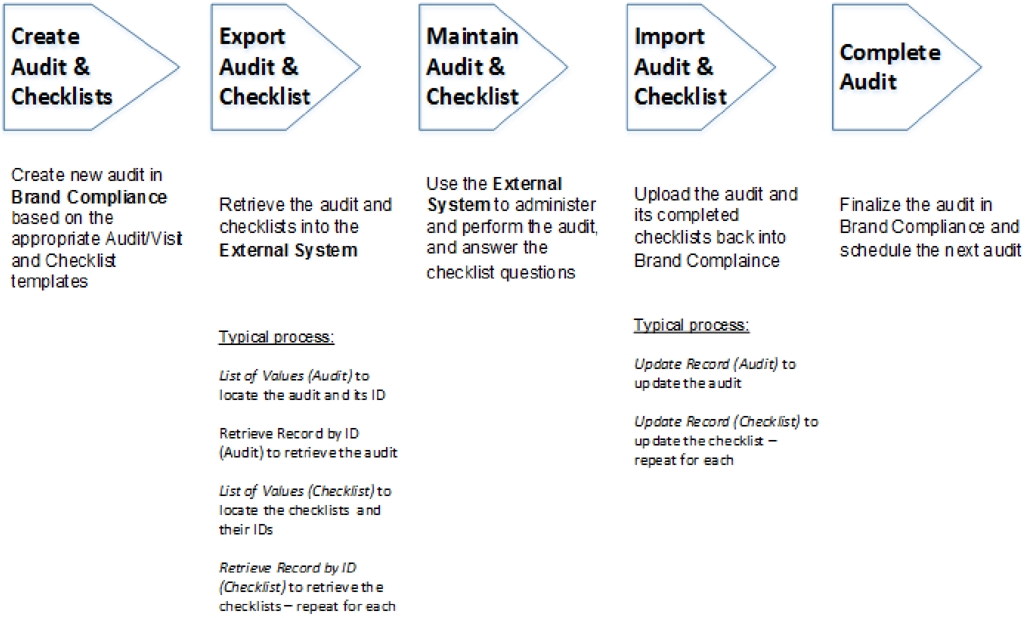


- For the creation of an Audit/Visit and its Checklists, in addition to providing the links to the associated Audit/Visit Template and Checklist Template records, certain values from the templates must also be included in the payload of the Audit/Visit Create and Update functions.

- It is recommended that the creation of Audits/Visits and Checklist be managed within the Brand Compliance application. Once created, records can be taken into the external system where the audit is performed, prior to importing back into Brand Compliance.

This approach ensures the necessary template fields will have been obtained with the initial Retrieve Record function.

The following diagram shows the stages of a typical scenario:


Create Audit & Checklists
Export Audit & Checklist
Maintain Audit & Checklist
Import Audit & Checklist
Complete Audit
Create new audit in Brand Compliance based on the appropriate Audit/Visit and Checklist templates
Retrieve the audit and checklists into the External System
Use the External System to administer and perform the audit, and answer the checklist questions
Upload the audit and its completed checklists back into Brand Complaince
Finalize the audit in Brand Compliance and schedule the next audit
Typical process:
List of Values (Audit) to locate the audit and its ID
Retrieve Record by ID (Audit) to retrieve the audit
List of Values (Checklist) to locate the checklists and their IDs
Retrieve Record by ID (Checklist) to retrieve the checklists – repeat for each
Typical process:
Update Record (Audit) to update the audit
Update Record (Checklist) to update the checklist – repeat for each

- If it is necessary to create audits or visits from scratch in the external system, the recommended approach would be to create a set of *gold standard* audits/visits in Brand Compliance, which are assigned to a dummy supplier and site.

  To create a new audit or visit, the process would be to retrieve the appropriate *gold standard* audit/visit through the GET function, amend the retuned XML data to reflect the audit/visit to be created, and then submit the creation of the new audit/visit (plus any non conformances and checklists) using the POST function.

- If a checklist audit is to be created using the Create Audit service, the Audit/Visit record must first be created using the Create Audit function, followed by the creation of each of its Checklist records.

  Creating an audit through the API which is linked to a checklist template does not automatically create the checklists. They must be individually created using the Create Checklist function.

- An audit/visit's non conformances are included as separate elements within the audit/visit data. An audit/visit's non conformances are allocated when creating or updating the audit/visit.

Note: The API does not automatically create non conformances based on the answers of checklist questions.

Linking to Related Records

Links to associated records are included in the XML as an id element, which holds the record ID of the related record. In some cases, a code element also holds the related record's business key.

For example, the XML returned from the SiteRestService API will contain a supplier element, which itself contains an id element and a code element. The value in the id element can be used by the SupplierRestService Retrieve Record by ID function to retrieve the full details of the Supplier record. Alternatively, the value in the code element can be used by the SupplierRestService Retrieve Record by Business Key function to retrieve the Supplier record.

The id element is also used in the XML to hold the record ID of a value that is held in a glossary such as a recipe ingredient or a country.

The following table shows where an API provides links to other records, and the id or code elements used to locate those records. The list of notes below the table indicates the service to use to retrieve the related record.

Related Record Links

API	Links to Records
Supplier	Contact: contactDetail/id (Contact record ID link). See note 7. Contact (User): person/id (User record ID link). See note 1. Site: site/id (Site record ID link). See note 3. Site: site/code site (Site business key link). See note 4.
Site	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: supplier/code (Supplier business key link). See note 6. Business Categories: businessCategories/id (Business Category record ID link). See note 10. Lead Business Category: leadBusinessCategory/id (Business Category record ID link). See note 10. Lead Product Technologist (User): leadTechnicalManager/id (User record ID link). See note 1. Lead Product Technologist (User):leadTechnicalManager/code (User business key link). See note 2. Other Technologists (User): otherTMs/id (User record ID link). See note 1. Other Technologists (User): otherTMs/code (User business key link). See note 2. Contact: siteContact/id (Contact record ID link). See note 7.
Contact	Contact (User): person/id (User record ID link). See note 1. Supplier: company/id (Supplier record ID link). See note 5. Supplier: Company/code (Supplier business key link). See note 6. Site: site/id (Site record ID link). See note 3. Site: Site/code (Site business key link). See note 4.
User	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: Supplier/code (Supplier business key link). See note 6.

API	Links to Records
Audit/Visit	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: supplier/code (Supplier business key link). See note 6. Site: site/id (Site record ID link). See note 3. Site: Site/code (Site business key link). See note 4. People Present (User): user/id (User record ID link). See note 1. People Present (User): user/code (User business key link). See note 2. Lead Product Technologist (User): leadTechnicalManager/id (User record ID link). See note 1. Lead Product Technologist (User): leadTechnicalManager/code (User business key link). See note 2. Business Categories: businessCategories/id (Business Category record ID link). See note 9. Lead Business Category: leadBusinessCategory/id (Business Category record ID link). See note 9. Non Conformance Assigned To (User): assignedTo/id (User record ID link). See note 1. Non Conformance Completed By (User): completedBy/id (User record ID link). See note 1. Non Conformance Approved By (User): approvedBy/id (User record ID link). See note 1.
Checklist	Audit/Visit: audit/id (Audit/Visit record ID link). See note 11.
Product Record	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: supplier/code (Supplier business key link). See note 6. Site: site/id (Site record ID link). See note 3. Site: site/code (Site business key link). See note 4. Business Categories: businessCategory/id (Business Category record ID link). See note 10. Technologist (User): technologist/id (User record ID link). See note 1. Technologist (User): technologist/code (User business key link). See note 2. Other Contacts (User): user/id record ID link). See note 1. Other Contacts (User): user/code (business key link). See note 2.
Produce Product Record	As per Product Record, except Business Categories, which are derived from the Product Specification.
Product Specification	Supplier: site/supplierId (Supplier record ID link). See note 5. Supplier: site/supplierCode (Supplier business key link). See note 6. Site: primarySites/site/id (Site record ID link). See note 3. Site: primarySites/site/code (Site business key link). See note 4. Product Record: productRecord/id (Product Record record ID link). See note 8. Product Record: productRecord/code (Product Record business key link). See note 9. Supplier Contact: contactDetails/id (Contact record ID link). See note 7. Supplier Contact (User): person/id (User record ID link). See note 1. For further details on locating a Specification's Product Record, see the Main Details page of the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*.

Notes:

1. User record ID link: UserRestService - Retrieve Record by ID
2. User business key link: UserRestService - Retrieve Record by Business Key
3. Site record ID link: SiteRestService - Retrieve Record by ID

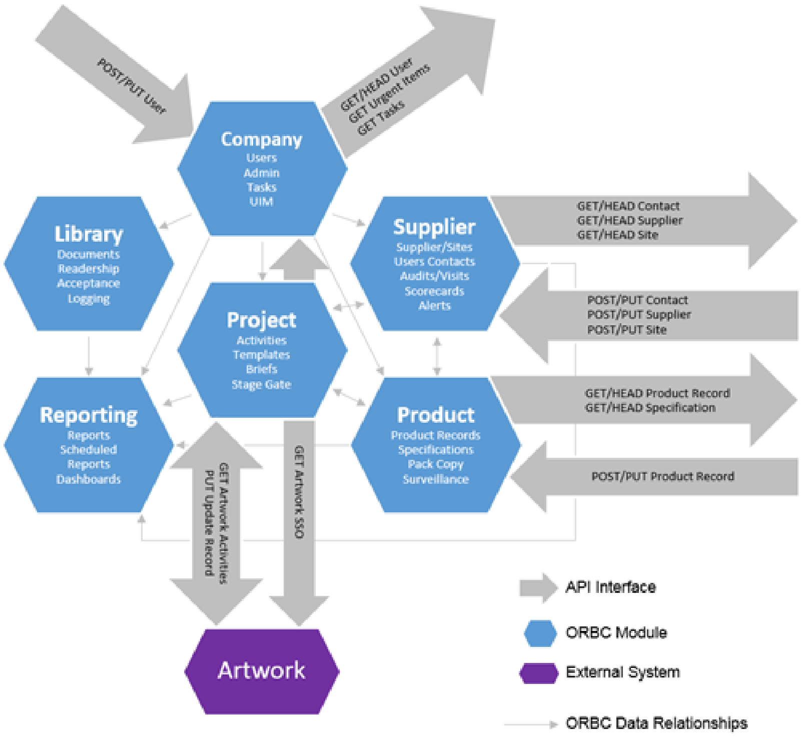
4. Site business key link: SiteRestService - Retrieve Record by Business Key
5. Supplier record ID link: SupplierRestService - Retrieve Record by ID
6. Supplier business key link: SupplierRestService - Retrieve Record by Business Key
7. Contact record ID link: ContactRestService - Retrieve Record by ID
8. Product Record record ID link: ProductRestService - Retrieve Record by ID
9. Product Record business key link: ProductRestService - Retrieve Record by Business Key
10. Business Category record ID link: BusinessCategoryService - Retrieve Record
11. Audit/Visit Record ID link: AuditRestService - Retrieve Record by ID (Audit)

4
RESTful APIs

The following RESTful APIs are available:

- UserRestService: Users
- SupplierRestService: Suppliers
- SiteRestService: Sites
- ContactRestService: Supplier/Site Contacts
- ProductRecordRestService: Product Records
- ProductSpecificationRestService: Product Specifications
- TaskRestService: Users' Tasks
- UrgentItemsRestService: Users' Urgent Items
- ArtworkRestService: Artwork Activities
- BusinessCategoryService: Business Categories
- AuditRestService: Audits and Visits

Figure 4–1 Overview of RESTful APIs



Parameters and Filtering

Various parameters can be included in calls to the APIs, generally to define what data is to be returned, but can also control how records are returned. The following table lists some commonly used parameters.

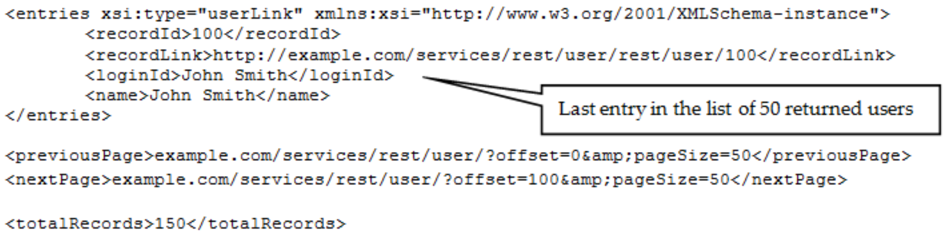
Common Parameters

Parameter	Type	Description
offset	int	Used with pageSize to control the paging of a returned list of records. Specifies the starting point for the retrieval of records. If not specified, zero is assumed. For example, to retrieve 150 records: **1.** A call with offset = 0 and pageSize = 50 returns the first 50 records. **2.** Then, a call with offset = 50 and pageSize = 50 returns the next 50. **3.** Then, a call with offset = 100 and pageSize = 50 returns the final 50.
pageSize batchSize	int	Used with offset to control the paging of a returned list of records. Specifies the number of entries in each page of returned list of records. If not specified, 30 is assumed. The maximum is 100.
isActive	Boolean	Used to only return records that are flagged as being active (TRUE).
modifiedSince	string	Used to locate records that have been created or updated since a specific date/time.

Parameter	Type	Description
modifiedUntil	string	Used to locate records that have been updated or created until a specific date/time.

When a list of records is returned, it may include a previousPage and nextPage element. These elements provide URI links to the previous or next pages of records respectively. A totalRecords element is included that shows the total number of records available for retrieval.

For example, using the List of Values function to retrieve a list of User records, where there are 150 records to be retrieved, with the pageSize parameter set to 50 returns the first 50 records within entity elements, along with a nextPage element containing a URI link to the next 50 records which can be used to retrieve the next 50 records. When the second page of 50 records are returned, the XML includes a nextPage URI link to the final 50 records, along with a previousPage URI link to the first 50. The following graphic illustrates this example.



```
<entries xsi:type="userLink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <recordId>100</recordId>
        <recordLink>http://example.com/services/rest/user/rest/user/100</recordLink>
        <loginId>John Smith</loginId>
        <name>John Smith</name>
</entries>

<previousPage>example.com/services/rest/user/?offset=0&amp;pageSize=50</previousPage>
<nextPage>example.com/services/rest/user/?offset=100&amp;pageSize=50</nextPage>

<totalRecords>150</totalRecords>
```

The totalRecords element shows the total number of retrievable records. The previousPage and nextPage elements are only present if there are previous and/or next pages to be retrieved.

Note: Date/Time parameters must be provided in the YYYY-MM-DD hh:mm:ss format.

When retrieving records, the returned XML only includes elements that actually contain data; empty elements are omitted.

The details of the returned records are contained in an entities element, and repeated in an entryArray element. The entryArray element is a representation from the Java objects and can be ignored.

Error Messages

The Brand Compliance APIs use standard web services messaging protocols to notify the success or failure of a call to the service. Where an API handles a specific error condition, details of the returned message can be found in the following sections. The following table lists various generic error messages that may be returned by calls to any of the APIs.

Generic Error Messages

Element	Message	Meaning
userId	HTTP Status 401 - Bad credentials	User ID or password is invalid
Password	HTTP Status 401 - Bad credentials	User ID or password is invalid

Element	Message	Meaning
Disabled	HTTP Status 401 - User is disabled	External System account is not enabled in Brand Compliance
offset	IllegalStateException: Offset must be a positive integer	Invalid offset value - must be between zero and 2,147,483,647
offset	HTTP 404 Not Found	Not numeric or an integer
pageSize	IllegalStateException: The Page Size must be between 1 and 100	Invalid page size value
	HTTP 404 Not Found	Record cannot be located due to an invalid key or ID
	HTTP 417 Expectation Failed	Record cannot be located due to an invalid ID
	HTTP Status 500 - Internal Server Error	Unspecified internal error occurred
	HTTP Status 500 - Internal Server Error	Non-numeric value in numeric parameters

Wildcard Searches

The % character can be used as a wildcard filter when locating the records.

Using the % at the start of the string will search for matches where the given string ends with the text being searched for, for example, %smith could be used to search for all users with a surname of smith.

Using the % at the end of the string will look for matches where the given string begins with the text being searched for, for example, mark% could be used to search for all users with the name mark.

The following table shows which APIs, functions, and parameters support the use of wildcards.

Wildcard Searches

API	Function	Parameter
User	List of Values	Name jobTitle country userRoles authorityProfiles
Supplier	List of Values	supplierCode supplierName supplierType Country leadBusinessUnit

API	Function	Parameter
Site	List of Values	siteCode siteName supplierName country businessCategory businessUnit supplierName
Contact	List of Values	Name supplierContactRoles siteContactRoles country selectedSites supplierCode supplierName
Product Specification	Retrieve List with Advanced Filtering	specStatus productNumber language sectionType countryWhereSold ProductCoverage Supplier site

Notes:

- Wildcard searches are not case sensitive so, for example, searching a name field for John Smith or john smith will return the same matches.
- When searching glossary parameters, be aware that they relate to the code (not the description) of the glossary entry.

UserRestService

This section describes the API for managing retailer and supplier users. The following functions are available:

- List of Values: retrieves a list of users
- Retrieve Record by ID: retrieves a User record using its unique identifier
- Retrieve Record by Business Key: retrieves a User record using its business key
- Check Record Modification Timestamp: retrieves the timestamp when a User record was last updated
- Create Record: creates a new User record
- Update Record: updates an existing User record

List of Values

Description

Retrieves a list of users in a paged list. Use this function to retrieve a simple list of user names and IDs, or to locate User record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/user
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
name	Optional	string	No	John Smith
jobTitle	Optional	string	No	Project Manager
language	Optional	string	No	en_GB
country	Optional	string	No	UK
userRoles	Optional	string	No	PROJECT ADMINISTRATOR
userType	Optional	string	Yes	RETAILER
authorityProfiles	Optional	string	No	PROJECT MANAGER
siteStatus	Optional	string	Yes	ACTIVE~INACTIVE
supplierActive	Optional	string	No	FALSE
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39

Example URLs

.../services/rest/user/?offset=2&pageSize=20
.../services/rest/user/?siteStatus=Active
.../services/rest/user/?name=Frank%

Response Details

For a successful response, XML is returned with a UserLinkList root element containing an entries element for each matched user. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	User record's internal ID
recordLink	string	URI to the UserRestService Retrieve record service for this user
loginId	string	Login ID / business key to the User record
name	string	User's name

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID

Description

Retrieves a single User record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual user.

Endpoint address: /services/rest/user/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/user/105

Response Details

For a successful response, XML is returned with a userFullDTO root element containing the individual attributes of the requested User record. If an ID is not specified, a list of all users is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the User to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single User record's details using its business key (login ID). Use this function to retrieve the full details of an individual user using its Brand Compliance login ID.

Endpoint address: /services/rest/byKey/{loginId}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {loginId} parameter that determines the record to retrieve.

Example URL

…/services/rest/user/byKey/Frank Jones

Response Details

For a successful response, XML is returned with a userFullDTO root element containing the individual attributes of the requested User record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
loginId	HTTP 404 Not found	Invalid {loginId} - blank or not found
loginId	HTTP 404 Not found	Invalid {loginId} - not found

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a User record. Use this function to determine when a user's details were last updated.

Endpoint address: /services/rest/user/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/user/105

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested User record. For example:

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not found	Invalid {id} - not numeric

Create Record

Description

Creates a new User record. Use this function to create a new user in Brand Compliance based on data sourced from the external system.

Dependencies: If creating a supplier user, the Supplier must be present in the application and its record ID obtained. For more information, see "Dependencies".

Endpoint address: /services/rest/user
HTTP method: POST

Request Details

The body of the request contains a UserFullDTO to specify the details of the user to create. Compared to retrieving a user (which uses the same UserFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created User record need to be included. As a minimum, this must include the fields shown in the following table:

Supplier User Mandatory Fields

Field Name	Element Name
Name	person / name
Email	person / email
Login Id	code
User Role	role / code
Supplier Code	person / supplier / code supplier / code
User Type	userType (Fixed value SUPPLIER)

Example Request XML

This example shows the minimum requirement to be able to create a supplier user.

```
<ns0:userFullDTO
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
       <ns0:code>jphillips</ns0:code>
       <ns0:person>
              <ns0:email>jphillips@supplier.co.uk</ns0:email>
              <ns0:name>Jane Phillips</ns0:name>
              <ns0:supplier>
                     <ns1:code>A0174</ns1:code>
              </ns0:supplier>
       </ns0:person>
       <ns0:role>
              <ns1:code>SUPPLIER ADMINISTRATOR</ns1:code>
       </ns0:role>
       <ns0:supplier>
              <ns1:code>A0174</ns1:code>
       </ns0:supplier>
       <ns0:userType>SUPPLIER</ns0:userType>
</ns0:userFullDTO>
```

Note: When creating a supplier user, the user type must be set to SUPPLIER. The supplier code must be set to match the required supplier from Brand Compliance.

Site User Mandatory Fields

Field Name	Element Name
Name	person / name
Email	person / email
Login Id	code
User Role	role / code
Supplier Code	person / supplier / code supplier / code
Sites	sites / code sites/ supplierCode
User Type	userType (Fixed value SITE)

Example Request XML

This example shows the minimum requirement to be able to create a site user.

```
<ns0:userFullDTO
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
       <ns0:code>jphillips</ns0:code>
       <ns0:person>
              <ns0:email>sphillips@supplier.co.uk</ns0:email>
              <ns0:name>Sean Phillips</ns0:name>
              <ns0:supplier>
                     <ns1:code>WS0001</ns1:code>
                     <ns1:id>22</ns1:id>
              </ns0:supplier>
       </ns0:person>
       <ns0:role>
              <ns1:code>SUPPLIER ADMINISTRATOR</ns1:code>
       </ns0:role>
       <ns0:sites>
              <ns1:code>WS0001-0001</ns1:code>
              <ns1:id>26</ns1:id>
              <ns1:supplierCode>WS0001</ns1:supplierCode>
              <ns1:id>22</ns1:id>
       </ns0:sites>
       <ns0:supplier>
              <ns1:code>WS0001</ns1:code>
              <ns1:id>22</ns1:id>
       </ns0:supplier>
       <ns0:userType>SITE</ns0:userType>
</ns0:userFullDTO>
```

Note: When creating a site user the user type must be set to SITE. The supplier code must be set to match the required supplier from Brand Compliance. The site codes must be set to match the required site from Brand Compliance.

Retailer User Mandatory Fields

Field Name	Element Name
Name	person / name
Email	person / email
Login Id	code
User Role	role / code
Supplier Code	person / supplier / code supplier / code (Fixed value RETAILER)
User Type	userType (Fixed value RETAILER)

Example Request XML

This example shows the minimum requirement to be able to create a retailer/portal owner user.

```
<ns0:userFullDTO
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
       <ns0:code>james howard</ns0:code>
       <ns0:person>
              <ns0:email>james.howard@oracle.com</ns0:email>
              <ns0:name>James Howard</ns0:name>
              <ns0:supplier>
                     <ns1:code>RETAILER</ns1:code>
              </ns0:supplier>
       </ns0:person>
       <ns0:role>
              <ns1:code>POWER USER</ns1:code>
       </ns0:role>
       <ns0:supplier>
              <ns1:code>RETAILER</ns1:code>
       </ns0:supplier>
       <ns0:userType>RETAILER</ns0:userType>
</ns0:userFullDTO>
```

Note: When creating a retailer/portal owner user, the user type must be set to RETAILER. The supplier code must be set to RETAILER.

Where the record is linked to another record, such as the Role in these cases, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

For a successful response, an HTTP 200 response is sent with a body containing a UserLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created User record's internal ID
recordLink	string	URI to the newly created User record, for use in a GET request
loginId	string	Login ID / business key to the newly created User record
name	string	User's name

Error Responses

If the supplied data does not result in a valid User (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
code	ERROR: class com.micros.creations.core.domain.User.code - The condition is invalid	Code not provided
email	ERROR: class com.micros.creations.core.domain.User.person.email - The condition is invalid	Email not provided
name	ERROR: class com.micros.creations.core.domain.User.person.name - The condition is invalid	Name not provided
role	ERROR: class com.micros.creations.core.domain.User.role - A user must have an active authority profile from either roles or additional authority profiles	Role or authority profiles not provided

Update Record

Description

Updates an existing User record. Use this function to update a user's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/{id}
HTTP method: PUT

Request Details

The body of the request contains a UserFullDTO to specify the updates to the User record. Compared to retrieving a user (which uses the same UserFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is similar to that for creating a user. After the call, the User record is updated to match the request.

> **Note:** When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the User record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a UserLink element. The UserLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	User record's internal ID
recordLink	string	URI to the User record, for use in a GET request
loginId	string	Login ID/business key to the User record
name	string	User's name

Error Responses

If the supplied data does not result in a valid user (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

SupplierRestService

This section describes the API for managing suppliers. The following functions are available:

- List of Values: retrieves a list of suppliers
- Retrieve Record by ID: retrieves a Supplier record using its unique identifier
- Retrieve Record by Business Key: retrieves a Supplier record using its business key
- Check Record Modification Timestamp: retrieves the timestamp when a Supplier record was last updated

- Create Record: creates a new Supplier record
- Update Record: updates an existing Supplier record

List of Values

Description

Retrieves a list of suppliers in a paged list. Use this function to retrieve a simple list of supplier names and IDs, or to locate Supplier record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/supplier
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
supplierStatus	Optional	string	Yes	AWAITING REGISTRATION
supplierCode	Optional	string	Yes	A0001
supplierName	Optional	string	Yes	West Road Site
supplierType	Optional	string	Yes	AGENT
country	Optional	string	Yes	UK
leadBusinessUnit	Optional	string	Yes	UK
isActive	Optional	string	No	TRUE (or blank)
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39

Example URLs

…/services/rest/supplier/?offset=2&pageSize=20
…/services/rest/supplier/?supplierStatus=AWAITING REGISTRATION
…/services/rest/supplier/?supplierName=API%

Response Details

For a successful response, XML is returned with a SupplierLinkList root element containing an entries element for each matched supplier. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Supplier record's internal ID
recordLink	string	URI to the SupplierRestService Retrieve record service for this supplier
code	string	Supplier code business key to the Supplier record
name	string	Supplier's name
localName	string	Supplier’s name in business language (if used)

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
supplierStatus	IllegalStateException: The Supplier Status <###> does not exist	Invalid supplier status
supplierStatus	IllegalStateException: The Supplier Statuses <###~###~###> do not exist	Invalid supplier status
supplierType	IllegalStateException: The Supplier Type <###> does not exist	Invalid supplier type
supplierType	IllegalStateException: The Supplier Types <###~###> do not exist	Invalid supplier type
country	IllegalStateException: The Country <###> does not exist	Invalid country
country	IllegalStateException: The Countries <###~###> do not exist	Invalid country
leadBusinessUnit	IllegalStateException: The Business Unit <###> does not exist	Invalid business unit
leadBusinessUnit	IllegalStateException: The Business Units <###~###> do not exist	Invalid business unit

Retrieve Record by ID

Description

Retrieves a single Supplier record’s details using the record’s internal ID (which is not visible in the UI). Use this function to retrieve the full details of an individual supplier.

Endpoint address: /services/rest/supplier/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/supplier/9

Response Details

For a successful response, XML is returned with a supplierFullDTO root element containing the individual attributes of the requested Supplier record. If an ID is not specified, a list of all users is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Supplier to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Supplier record's details using its business key (supplier code). Use this function to retrieve the full details of an individual supplier using the Brand Compliance supplier code.

Endpoint address: /services/rest/supplier/byKey/{code}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {code} parameter that determines the record to retrieve.

Example URL

…/services/rest/supplier/byKey/A0901

Response Details

For a successful response, XML is returned with a supplierFullDTO root element containing the individual attributes of the requested Supplier record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not Found	Invalid {code} - blank
code	HTTP 404 Not Found	Invalid {code}

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Supplier record. Use this function to determine when a supplier's details were last updated.

Endpoint address: /services/rest/supplier/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/supplier/9

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Supplier record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Responses

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 404 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Supplier record. Use this function to create new suppliers in Brand Compliance based on data sourced from the external system.

Dependencies: Create sites and contacts after creating the supplier. For more information, see "Dependencies".

Endpoint address: /services/rest/supplier
HTTP method: POST

Request Details

The body of the request contains a SupplierFullDTO to specify the detail of the supplier to create. Compared to retrieving a supplier (which uses the same SupplierFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created Supplier record need to be included. As a minimum, this must include the fields shown in the following table.

Supplier Mandatory Fields

Field Name	Element Name
Supplier Name	name
Contact Name	supplierContactName
Email	email
Supplier Type	supplierType/code
Lead Business Unit	businessUnit/code
Billing Code	billingCode/code
Status	Status
Supplier Code Confirmed Flag	supplierCodeConfirmed

Example Request XML

This example shows the minimum requirement to be able to create a supplier.

```
<ns0:supplierFullDTO
    xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
    xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">

       <ns0:billingCode>
              <ns1:code>SMALL</ns1:code>
       </ns0:billingCode>
       <ns0:businessUnit>
              <ns1:code>UK</ns1:code>
       </ns0:businessUnit>
       <ns0:email>supplier.contactemail@supplier.co.uk</ns0:email>
       <ns0:name>Name of Supplier</ns0:name>
       <ns0:status>AWAITING REGISTRATION</ns0:status>
       <ns0:supplierContactName>Supplier Contact Name</ns0:supplierContactName>
       <ns0:supplierCodeConfirmed>false</ns0:supplierCodeConfirmed>
       <ns0:supplierType>
              <ns1:code>SUPPLIER_TYPE</ns1:code>
       </ns0:supplierType>

</ns0:supplierFullDTO>
```

Set the isActive Boolean flag to "true" if the supplier registration process wizard is to be bypassed. This would typically be the case where the data being passed through the API on creating the account includes the main name and address and contact details. Not setting the flag, or setting it to "false," will cause the registration process wizard to be presented when the first supplier logs in.

Where the record is linked to another record, such as the Supplier Type in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a SupplierLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Supplier record's internal ID
recordLink	string	URI to the newly created Supplier record, for use in a GET request
code	string	Supplier code business key to the newly created Supplier record
name	string	Supplier's name
localName	string	Supplier’s name in business language (if used)

Error Messages

If the supplied data does not result in a valid Supplier (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
billingCode	ERROR: class <<hostname>>.Supplier.billingCode - The condition is invalid	No tag provided
billingCode	IllegalStateException: Cannot locate keyword for class <<hostname>>.BillingCode using code:null	No <ns1:code> tag
billingCode	IllegalStateException: Cannot locate keyword for class <<hostname>>..BillingCode using code:<###>	Invalid <ns1:code>
businessUnit	ERROR: class <<hostname>>..Supplier.businessUnit - The condition is invalid	No tag provided
businessUnit	IllegalStateException: Cannot locate keyword for class <<hostname>>..domain.BusinessUnit using code:null	No <ns1:code> tag
businessUnit	IllegalStateException: Cannot locate keyword for class <<hostname>>..BusinessUnit using code:<###>	Invalid <ns1:code>
code	ERROR: class com.micros.creations.core.domain.Supplier.code - ??Another object exists with this value??	Supplier code already exists
email	ERROR: class com.micros.creations.core.domain.Supplier.email - The condition is invalid	No tag provided
email	ERROR: class <<hostname>>.Supplier.email - The attribute is invalid	Malformed email address
supplierContactName	ERROR: class <<hostname>>.Supplier.supplierContactName - The condition is invalid	No tag provided

Element	Message	Meaning
supplierContactName	ERROR: class <<hostname>>.Supplier.supplierContactName - The condition is invalid	No value provided
name (supplier name)	ERROR: class <<hostname>>.Supplier.name - The condition is invalid	No tag provided
name (supplier name)	ERROR: class <<hostname>>.Supplier.name - The condition is invalid	No value provided
supplierType	ERROR: class <<hostname>>.Supplier.supplierType - The condition is invalid	No tag provided
supplierType	IllegalStateException: Cannot locate keyword for class <<hostname>>.SupplierType using code:null	No <ns1:code> tag
supplierType	IllegalStateException: Cannot locate keyword for class <<hostname>>.SupplierType using code:<###>	Invalid <ns1:code>

Update Record

Description

Updates an existing Supplier record. Use this function to update a supplier's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/supplier/{id}
HTTP method: PUT

Request Details

The body of the request contains a SupplierUpdateDTO to specify the updates to the Supplier record. Compared to retrieving a supplier (which uses the SupplierFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is similar to that for creating a supplier, but crucially, the links to other top-level records (Sites, SiteContact, SupplierContact) are omitted. The omission of those ensures that when updating a supplier, only the supplier details need to be specified, and not the details for the related records that may not require updating (and which should be updated to calls to their respective services). After the call, the Supplier record is updated to match the request.

> **Note:** When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Supplier record (except for the Sites, SiteContacts, and SupplierContacts).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a SupplierLink element. The SupplierLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Supplier record's internal ID
recordLink	string	URI to the Supplier record, for use in a GET request
code	string	Supplier code business key to the Supplier record
name	string	Supplier's name
localName	string	Supplier’s name in business language (if used)

Error Messages

If the supplied data does not result in a valid Supplier (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
billingCode	IllegalStateException: Cannot locate keyword for class <<hostname>>.BillingCode using code:<###>	Invalid <ns1:code>
businessUnit	IllegalStateException: Cannot locate keyword for class <<hostname>>domain.BusinessUnit using code:<###>	Invalid <ns1:code>
code	MySQLIntegrityConstraintViolationException: Duplicate entry <###> for key 'c_code'	Supplier code provided that already exists for another supplier
name	PropertyValueException: not-null property references a null or transient value: <<hostname>>.domain.Supplier.name	No tag provided
status	NullPointerException:	No status tag provided
status	IllegalArgumentException: Empty string is not allowed, should be null on method: <<hostname>>.domain.Supplier@1df35f3[id=<###>,code=<###>]	No status code provided

SiteRestService

This section describes the API for managing sites. The following functions are available:

- List of Values: retrieves a list of sites
- Retrieve Record by ID: retrieves a Site record using its unique identifier
- Retrieve Record by Business Key: retrieves a Site record using its business key

- Check Record Modification Timestamp: retrieves the timestamp when a Site record was last updated
- Create Record: creates a new Site record
- Update Record: updates an existing Site record

List of Values

Description

Retrieves a list of sites in a paged list. Use this function to retrieve a simple list of supplier names and IDs, or to locate Site record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/site
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
siteStatus	Optional	string	Yes	AWAITING REGISTRATION
siteCode	Optional	string	Yes	A0001-0001
siteName	Optional	string	Yes	West Road Site
siteType	Optional	string	Yes	CANNERY
country	Optional	string	Yes	UK
businessCategory	Optional	string	Yes	CATEGORY_3_1
businessUnit	Optional	string	Yes	UK
supplierName	Optional	string	Yes	ABC Ltd
supplierActive	Optional	Boolean	No	TRUE (or blank)
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39

Example URLs

…/services/rest/site/?offset=2&pageSize=20
…/services/rest/site/?siteStatus=ACTIVE
…/services/rest/site/?supplierName=API%

Response Details

For a successful response, XML is returned with a SiteLinkList root element containing an entries element for each matched site. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Site record's internal ID
recordLink	string	URI to the SiteRestService Retrieve record for this site
code	string	Site code business key to the Site record
name	string	Site's name
localName	string	Site’s name in business language (if used)

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
siteStatus	IllegalStateException: The Supplier Status <###> does not exist	Invalid site status
siteStatus	IllegalStateException: The Supplier Statuses <###~###~###> do not exist	Invalid site status
siteType	IllegalStateException: The Supplier Type <###> does not exist	Invalid supplier type
siteType	IllegalStateException: The Supplier Types <###~###> do not exist	Invalid supplier type
country	IllegalStateException: The Country <###> does not exist	Invalid country
country	IllegalStateException: The Countries <###~###> do not exist	Invalid country
businessCategory	IllegalStateException: The Business Category <###> does not exist	Invalid business category
businessCategory	IllegalStateException: The Business Categories <###~###> do not exist	Invalid business category
businessUnit	IllegalStateException: The Business Unit <###> does not exist	Invalid business unit
businessUnit	IllegalStateException: The Business Units <###~###> do not exist	Invalid business unit
modifiedSince	Invalid dates cause stack trace to come back to "response"	
modifiedUntil	Invalid dates cause stack trace to come back to "response"	

Retrieve Record by ID

Description

Retrieves a Site record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual site.

Endpoint address: /services/rest/site/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/site/87

Response Details

For a successful response, XML is returned with a siteFullDTO root element containing the individual attributes of the requested Site record. If an ID is not specified, a list of all sites is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Site to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Site record's details using its business key (supplier and codes). Use this function to retrieve the full details of an individual site using the combination of the Brand Compliance supplier and site codes.

Endpoint address: /services/rest/site/byKey/{supplierCode}/{siteCode}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {supplierCode} and {siteCode} parameters that determine the record to retrieve.

Example URL

…/services/rest/site/byKey/A0001/A0001-0001

Response Details

For a successful response, XML is returned with a siteFullDTO root element containing the individual attributes of the requested Site record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
supplierCode	HTTP 404 Not Found	Invalid {supplierCode} - blank
siteCode	HTTP 404 Not Found	Invalid {siteCode}

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Site record. Use this function to determine when a site's details were last updated.

Endpoint address: /services/rest/site/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/site/87

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Site record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Site record. Use this function to create new sites in Brand Compliance based on data sourced from the external system.

Dependencies: The Supplier must be present in the application and its record ID obtained. If assigning a Business Category or Product Technologist, the record must be present in the application and its record ID obtained. Create Contacts after creating the Site. For more information, see "Dependencies".

Endpoint address: /services/rest/site
HTTP method: POST

Request Details

The body of the request contains a SiteFullDTO element to specifying the details of the site to create. Compared to retrieving a site (which uses the same SiteFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created Site record need to be included. As a minimum, this must include the fields shown in the following table.

Site Mandatory Fields

Field Name	Element Name
Site Name	name
Site Type	siteType/code
Categories	businessCategories
Lead Product Technologist	leadTechnicalManager
Supplier	supplier/code
Site Status	siteStatus

Example Request XML

This example shows the minimum requirement to be able to create a site against a supplier.

```
<ns1:siteFullDTO
    xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
    xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">

       <ns0:businessCategories>
              <ns1:code>CATEGORY2A</ns1:code>
       </ns0:businessCategories>
       <ns0:leadTechnicalManager>
              <ns1:code>techadmin</ns1:code>
       </ns0:leadTechnicalManager>

       <ns0:name>Site Name</ns0:name>
       <ns0:siteType>
              <ns1:code>SITE_TYPE_EXAMPLE</ns1:code>
       </ns0:siteType>
       <ns0:siteStatus>
              <ns1:status>ACTIVE</ns1:status>
       </ns0:siteStatus>
       <ns0:supplier>
              <ns1:code>WS0001</ns1:code>
       </ns0:supplier>

</ns1:siteFullDTO>
```

Where the record is linked to another record, such as the Site Type in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a SiteLink root element containing the site data, and a SupplierLink element, consisting of the parent supplier data. The returned elements are shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Site record's internal ID
recordLink	string	URI to the newly created Site record, for use in a GET request
code	string	Supplier and site code business key to the newly created Site record
name	string	Site's name
localName	string	Site’s name in business language (if used)
supplierLink/recordId	long	Newly created site’s Supplier record internal ID
supplierLink/recordLink	string	URI to the newly created site’s Supplier record, for use in a GET request
recordLink/code	string	Supplier code business key to the newly created site’s Supplier record
recordLink/name	string	Supplier's name
recordLink/localName	string	Supplier’s name in business language (if used)

Error Messages

If the supplied data does not result in a valid Site (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
supplier/id	Null Pointer Exception	Supplier ID not provided
supplier/id	No row with the given identifier exists: [com.micros.creations.core.domain.Supplier#<###>]	Supplier not found
siteStatus	Null Pointer Exception	Site Status not provided
siteStatus	Cannot locate keyword for class com.micros.creations.core.domain.SiteStatus using status:<###>	Site Status not found
siteType	ERROR: class com.micros.creations.core.domain.Site.siteType - The condition is invalid	Site Type not provided
siteType	Cannot locate keyword for class com.micros.creations.core.domain.SiteType using code:<###>	Site Type not found

Element	Message	Meaning
name	class com.micros.creations.core.domain.Site .name - The condition is invalid	Site name not provided
leadTechnicalManager	ERROR: class com.micros.creations.core.domain.Site .leadTechnicalManager - The condition is invalid	Lead Technical Manager not provided
leadTechnicalManager	IllegalStateException: Cannot locate keyword for class com.micros.creations.core.domain.User using code:<###>	Lead Technical Manager not found
businessCategories	ERROR: class com.micros.creations.core.domain.Site .businessCategories - The condition is invalid	Business Category not provided
businessCategories	Cannot locate keyword for class com.micros.creations.core.domain.BusinessCategory using code:<###>	Business Category not found

Update Record

Description

Updates an existing Site record. Use this function to update a site's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/site/{id}
HTTP method: PUT

Request Details

The body of the request contains a SiteUpdateDTO to specify the updates to the Site record. Compared to retrieving a site (which uses the SiteFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is similar to that for creating a site, but crucially, the links to other top-level records (SiteContact) are omitted. The omission of those ensures that when updating a site, only the site details need to be specified, and not the details for the related records that may not require updating (and which should be updated with calls to their respective services). After the call, the Site record is updated to match the request

Note: When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Site record (except for the SiteContacts).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a SiteLink root element containing the site data, and a SupplierLink element, consisting of the parent supplier data. The returned elements are shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Updated Site record's internal ID
recordLink	string	URI to the updated Site record, for use in a GET request
code	string	Site code business key to the updated Site record
name	string	Site's name
localName	string	Site’s name in business language (if used)
supplierLink/recordId	long	Updated site’s Supplier record internal ID
supplierLink/recordLink	string	URI to the updated site’s Supplier record, for use in a GET request
recordLink/code	string	Supplier code business key to the updated site’s Supplier record
recordLink/name	string	Supplier's name
recordLink/localName	string	Supplier’s name in business language (if used)

Error Responses

If the supplied data does not result in a valid Site (such as a missing mandatory field), an HTTP 417 response is sent an with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

ContactRestService

This section describes the API for managing supplier and site contacts. The following functions are available:

- List of Values: retrieves a list of contacts
- Retrieve Record by ID: retrieves a Contact record using its unique identifier
- Retrieve Supplier Contact Record by Business Key: retrieves a Supplier Contact record using its business key
- Retrieve Site Contact Record by Business Key: retrieves a Site Contact record using its business key
- Check Record Modification Timestamp: retrieves the timestamp when a Contact record was last updated
- Create Record: creates a new Contact record
- Update Record: updates an existing Contact record

List of Values

Description

Retrieves a list of contacts in a paged list. Use this function to retrieve a simple list of contact names and IDs, or to locate Contact record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/contact
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
name	Optional	string	Yes	John Smith
supplierContactRoles	Optional	string	Yes	HEAD OF MANUFACTURING
siteContactRoles	Optional	string	Yes	HEAD OF MANUFACTURING
country	Optional	string	Yes	UK
selectedSites	Optional	string	No	A0001-0001
supplierCode	Optional	string	Yes	A0001
supplierName	Optional	string	Yes	ABC Ltd.
siteStatus	Optional	string	Yes	ACTIVE~INACTIVE
supplierActive	Optional	string	No	TRUE or blank
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39

Example URLs

…/services/rest/contact/?offset=2&pageSize=20
…/services/rest/contact/?siteStatus=ACTIVE
…/services/rest/contact/?supplierName=API%

Response Details

For a successful response, XML is returned with a ContactLinkList root element containing an entries element for each matched contact. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Contact record's internal ID
recordLink	string	URI to the ContactRestService Retrieve record service for this contact
email	string	Contact’s email address
name	string	Contact's name
siteContact	Boolean	Indicates if a contact for the site (true), else false
supplierContact	Boolean	Indicates if a contact for the supplier (true), else false

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
supplierContactRoles	IllegalStateException: The Supplier Contact Role <###> does not exist	Invalid supplier contact role
supplierContactRoles	IllegalStateException: The Supplier Contact Roles <###~###> do not exist	Invalid supplier contact role
siteContactRoles	IllegalStateException: The Site Contact Role <###> does not exist	Invalid site contact role
siteContactRoles	IllegalStateException: The Site Contact Roles <###~###> do not exist	Invalid site contact role
country	IllegalStateException: The Country <###> does not exist	Invalid country code
country	IllegalStateException: The Countries <###~###> do not exist	Invalid country code
siteStatus	IllegalStateException: The Supplier Status <###> does not exist	Invalid supplier status code

Retrieve Record by ID

Description

Retrieves a single Contact record's details using the record's unique ID. Use this function to retrieve the full details of an individual supplier or site contact.

Endpoint address: /services/rest/contact/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/contact/405

Response Details

For a successful response, XML is returned with a contactAndPersonDTO root element containing the individual attributes of the requested Contact record and the person it associates to.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Contact to return with id:<###>	Invalid {id} - not found

Retrieve Supplier Contact Record by Business Key

Description

Retrieves a single Supplier Contact record's details using its business key (supplier code, contact name, and email address). Use this function to retrieve the full details of an individual supplier contact using the combination of the Brand Compliance supplier codes, name of the contact, and the contact's email address.

Endpoint address:
/services/rest/contact/byKey/supplier/{supplierCode}/{name}/{email}
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
supplierCode	Mandatory	string	No	A0001
name	Mandatory	string	No	John Doe
email	Mandatory	string	No	john.doe@email.com

Example URL

…/services/rest/contact/byKey/supplier/A0001/John Doe/john.doe@email.com

Response Details

For a successful response, XML is returned with a contactAndPersonDTO root element containing the individual attributes of the requested Contact record and the person it associates to.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Responses

In the document cannot be located, an HTTP 404 response is sent.

Retrieve Site Contact Record by Business Key

Description

Retrieves a single Site Contact record's details using its business key (supplier code, site code, contact name, and contact email). Use this function to retrieve the full details of an individual site contact using the combination of the Brand Compliance supplier and site codes, name of the contact, and the contact's email address.

Endpoint address:
/services/rest/contact/byKey/site/{supplierCode}/{siteCode}/{name}/{email}
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
supplierCode	Mandatory	string	No	A0001
siteCode	Mandatory	string	No	A0001-0001
name	Mandatory	string	No	John Doe
email	Mandatory	string	No	john.doe@email.com

Example URL

…/services/rest/contact/byKey/site/A0001/A0001-0001/
John Doe/john.doe@email.com

Response Details

For a successful response, XML is returned with a contactAndPersonDTO root element containing the individual attributes of the requested Contact record and the person it associates to.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Responses

In the document cannot be located, an HTTP 404 response is sent.

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Contact record. Use this function to determine when a supplier or site contact's details were last updated.

Endpoint address: /services/rest/contact/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/contact/405

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header to show the last modification date and time of the last update of the requested Contact record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Contact record (and the associated Person record). Use this function to create new supplier or site contacts in Brand Compliance based on data sourced from the external system.

Dependencies: The Supplier (and Site, if a site contact) must be present in the application and its record ID obtained. Omit the Site element if not a site contact. The User/Person must be present in the application and its record ID obtained. For more information, see "Dependencies".

Endpoint address: /services/rest/contact
HTTP method: POST

Request Details

The body of the request contains a ContactAndPersonDTO to specify the details of the contact to create. Compared to the retrieving a contact (which uses the same ContactAndPersonDTO type), this request is much shorter. Only the attributes that are to be populated on the created Contact record need to be included. As a minimum, this must include the fields shown in the following table.

Contact Mandatory Fields

Field Name	Element Name
Name	contactFullDTO/person/name personFullDTO/name
User Id	contactFullDTO/person/id
Supplier	contactFullDTO/person/supplier/id contactFullDTO/company/code personFullDTO/supplier/code personFullDTO/supplier/id
Email	contactFullDTO/person/email personFullDTO/email
Phone	contactFullDTO/contactDetails/phoneNumber
Supplier Contact Roles	contactFullDTO/supplierContactRole/code
Contact Type	contactFullDTO/dtype
Site Contact Flag	contactFullDTO/siteContact
Supplier Contact Flag	contactFullDTO/supplierContact

Example Request XML

This example shows the minimum requirement to be able to create a contact against a supplier.

```
<ns0:ContactAndPersonDTO
xmlns:ns4="http://www.oracle.com/orbcmcs/service/rest/model"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full"
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple">
       <ns0:contactFullDTO>
              <ns0:contactDetails>
                     <ns0:fax>fax</ns0:fax>
                     <ns0:phoneNumber>phone</ns0:phoneNumber>
                     <ns0:id>167</ns0:id>
              </ns0:contactDetails>
              <ns0:dtype>SupplierContact</ns0:dtype>
              <ns0:person>
                     <ns1:email>SA1@example.com</ns1:email>
                     <ns1:name>API Supplier Admin 1</ns1:name>
                     <ns1:supplier>
                            <ns1:code>WS0001</ns1:code>
                     </ns1:supplier>
                     <ns1:id>49</ns1:id>
              </ns0:person>
              <ns0:siteContact>false</ns0:siteContact>
              <ns0:siteSelection>SELECTED_SITES</ns0:siteSelection>
              <ns0:supplierContact>true</ns0:supplierContact>
              <ns0:supplierContactRole>
                     <ns1:code>ACCOUNT MANAGER</ns1:code>
                     <ns1:id>5</ns1:id>
              </ns0:supplierContactRole>
              <ns0:company>
                     <ns1:code>WS0001</ns1:code>
              </ns0:company>
       </ns0:contactFullDTO>
       <ns0:personFullDTO>
              <ns0:contactDetails>
                     <ns0:id>166</ns0:id>
              </ns0:contactDetails>
              <ns0:email>SA1@example.com</ns0:email>
              <ns0:name>API Supplier Admin 1</ns0:name>
              <ns0:supplier>
                     <ns1:code>WS0001</ns1:code>
                     <ns1:companyNumber>123</ns1:companyNumber>
              </ns0:supplier>
              <ns0:id>49</ns0:id>
       </ns0:personFullDTO>
</ns0:ContactAndPersonDTO>
```

If a contact has multiple supplier or site contact roles, they should be separate entries as follows:

```
<ns0:supplierContactRole>
       <ns1:code>ACCOUNT MANAGER</ns1:code>
</ns0:supplierContactRole>
<ns0:supplierContactRole>
       <ns1:code>AUDITS AND VISITS CONTACT</ns1:code>
</ns0:supplierContactRole>
<ns0:supplierContactRole>
       <ns1:code>EMERGENCY CONTACT</ns1:code>
</ns0:supplierContactRole>
```

Where the record is linked to another record, such as the Contact Role in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ContactLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Contact record's internal ID
recordLink	string	URI to the newly created Contact record, for use in a GET request
email	string	Email address of the newly created contact
name	string	Contact's name
siteContact	Boolean	Indicates if a contact for the site (true), else false
supplierContact	Boolean	Indicates if a contact for the supplier (true), else false

Error Messages

Element	Message	Meaning
phoneNumber	phoneNumber - The condition is invalid	Missing tag

Element	Message	Meaning
dtype	IllegalArgumentException: Invalid value for dtype:<###>	Invalid value SupplierContact; SiteContact
name & email	IllegalStateException: Cannot locate Person record with name: <###> and email: <###>	A valid user must be found with matching name and email address
company	IllegalStateException: Cannot locate keyword for class com.micros.creations.core.domain.Company using code:<###>	Invalid company/supplier code

Update Record

Description

Updates an existing Contact record. Use this function to update a supplier or site contact's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/contact/{id}
HTTP method: PUT

Request Details

The body of the request contains a ContactAndPersonDTO element to specify the updates to the Contact record. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is the same as that for creating a contact. After the call, the Contact record is updated to match the request.

> **Note:** When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Contact record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ContactLink element. The ContactLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Contact record's internal ID
recordLink	string	URI to the Contact record, for use in a GET request
email	string	Contact’s email address
name	string	Contact's name
siteContact	Boolean	Indicates if a contact for the site (true), else false

Element	Type	Description
supplierContact	Boolean	Indicates if a contact for the supplier (true), else false

Error Messages

If the supplied data does not result in a valid Contact (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

ProductRecordRestService

This section describes the API for managing Product records. The following functions are available:

- List of Values: retrieves a list of Product records
- Retrieve Record by ID: retrieves a Product record using its unique identifier
- Retrieve Record by Business Key: retrieves a Product record using its business key
- Check Record Modification Timestamp: retrieves the timestamp when a Product record was last updated
- Create Record: creates a new Product record
- Update Record: updates an existing Product record

List of Values

Description

Retrieves a list of Product records in a paged list. Use this function to retrieve a simple list of product titles and IDs, or to locate Product record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/productRecord
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
specType	Optional	string	Yes	FOOD~CNF
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39

Example URLs

…/services/rest/productRecord/?offset=2&pageSize=20
…/services/rest/productRecord/?specType=FOOD

Response Details

For a successful response, XML is returned with a ProductRecordLinkList root element containing an entries element for each matched Product Record. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Product record's internal ID
recordLink	string	URI to the ProductRecordRestService Retrieve record service for this Product record
code	string	Code business key to the Product record
title	string	Product’s title

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
specType	IllegalStateException: The Spec Type <###> does not exist	Invalid specification type
modifiedSince	Invalid dates cause stack trace to come back to "response"	
modifiedUntil	Invalid dates cause stack trace to come back to "response".	

Retrieve Record by ID

Description

Retrieves a single Product record’s details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual Product Record.

Endpoint address: /services/rest/productRecord/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/productRecord/99

Response Details

For a successful response, XML is returned with a productRecordFullDTO root element containing the individual attributes of the requested Product record. If an ID is not specified, a list of all Product records is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Product Record to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Product record's details using its business key (code). Use this function to retrieve the full details of an individual Product record using its Brand Compliance product code.

The Product record's business key is the product code that is visible in the UI. This an individual number assigned by Brand Compliance. It is not the product's traded unit identifier such as a SKU, that is, the product number, which is also visible in the UI.

Endpoint address: /services/rest/productRecord/byKey/{code}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {code} parameter that determines the record to retrieve.

Example URL

…/services/rest/productRecord/byKey/31

Response Details

For a successful response, XML is returned with a productRecordFullDTO root element containing the individual attributes of the requested Product record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not Found	Invalid {code} - blank
code	HTTP 404 Not Found	Invalid {code} - not found

Element	Message	Meaning
code	NumberFormatException: For input string: <###>	Invalid {code} - not numeric

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Product record. Use this function to determine when a Product record's details were last updated.

Endpoint address: /services/rest/productRecord/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/productRecord/99

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header to showing the date and time of the last update of the requested Product record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Product record. Use this function to create new Product records in Brand Compliance based on data sourced from the external system.

Dependencies: If linked to a supplier or site, the Supplier/Site must be present in the application and its record ID obtained. If assigning a Business Category, Product Technologist, or Other Contact, the record must be present in the application and its record ID obtained. Product Technologist and Other Contacts may be omitted to default to TBC if the TBC user is present in the application. The status must be Active in order to be linked to a specification; for the status to be Active, the Product record must be linked to a supplier. For more information, see "Dependencies".

Endpoint address: /services/rest/productRecord
HTTP method: POST

Request Details

The body of the request contains a ProductRecordFullDTO to specify the Product record to create. Compared to retrieving a Product record (which uses the same ProductRecordFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created Product record need to be included. As a minimum, this must include the fields shown in the following table.

Product Record Mandatory Fields

Field Name	Element Name
Product Title	title
Status	productRecordStatus
Variant Name	variantName
Quantity	quantity
Specification Type	specType - see below
Specification Type Format	specTypeFormat - see below
Technologist	technologist
Other Contacts	productRecordOtherUser
Supplier Details	supplier
Site Details	site

Example Request XML

This example shows the minimum requirement to be able to create a Product record.

```
<ns0:productRecordFullDTO
    xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
    xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">

       <ns0:code>129</ns0:code>
       <ns0:productCovered>
             <ns0:quantity>Small</ns0:quantity>
             <ns0:retailerProductNumber>ABC123</ns0:retailerProductNumber>
             <ns0:variantName>Small White Rolls</ns0:variantName>
       </ns0:productCovered>
       <ns0:productRecordOtherUser>
              <ns0:role>
                     <ns1:code>BUYER</ns1:code>
              </ns0:role>
              <ns0:user>
                     <ns1:code>TBC</ns1:code>
              </ns0:user>
       </ns0:productRecordOtherUser>
       <ns0:productRecordOtherUser>
              <ns0:role>
                     <ns1:code>PRODUCT DEVELOPMENT MANAGER</ns1:code>
              </ns0:role>
              <ns0:user>
                     <ns1:code>TBC</ns1:code>
              </ns0:user>
       </ns0:productRecordOtherUser>
       <ns0:specTypeFormat>
              <ns1:code>FOODUK</ns1:code>
              <ns1:specType>FOOD</ns1:specType>
```

```
        </ns0:specTypeFormat>
        <ns0:technologist>
               <ns1:code>producttechnologist</ns1:code>
        </ns0:technologist>
        <ns0:title>White Bread Rolls</ns0:title>
        <ns0:productRecordStatus>
               <ns1:status>DRAFT</ns1:status>
        </ns0:productRecordStatus>

</ns0:productRecordFullDTO>
```

The productCoverage element holds the product's title, quantity, and code regardless of whether the portal is operating in multi variant mode where a single Product record may represent multiple variants (such as sizes) of the product, or in single variant mode, where the Product record represents just a single variant of the product. With single variant mode, there will only ever be a single productCoverage element.

Single Variant Mode

```
<productCovered>
   <productTitle/>
   <variantName/>
   <quantity/>
   <retailerProductNumber/>
</productCovered>
```

Multi Variant Mode

```
<productCovered>
    <productTitle/>
    <variantName/>
    <quantity/>
    <retailerProductNumber/>
</productCovered>
<productCovered>
    <productTitle/>
    <variantName/>
    <quantity/>
    <retailerProductNumber/>
</productCovered>
```

If a specification type has more than one format, such as a Food specification having separate formats for Food - UK and Food - US, the specTypeFormat/code element is used to specify the specification format and specType is used to identify the specification type, for example:

```
<ns0:specTypeFormat>
   <ns1:code>FOODUK</ns1:code>
   <ns1:specType>FOOD</ns1:specType>
</ns0:specTypeFormat>
```

The code of the specification type/format must be used rather than the description.

Where the record is linked to another record, such as the Technologist in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

Note: The API does not apply status-based validation as is applied using the Brand Compliance UI when manually progressing a Product record.

It is therefore possible, using the API, to create a Product record at Active status without the supplier or site being populated. In this case, it will not be possible to link Product Specifications to the Product record until a valid supplier and site has been assigned manually.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ProductRecordLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Product record's internal ID
recordLink	string	URI to the newly created Product record, for use in a GET request
code	string	Code business key to the newly created Product record
title	string	Product’s title

Error Messages

If the supplied data does not result in a valid Product record (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
productCoverage	ERROR: class <<hostname>>.ProductRecord.productCovered - The condition is invalid	No tag provided
code	ERROR: class <<hostname>>.ProductRecord.code - A Product Record's code is mandatory	No tag provided
code	ERROR: class <<hostname>>.ProductRecord.code - A Product Record's code is mandatory	No value provided
code	ERROR: class <<hostname>>.ProductRecord.code - A Product Record's code must be unique:<###>	Code already exists
code	HTTP Status 400 - Bad request	Code is not numeric

Element	Message	Meaning
title	DtoValidationException: title is mandatory	No tag provided No value provided
productRecordStatus	IllegalStateException: Cannot locate keyword for class <<hostname>>.PoductRecordStstus using status:<###>	No tag provided No value provided Status not found
variantName	ERROR: class <<hostname>>.ProductRecord.productCovered[###].variantName - Variant name is mandatory	No tag provided No value provided
quantity	ERROR: class <<hostname>>.ProductRecord.productCovered[###].quantity - Quantity name is mandatory	No tag provided No value provided
specType	ObjectNotFoundException: No row with the given identifier exists:[<<hostname>>.SpecTypeFormat#<###>]	No value provided Specification type not found
technologist	MySQLIntegrityConstraintViolationException: Cannot add or update a child row: a foreign key constraint fails ('<###>'.'t_prdrcd',CONSTRAINT '<###>' FOREIGN KEY ('fk_technologist') REFERENCES 't_user' ('c_id'))	No tag provided No value provided
technologist	ObjectNotFoundException: No row with the given identifier exists: [com.micros.creations.core.domain.User#<###>]	Technologist not found
productRecordOtherUser /user	DtoValidationException: ProductRecordOtherUser has no valid user for role:<###> ProductRecordOtherUser Roles are mandatory for the following missing Roles: [<###>]	No tag provided User not found
productRecordOtherUser /user	ObjectNotFoundException: No row with the given identifier exists:[<<hostname>>.Usert#<###>]	No value provided
productRecordOtherUser /role	DtoValidationException: ProductRecordOtherUser has no valid user for role:<###> ProductRecordOtherUser Roles are mandatory for the following missing Roles: [<###>]	No tag provided User not found
productRecordOtherUser /role	ObjectNotFoundException: No row with the given identifier exists:[<<hostname>>.Role#<###>]	No value provided
supplier	IllegalStateException: Cannot locate keyword for class <<hostname>>.Company using code:<###>	No tag provided

Element	Message	Meaning
supplier	MySQLIntegrityConstraintViolationException: Cannot add or update a child row: a foreign key constraint fails ('<###>'.'t_prdrcd',CONSTRAINT '<###>' FOREIGN KEY ('fk_supplier') REFERENCES 't_company' ('c_id'))	No value provided Supplier not found
site	IllegalStateException: Cannot locate keyword for class <<hostname>>.Site using id:<###>, code:<###>, supplierCode:<###> and supplierId:<###>	No tag provided No value provided Supplier not found
countryWhereSold	IllegalStateException: Cannot locate keyword for class <<hostname>>.CountryWhereSold using code:<###> and spec type:<###>	No value provided
countryWhereSold	ObjectNotFoundException: No row with the given identifier exists:[<<hostname>>.CountryWhereSold#<###>]	Country not found
	ERROR: class <<hostname>>.ProductRecord.barcode[<###> - Bar Code must be either 8 or 13 characters long	Invalid code length
	ERROR: class<<hostname>>.ProductRecord.shippingCaseCode[<###> - Shipping Case Code must be either 14 digits long	Invalid code length

Update Record

Description

Updates an existing Product record. Use this function to update a Product record's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/productRecord/{id}
HTTP method: PUT

Request Details

The body of the request contains a ProductRecordFullDTO to specify the updates to the Product record. Compared to retrieving a Product record (which uses the same ProductRecordFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is the same as that for creating a Product Record. After the call, the Product Record is updated to match the request.

> **Note:** When updating records, all values must be included. If a value is or element is omitted from the request, the field contents will be cleared on the Product record.
>
> When a Product Specification is initially linked to the Product record, any values in the product title, barcode, shipping case code, and quantity fields are carried through to the Specification. If subsequent changes are made to the barcode or shipping case code using the Product Record API, the changes will not be automatically cascaded to the Specification, however, if the Specification is at draft status it may be manually delinked and relinked to the Product record in order to refresh these values.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ProductRecordLink element. The ProductRecordLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Product record's internal ID
recordLink	string	URI to the Product record, for use in a GET request
code	string	Code business key to the Product record
title	string	Product’s title

Error Responses

If the supplied data does not result in a valid Product record (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

ProductSpecificationRestService

This section describes the API for managing Product Specifications. The following functions are available:

- List of Values: retrieves a list of Product Specifications
- Retrieve Record by ID: retrieves a Product Specification using its unique identifier
- Retrieve Record by Business Key: retrieves a Product Specification using its business key
- Check Record Modification Timestamp: retrieves the timestamp when a Product Specification was last updated
- Retrieve List with Advanced Filtering: retrieves a Product Specification using additional filtering options

- Create Record: creates a new Product Specification record
- Update Record: updates an existing Product Specification record

List of Values

Description

Retrieves a list of Product Specifications in a paged list. Use this function to retrieve a simple list of product titles and IDs, or to locate Product Specification IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/specification
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
specType	Optional	string	Yes	FOOD~CNF
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39

Example URLs

…/services/rest/specification/?offset=2&pageSize=20
…/services/rest/specification/?specType=FOOD

Response Details

For a successful response, XML is returned with a ProductSpecificationLinkList root element containing an entries element for each matched Product Specification. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Product Specification's internal ID
recordLink	string	URI to the ProductSpecificationRestService Retrieve record service for this specification
specNumber	string	Specification number business key to the Product Specification record
specVersion	int	Specification version business key to the Product Specification record
title	string	Product Specification’s title

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
	<errorCode>NOFILTER <errorMessage>No Filter Specified, please try again	No parameter provided
batchsize	<errorCode>INVALIDBATCHSIZE <errorMessage>The Batch Size must be between 1 and 100	Invalid batch size No batch size provided No batch size tag
specStatus	<errorCode>INVALIDSPECSTATUS <errorMessage>The Status <###> does not exist	Spec Status not found Spec Status is blank
specType	<errorCode>INVALIDSPECTYPE <errorMessage>The Spec Type <###> does not exist	Spec Type not found Spec Type is blank
countryWhereSold	<errorCode>INVALIDCOUNTRYWHERESOLD <errorMessage>The Country Where Sold >###> does not exist	Country not found Country is blank
language	<errorCode>INVALIDLANGUAGE <errorMessage>The Language <###> does not exist	Language not found Language is blank
sectionType	<errorCode>INVALIDSECTIONTYPE <errorMessage>The Section Type does not exist	Section type not found Section type is blank
fromDate	HTTP 400 Bad Request	Invalid date/time format
toDate	HTTP 400 Bad Request	Invalid date/time format

Retrieve Record by ID

Description

Retrieves a single Product Specification’s details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual specification.

Endpoint address: /services/rest/specification/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/specification/97

Response Details

For a successful response, XML is returned with a productSpecificationFullDTO root element containing the individual attributes of the requested specification. If an ID is not specified, a list of all specifications is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 5 - Product (CNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 6 - Product (FNF Specification)*, and *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 7 - Product (BWS Specification)* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Product Specification to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Product Specification's details using its business key (specification number and version). Use this function to retrieve the full details of an individual specification using its Brand Compliance specification number and version.

Endpoint address: /services/rest/specification/byKey/{specNumber}/{specVersion}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {specNumber} and {specVersion} parameters that determines the record to retrieve.

Example URL

.../services/rest/specification/byKey/104/1

Response Details

For a successful response, XML is returned with a productSpecificationFullDTO root element containing the individual attributes of the requested specification.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 5 - Product (CNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 6 - Product (FNF Specification)*, and *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 7 - Product (BWS Specification)* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
specNumber	HTTP 404 Not Found	Invalid {specNumber} - blank
specNumber	HTTP 404 Not Found	Invalid {specNumber} - not found
specVersion	HTTP 404 Not Found	Invalid {specVersion} - blank
specVersion	HTTP 404 Not Found	Invalid {specVersion} - not found

Retrieve List with Advanced Filtering

Description

Retrieves a list of Product Specifications in a paged list using advanced filtering. Use this function to retrieve specifications based on the values of specific fields.

Endpoint address: /services/rest/specification/advanced
HTTP method: POST

Request Details

Parameters are passed in the request in XML format.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example	Validation
offset	Mandatory	int	No	60	>0
batchsize	Mandatory	int	No	30	>0 and <=100
specType	Optional	string	Yes	FOOD	FOOD FNF CNF PRODUCE BWS

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example	Validation
specStatus	Optional	string	Yes	RETAILER_ DRAFT	SUPPLIER_DRAFT RETAILER_DRAFT COLLABORATIVE_ DRAFT GATE_STEP PART_PACK_ COPY_SENT PACK_COPY_SENT PACK_COPY_ READY READY_FOR_ AUTHORISATION SUPPLIER_ AUTHORISED ACTIVE OFF_RANGE DE_LISTED SUPERSEDED NOT_ PROGRESSED APPROVE_FOR_ LABELLING PRODUCE_DRAFT PRODUCE_PACK_ COPY PRODUCE_ APPROVED PRODUCE_ ARCHIVED
productNumber	Optional	string	Yes	12345	Filters on the Product Number column of the Product Coverage table.
language	Optional	string	Yes	en_GB	The Code for the required pack copy language, such as, en_GB. Specifying a preferred language will return only values in that language. If not specified, the base language is assumed.

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example	Validation
sectionType	Optional	string	Yes	RECIPE AND RAW MATERIALS	MAIN DETAILS RECIPE AND RAW MATERIALS NUTRITION ALLERGY AND DIETARY ADVICE PACKAGING FINISHED PRODUCT STANDARDS STORAGE OTHER LABELLING COPY CLAIMS SUBSTANTIATION PROCESS CONTROLS BATCH CODING PRODUCT APPROVAL REQUIREMENTS POST LAUNCH INFORMATION COMPONENTS FNF STORAGE PRODUCT REQUIREMENTS COUNTER TICKET PRODUCT CHARACTER COMP
countryWhereSold	Optional	string	Yes	FOOD_ COUNTRY_ WHERE_ SOLD	The Code for the required Country Where Sold.
Product Coverage	Optional	string	Yes	Chicken (500g)	Filters on the Product Name column of the Product Coverage table.
supplier	Optional	string	Yes	A0001	Filters on the Primary Sites supplier code for Produce Specifications or the Supplier code for all other types of Specification.

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example	Validation
site	Optional	string	Yes	A0001-001	Filters on the code of the Sites in the Primary Sites table of the Product Specification. Use the full site code incorporating the Supplier Code, such as A0001-001.
fromDate	Optional	string	No	2013-06-10T09:00:00	Can be used in conjunction with toDate to form a date range or can be specified individually.
toDate	Optional	string	No	2013-06-10T09:00:00	Can be used in conjunction with fromDate to form a date range or can be specified individually.

For example, to retrieve the Main Details and Nutrition sections for active Food specifications where the product number is ABC001 from a specific supplier/site, the request would be:

```
<request>
  <batchsize>100</batchsize>
  <offset>0</offset>
  <specStatus>ACTIVE</specStatus>
  <specType>FOOD</specType>
  <countryWhereSold>UK</countryWhereSold>
  <productNumber>ABC001</productNumber>
  <language>en_GB</language>
  <sectionType>MAIN DETAILS</sectionType>
  <sectionType>NUTRITION</sectionType>
  <supplier>A0001</supplier>
  <site>A0001-0001</site>
  <fromDate>2016-01-01T01:00:00</fromDate>
  <toDate>2016-10-01T01:00:00</toDate>
</request>
```

If no filtering is required, omit the parameter from the call. For example, if all specification types are to be included, omit the specType element (rather than including it with no value specified).

If multiple values for a parameter are to be included, repeat the parameter in the call. For example, if Main Details and Nutrition section type are to be returned, include two specSection elements in the call, one for MAIN DETAILS and one for NUTRITION.

Searching with either fromDate or toDate specified, as well as one or more specStatus, filters the specifications on the Status Change History. Use this to retrieve specifications that changed to a specific status during a date range.

Searching with either fromDate or toDate specified without specStatus filters on the specification's last amended date.

Response Details

For a successful response, XML is returned with a GetProductSpecificationServiceResponse root element containing details of the specifications that match the selection criteria.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 5 - Product (CNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 6 - Product (FNF Specification)*, and *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 7 - Product (BWS Specification)* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
	<errorCode>NOFILTER <errorMessage>No Filter Specified, please try again	No parameter provided
batchSize	<errorCode>INVALIDBATCHSIZE <errorMessage>The Batch Size must be between 1 and 100	Invalid batch size No batch size provided No batch size tag
specStatus	<errorCode>INVALIDSPECSTATUS <errorMessage>The Status <###> does not exist	Spec Status not found Spec Status is blank
specType	<errorCode>INVALIDSPECTYPE <errorMessage>The Spec Type <###> does not exist	Spec Type not found Spec Type is blank
countryWhereSold	<errorCode>INVALIDCOUNTRYWHERESOLD <errorMessage>The Country Where Sold >###> does not exist	Country not found Country is blank
language	<errorCode>INVALIDLANGUAGE <errorMessage>The Language <###> does not exist	Language not found Language is blank
sectionType	<errorCode>INVALIDSECTIONTYPE <errorMessage>The Section Type does not exist	Section type not found Section type is blank
fromDate	HTTP 400 Bad Request	Invalid date/time format
toDate	HTTP 400 Bad Request	Invalid date/time format

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Product Specification. Use this function to determine when a specification's details were last updated.

Endpoint address: /services/rest/specification/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/specification/97

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested specification.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Product Specification to return with id:<###>	Invalid {id} - not found

Create Record

Description

Creates a new Product Specification record. Use this function to create a new specification in Brand Compliance based on data sourced from the external system.

Dependencies: If linking to a Product Record, the active Product Record must be present and in the application and its record ID obtained; the Supplier and Site (and associated Users, Contacts, and Business Categories) must be present in the application and the record IDs obtained. For more information, see "Dependencies".

Endpoint address: /services/rest/specification
HTTP method: POST

Request Details

The body of the request contains a ProductSpecificationFullDTO to specify the detail of the specification to create. Compared to retrieving a specification (which uses the same ProductSpecificationFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created specification need to be included. As a minimum, this must include the fields shown in the following table.

Product Specification Mandatory Fields

Field Name	Element Name
Specification Name	title
Legislation	legislation
Pack Copy Language	packCopyLanguage / code

Field Name	Element Name
Specification Type	specTypeFormat / code
Specification Number (unique)	specNumber
Version	specVersion
Status	productSpecificationStatus / status
Multi-pack Specification	isMultipack
Main Details Section (container)	specificationSectionDetail/specificationSectionFoodMainDetailsSection

> **Note:** These fields represent the minimum necessary to create a specification. Various other fields will be mandatory in order to then progress the specification through its workflow.
>
> The Main Details Section "container" is necessary to create the specification's Main Details section. For specification types other than Food, the specificationSectionFoodMainDetailsSection will be alternatively named accordingly.

Example Request XML

This example shows the minimum requirement to be able to create a specification.

```
<ns0:productSpecificationFullDTO
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full"
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple">
   <ns0:isMultipack>false</ns0:isMultipack>
   <ns0:legislation>MODULE_TYPE_EU</ns0:legislation>
   <ns0:packCopyLanguage>
      <ns1:code>en_GB</ns1:code>
   </ns0:packCopyLanguage>
   <ns0:specNumber>8925</ns0:specNumber>
   <ns0:specTypeFormat>
      <ns1:code>FOODUK</ns1:code>
   </ns0:specTypeFormat>
   <ns0:specVersion>1</ns0:specVersion>
   <ns0:specificationSectionDetail>
      <ns0:specificationSectionFoodMainDetailsSection>
      </ns0:specificationSectionFoodMainDetailsSection>
   </ns0:specificationSectionDetail>
   <ns0:title>API Test Specification</ns0:title>
   <ns0:productSpecificationStatus>
      <ns1:status>RETAILER_DRAFT</ns1:status>
   </ns0:productSpecificationStatus>
</ns0:productSpecificationFullDTO>
```

Where the record is linked to another record, such as the Specification Type in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary,* *Volume 5 - Product (CNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 6 - Product (FNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 7 - Product (BWS Specification)*, and *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 8 - Product (Produce Specification)* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

For a successful response, an HTTP 200 response is sent with a body containing a ProductSpecificationLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Product Specification's internal ID
recordLink	string	URI to the newly created Product Specification record, for use in a GET request
specNumber	long	Specification Number business key to the newly created Product Specification record
specVersion	int	Specification Version business key to the newly created Product Specification record
title	string	Product Specification's title

Error Responses

If the supplied data does not result in a valid Product Specification (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Update Record

Description

Updates an existing Product Specification. Use this function to update a Product Specification's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/specification/{id}
HTTP method: PUT

Request Details

The body of the request contains a ProductSpecificationFullDTO element to specify the updates to the specification. Compared to retrieving a specification (which uses the same ProductSpecificationFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is similar to that for creating a specification. After the call, the Product Specification record is updated to match the request.

Note: When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Product Specification record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 6 - Product (FNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 7 - Product (BWS Specification)*, and *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 8 - Product (Produce Specification)* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ProductSpecificationLink element. The ProductSpecificationLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Product Specification's internal ID
recordLink	string	URI to the Product Specification record, for use in a GET request
code	string	Code business key to the Product Specification
specNumber	long	Specification Number business key to theProduct Specification record
specVersion	int	Specification Version business key to the Product Specification record
title	string	Specification’s title

Error Responses

If the supplied data does not result in a valid Product Specification (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating an Internal Error Id. The request should not be reattempted with the same content.

TaskRestService

This section describes the API for managing user tasks. The following function is available:

- List of Tasks: Retrieves a list of user tasks

List of Tasks

Description

Retrieves a list of tasks for a user in the given language. Use this function to retrieve a list of a specific user's entries in their Brand Compliance Task Manager app. Parameters are used to specify the name of the user, and the language of the returned task details.

Endpoint address: /services/rest/task
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
userId	Mandatory	string	No	John Smith
language	Mandatory	string	No	en_GB

Example URL

.../services/task/?language=en_GB

The userId parameter is the login ID of the user for which the tasks list is to be retrieved. The language parameter is the code of the language record/locale in which to retrieve the task details.

Response Details

For a successful response, XML is returned with a TaskDTOList root element containing a tasks element for each matched task. The root element consists of the elements in the following table.

Returned Elements

Element	Type	Description
message	string	Name of the task, in the language specified
messageId	string	A language-agnostic identifier for the task
myCreationsLink	string	A URI to the Brand Compliance system, which will open the list of items for that task
taskItemCount	int	The number of items in the task

Error Messages

Element	Message	Meaning
userId	userId required	User not provided
userId	Cannot find user with login id <###>	User not found
language	language required	Language not provided
language	Invalid locale format: <###>	Language not found

UrgentItemsRestService

This section describes the API for retrieving a count of a user's urgent items. The following function is available:

- Number of Urgent Items: retrieves the number of urgent items pending for a user

Number of Urgent Items

Description

Retrieves a count of the number of Urgent Item tasks for a user. Use this function to determine the number of pending urgent tasks a specific user has in their Brand Compliance Urgent Item Manager app. A parameter is passed to specify the name of the user.

Endpoint address: /services/rest/urgentItems
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
userId	Mandatory	string	No	John Smith

The userId parameter is the login ID of the user for which the number of pending urgent items are to be retrieved.

Response Details

For a successful response, XML is returned with an UrgentItemsModel root element containing an itemCount element that specifies the number of Urgent Items.

Error Messages

Element	Message	Meaning
userId	User not provided: userId required	User not provided
userId	Cannot find user with login id <###>	User not found

ArtworkRestService

This section describes the API for Artwork integration. The following functions are available:

- Started Activities: retrieves a list of Artwork Activities that have started
- Update Record: updates existing Artwork Activities
- SSO: enables single sign-on between Brand Compliance and the Artwork system

Started Activities

Description

Retrieves a list of Project Activities that have their useMyArtwork flag set, and their status has changed to Started within the specified date range. This function is used for the integration of Brand Compliance with an external Artwork Management system.

Endpoint address: /services/artwork/started
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	0
pageSize	Optional	int	No	30
fromDate	Mandatory	string	No	2014-12-30 23:59:59
toDate	Mandatory	string	No	2015-12-30 23:59:59

Response Details

For a successful response, XML is returned with a CreateArtworkRequestList root element containing an entries element for each matching Artwork project activity. A totalRecords element identifies the number of records returned. Separate activityDetails and project elements group the returned activity and project data. The elements consist of the elements in the following tables.

Returned Elements - activityDetails

Element	Type	Description
activityName	string	Project activity's title
activityRecordId	long	Project activity record's internal ID
actualStartDate	string	Actual start date of the project activity
checkpointOnly	Boolean	Is the project activity a checkpoint only type
criticalPath	Boolean	Is the project activity a critical path type
duration	int	Duration of the project activity
isGate	Boolean	Is the project activity a gate type
isKey	Boolean	Is the project activity a key type
projectId	long	Project record's internal ID
projectTitle	string	Project's title
proposedEndDate	string	Proposed end date of the project activity
proposedStartDate	string	Proposed start date of the project activity
responsibleUserRoles	string	Role names of the responsible users
sequenceNumber	string	Project activity's sequence number
statusCode	string	Code of the project activity's status

Returned Elements - projects

Element	Type	Description
categories	string	Names of the project's categories
masterProject	Boolean	If the project is a master type
projectId	string	Project's business key ID
projectManager	string	Name of the project's project manager
projectRecordId	long	Project record's internal ID
projectTitle	string	Project's title
status	string	Status of the project
supplier	string	Name of the supplier associated to the project
teamDetails/role	string	Name of the team role
templateFolder	string	Name of the project template folder
templateType	string	Type of the project's template
templateUsed	string	Name of the project's template

Error Messages

Element	Message	Meaning
fromDate	fromDate parameter required	From date not provided
toDate	toDate parameter required	To date not provided
fromDate	fromDate and toDate parameters are required	From and To dates not provided
toDate	From date must be before or equal to To date	From date is after To date

Update Record

Description

Updates the sub status of existing Artwork Project Activities. This function is used for the integration of Brand Compliance with an external Artwork Management system, to update the status of an Artwork activity from the external system.

Endpoint address: /services/artwork/update
HTTP method: PUT

Request Details

The body of the request contains an updateActivityStatus/activityUpdateRequest element consisting of the elements shown in the following table.

Request Elements

Element	Type	Description
activityName	string	Project activity's name
activityRecordId	string	Project activity record's internal ID

Element	Type	Description
projectId	string	ID business key to the Project record
projectRecordId	string	Project record's internal ID
subStatusCode	string	Code business key to the activity's sub status

After the call, the Project Activity record's sub status is updated to match the request.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 9 - Project* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an ArtworkActivityLink element. The ArtworkActivityLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Project activity's internal ID
recordLink	string	URI to the Product Activity record, for use in a GET request
code	string	Code business key to the Project Activity record
title	string	Project activity’s title

Error Messages

If the supplied data does not result in a valid Project Activity (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

SSO

Description

Enables single sign-on between Brand Compliance and the external Artwork Management system.

Endpoint address: /services/artwork/started
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
token	Mandatory	string	No	The token is automatically generated when the user clicks the Artwork link; it is not feasible to manually call this service.

Response Details

For a successful response, XML is returned.

Error Messages

If the values specified for a parameter are invalid, an HTTP 417 status is returned with an ErrorMessage XML body.

Element	Message	Meaning
token	Token required	Token not provided
token	Token not found	Token not found

BusinessCategoryService

This section describes the API for managing Business Categories. The following functions are available:

- List of Values: retrieves a list of categories
- Retrieve Record by ID: retrieves a Business Category record using its unique identifier
- Check Record Modification Timestamp: retrieves the timestamp when a Business Category record was last updated
- Create Record: creates a new Business Category record
- Update Record: updates an existing Business Category record
- Delete Record: deletes an existing Business Category record

List of Values

Description

Retrieves a list of categories in a paged list. Use this function to retrieve a simple list of categories and IDs, or to locate Business Category record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/businessCategory
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
code	Optional	string	Yes	POTATOES_LOOSE
parentCode	Optional	string	Yes	POTATOES
entityDescription	Optional	string	Yes	Loose Potatoes
specificationType	Optional	string	Yes	FOOD~PRODUCE
topLevelCategory	Optional	string	No	TRUE
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39

Example URLs

.../services/rest/businessCategory/?offset=2&pageSize=20
.../services/rest/businessCategory/?code=POTATOES_LOOSE
.../services/rest/businessCategory/?entityDescription=Potato%

The code parameter will attempt to find a single category with a matching code. The parentCode parameter will find any categories whose parent has the given code.

The entityDescription matches any category whose description (irrespective of language) matches the value given; this is just the description of the category in question and does not include those of parents.

The specificationType parameter is a tilde (~) separated list of types.

Note: Including a value in the specificationType filter will exclude any categories for which no specification type has been selected.

The topLevelCategory parameter determines whether only top-level categories (those with no parent) will be included in the results.

Response Details

For a successful response, XML is returned with a BusinessCategoryLinkList root element containing an entries element for each matched category. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Business Category record's internal ID
recordLink	string	URI to the BusinessCategoryService Retrieve record service for this category
code	string	Code business key to the Business Category record
entityDescription	string	Category's name

Element	Type	Description
path	string	Category's full path (including parent levels - separated by '/')

Error Messages

Element	Message	Meaning
specificationType	<###> is not a valid Specification Type	Specification type not found

Retrieve Record by ID

Description

Retrieves a single Business Category record's details using the record's unique ID. Use this function to retrieve the full details of an individual category.

Endpoint address: /services/rest/businessCategory/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/businessCategory/105

Response Details

For a successful response, XML is returned with a BusinessCategoryFullDTO root element containing the individual attributes of the requested Business Category record. If an ID is not specified, a list of all categories is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not found	Invalid {id} - not numeric
id	Invalid record id	Invalid {id} - not found

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Business Category record. Use this function to determine when a category's details were last updated.

Endpoint address: /services/rest/businessCategory/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/businessCategory/105

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Business Category record. For example:

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not found	Invalid {id} - not numeric

Create Record

Description

Creates a new Business Category record. Use this function to create a new category in Brand Compliance based on data sourced from the external system.

Dependencies: If a lower-level category, the parent Business Category must be present in the application and its record ID obtained. For more information, see "Dependencies".

Endpoint address: /services/rest/businessCategory
HTTP method: POST

Request Details

The body of the request contains a BusinessCategoryFullDTO to specify the details of the category to create. Compared to retrieving a user (which uses the same BusinessCategoryFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created Business Category record need to be included. As a minimum, this must include the fields shown in the following table:

Business Category Mandatory Fields

Field Name	Element Name
Code	code
Description	description
Parent Code	parentCode
Parent or Child level	topLevelCategory

Note: If creating a top level category, topLevelCategory must be set to true; if creating a lower level category, topLevelCategory must be set to false, and parentCode becomes mandatory (the parent category's code).

Example Request XML

```
<ns0:businessCategoryFullDTO
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full"
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple">
   <ns0:code>X1</ns0:code>
   <ns0:localeData>
      <ns0:description>Category X - Level 1</ns0:description>
   </ns0:localeData>
   <ns0:specificationType>PRODUCE</ns0:specificationType>
   <ns0:specificationType>CNF</ns0:specificationType>
   <ns0:specificationType>FOOD</ns0:specificationType>
   <ns0:specificationType>FNF</ns0:specificationType>
   <ns0:specificationType>BWS</ns0:specificationType>
   <ns0:topLevelCategory>true</ns0:topLevelCategory>
</ns0:businessCategoryFullDTO>
```

The following example adds a second level category to the above top-level parent category:

```
<ns0:businessCategoryFullDTO
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full"
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple">
   <ns0:code>X2</ns0:code>
   <ns0:localeData>
      <ns0:description>Category X - Level 2</ns0:description>
   </ns0:localeData>
   <ns0:parentCode>X1</ns0:parentCode>
   <ns0:specificationType>PRODUCE</ns0:specificationType>
   <ns0:specificationType>CNF</ns0:specificationType>
   <ns0:specificationType>FOOD</ns0:specificationType>
   <ns0:specificationType>FNF</ns0:specificationType>
   <ns0:specificationType>BWS</ns0:specificationType>
   <ns0:topLevelCategory>false</ns0:topLevelCategory>
</ns0:businessCategoryFullDTO>
```

The value of the code element must be unique.

The description and path elements are locale-dependent, so are contained within a localeData element, with a locale element supported language code, for example:

```
<ns0:localeData>
        <ns0:description>Cheese</ns0:description>
        <ns0:locale>en_GB</ns0:locale>
</ns0:localeData>

<ns0:localeData>
       <ns0:description>Fromage</ns0:description>
       <ns0:locale>fr</ns0:locale>
</ns0:localeData>
```

The hierarchy of category levels is maintained through the parent code, linking a category to its immediate parent. All levels except the top level must therefore have a valid parent code specified.

The number of levels in the hierarchy is defined in the Brand Compliance Admin area.

Categories may be assigned to specific product specifications, or available for use by all; if specified, the specification type must be match those assigned to its parent categories.

Where the record is linked to another record, such as the Locale in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

For a successful response, an HTTP 200 response is sent with a body containing a BusinessCategoryLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Business Category record's internal ID
recordLink	string	URI to the newly created Business Category record, for use in a GET request
code	string	Code ID business key to the newly created Business Category record
entityDescription	string	Category's description

Error Responses

If the supplied data does not result in a valid Business Category (such as a missing mandatory field), an HTTP 417 response is sent with an XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
code	Code {0} has already been used	Codes must be unique
parent	The Parent Code cannot be blank when the Top Level Category flag is false	A parent code must be specified unless the category is a top level category
topLevelCategory	The Top Level Category flag cannot be true when the Parent Code has a value	A category cannot be a top level category and have a parent
description	Description must be provided	Descriptions are mandatory
code	The parent Business Category with code {0} cannot be found	The parent category must exist

Element	Message	Meaning
parent	A new Business Category cannot be added to the Business Category with the code{0} because it will fall outside of the Business Category Hierarchy	Categories can only be added within the bounds of the Business Category Hierarchy as defined in Brand Compliance's Admin/System Control/Business Category Configuration
specificationType	{0} is not a valid Specification Type(s)	The specification type or types must be valid
specificationType	The Specification Types must be a subset of the parent Business Category's Specification Types: {0}	This message is issued when no specification types are specified on a child, but there are specification types assigned to the parent It lists the specification types selected on the parent
specificationType	The Business Category Specification Types {0} are not a subset of the parent Business Category Specification Types {1}	This message is issued when the category's specification types have not all be selected on the parent The child's specification types are listed in {0} and the parent's specification types are listed in {1}

Update Record

Description

Updates an existing Business Category record. Use this function to update a category's details in Brand Compliance based on data sourced from the external system, or to move a category to appear beneath a different parent category.

Endpoint address: /services/rest/businessCategory/{id}
HTTP method: PUT

Request Details

The body of the request contains a BusinessCategoryUpdateDTO to specify the updates to the Business Category record.

The request content is similar to that for creating a category, but does not include the child categories link. If children are to be managed, then a separate call is required for each child. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

After the call, the Business Category record is updated to match the request.

Note: When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Business Category record.

The id element is used to locate the category to update.

The hierarchy of category levels is maintained through the parent code, linking a category to its immediate parent. All levels except the top level must therefore have a valid parent code specified.

The number of levels in the hierarchy is defined in the Brand Compliance Admin area.

Categories may be assigned to specific product specifications, or available for use by all; if specified, the specification type must be match those assigned to its parent categories.

Categories may be moved to the same level within another part of the hierarchy. When a category is moved, its children are automatically moved.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Brand Compliance UI.

The body of the request contains a BusinessCategoryUpdateDTO element to specify how the category should appear after the update.

Response Details

If successful, an HTTP 200 response is sent with a body containing a BusinessCategoryLink element. The BusinessCategoryLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Business Category record's internal ID
recordLink	string	URI to the Business Category record, for use in a GET request
code	string	Code ID business key to the Business Category record
entityDescription	string	Category's description
path	string	Category's full path (including parent levels - separated by '/')

Error Responses

If the supplied data does not result in a valid Business Category (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
recordId	Invalid record id	The ID used to call the PUT process does not relate to an existing record
parent	The Parent Code cannot be blank when the Top Level Category flag is false	A parent code must be specified unless the category is a top level category
topLevelCategory	The Top Level Category flag cannot be true when the Parent Code has a value	A category cannot be a top level category and have a parent
code	The parent Business Category with code {0} cannot be found	The parent category must exist

Element	Message	Meaning
code	A new Business Category cannot be added to the Business Category with the code{0} because it will fall outside of the Business Category Hierarchy	Categories can only be added within the bounds of the Business Category Hierarchy as defined in Brand Compliance's Admin/System Control/Business Category Configuration
parent	Business Category cannot be moved from a parent at level {0} to a parent at level {1}	Categories can only be moved between parent categories at the same level in the hierarchy
specificationType	{0} is not a valid Specification Type(s)	The specification type(s) must be valid
specificationType	The Specification Types must be a subset of the parent Business Category's Specification Types: {0}	This message is issued when no specification types are specified on a child but there are specification types assigned to the parent It lists the specification types selected on the parent
specificationType	The Business Category Specification Types {0} are not a subset of the parent Business Category Specification Types {1}	This message is issued when the category's specification types have not all be selected on the parent The child's specification types are listed in {0} and the parent's specification types are listed in {1}

Delete Record

Description

Deletes an existing Business Category record, along with any associated child categories. Use this function to remove a single category or a category and all its lower level categories.

Endpoint address: /services/rest/businessCategory/{id}
HTTP method: DELETE

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to delete.

Response Details

If the Business Category is deleted successfully, an HTTP 200 response is sent.

Error Responses

If a Business Category cannot be deleted due to it being referenced by another record (or a child of the category being referenced by another record), an HTTP 417 response is sent with an ErrorMessage/Message XML body containing the message: "Could not delete: with a URI link to the category."

Error Messages

Element	Message	Meaning
id	HTTP 417 Could not delete: {URI}	Category is referenced by another record
id	HTTP 404 Not found	Invalid {id} - not numeric
id	Invalid record id for deletion	Invalid {id} - not found

AuditRestService

This section describes the API for managing audit and visits. The following functions are available:

- List of Values (Audit): retrieves a list of audits and visits
- Retrieve Record by ID (Audit): retrieves an Audit or Visit record using its unique identifier
- Retrieve Record by Business Key (Audit): retrieves an Audit or Visit record using its business key
- Check Record Modification Timestamp (Audit): retrieves the timestamp when an Audit or Visit record was last updated
- Create Record (Audit): creates a new Audit or Visit record
- Update Record (Audit): updates an existing Audit or Visit record
- List of Values (Checklist): retrieves a list of checklists
- Retrieve Record by ID (Checklist): retrieves a Checklist record using its unique identifier
- Retrieve Record by Business Key (Checklist): retrieves a Checklist record using its business key
- Create Record (Checklist): creates a new Checklist record
- Update Record (Checklist): updates an existing Checklist record
- Delete Record (Checklist): deletes an existing Checklist record

List of Values (Audit)

Description

Retrieves a list of Audits or Visits in a paged list. Use this function to locate Audit or Visit record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/audit
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39
status	Optional	string	Yes	Scheduled~In Progress

Example URL

…/services/rest/audit/?offset=2&pageSize=20

Response Details

For a successful response, XML is returned with an AuditVisitLinkList root element containing an entries element for each matched audit or visit. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Audit/Visit record's internal ID
recordLink	string	URI to the AuditRestService Retrieve record service for this audit/visit
code	string	Audit/Visit's code business key
dueDate	string	Audit/Visit's due date
siteCode	string	Audit/Visit's site code
supplierCode	string	Audit/Visit's supplier code
templateCode	string	Audit/Visit's template code

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID (Audit)

Description

Retrieves a single Audit or Visit record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual audit or visit.

Endpoint address: /services/rest/audit/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/audit/105

Response Details

For a successful response, XML is returned with an AuditFullDTO root element containing the individual attributes of the requested Audit or Visit record. If an ID is not specified, a list of all audits/visits is returned (per the List of Values function).

File Structure:

The main elements of the auditFullDTO root element:

- auditTemplateSnapshot and Template - the settings from the associated template.
- nonConformances - If the audit/visit has any non conformances/issues raised against it, they appear within a nonConformances element - a separate element for each.
- supplier and site - details of the supplier and site that the audit/visit is related to.
- udfData - the contents of any user-defined custom fields associated to the Audit/Visit record. nonConformances may also contain udfData elements, if the Non Conformance record has user-defined custom fields configured.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key (Audit)

Description

Retrieves a single Audit or Visit record's details using its business key (code). Use this function to retrieve the full details of an individual audit or visit using its key code.

Endpoint address: /services/rest/audit/byKey/{code}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {code} parameter that determines the record to retrieve.

Example URL

…/services/rest/audit/byKey/110

Response Details

For a successful response, XML is returned with an AuditFullDTO root element containing the individual attributes of the requested Audit or Visit record.

File Structure:

The main elements of the auditFullDTO root element:

- auditTemplateSnapshot and Template - the settings from the associated template.
- nonConformances - If the audit/visit has any non conformances/issues raised against it, they appear within a nonConformances element - a separate element for each.
- supplier and site - details of the supplier and site that the audit/visit is related to.
- udfData - the contents of any user-defined custom fields associated to the Audit/Visit record. nonConformances may also contain udfData elements, if the Non Conformance record has user-defined custom fields configured.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not Found	Invalid {code} - blank or not found
code	HTTP 404 Not Found	Invalid {code} - not found

Check Record Modification Timestamp (Audit)

Description

Retrieves the last modification date and time of an Audit or Visit record. Use this function to determine when the audit or visit was last updated.

Endpoint address: /services/rest/audit/{id}
HTTP method: HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/audit/105

Response

If successful, an HTTP 200 response is sent containing the Last-Modified header to show the last modification date and time of the last update of the requested Audit or Visit record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not Found	Invalid {id} - not numeric

Create Record (Audit)

Description

Creates a new Audit or Visit record. Use this function to create a new audit or visit in Brand Compliance based on data sourced from the external system.

Dependencies: The Supplier and Site must be present in the application and its record ID obtained. If assigning a Business Category, the ID of the Business Category record must be present in the application and its record ID obtained. If assigning a Product Technologist or People Present, the users must be present in the application and the record IDs of the User record obtained. If non conformances/issues are to be included, and they are associated to a specific user of the application, the record IDs of the User record must be obtained. An audit/visit must be associated to an audit/visit template. For more information, see "Dependencies".

Endpoint address: /services/rest/audit
HTTP method: POST

Request Details

The body of the request contains an AuditFullDTO element to specify the detail of the audit or visit to create, which is based on the AuditFullDTO element returned when retrieving an audit/visit. As a minimum, the fields shown in the following table must be populated.

Audit/Visit Mandatory Fields

Field Name	Element Name
Audit/Visit Template	template / id (or code) auditTemplateSnapshot
Audit/Visit Code	code
Lead Technologist	leadTechnicalManager / code
Supplier	supplier / id
Site	site / id
Due Date	dueDate
Record Type	recordType
Status	status / id
From Date	fromDate
To Date	toDate
People Present	peoplePresent / name

Note: If the status is Scheduled, fromDate, toDate, and peoplePresent are not mandatory.

People Present Table:

This table is used to identify the people present during the audit or visit. The options for completing the table depend on the settings in the audit/visit's template:

- The table must contain at least one entry (whether an audit or a visit).
- If a visit or an internal audit, the entry should be a user of the system who has a Technologist role and is set as an approved auditor. The template may have been set to just allow specific auditors.
- If a third-party audit, the entry should be a Certification Body. The template may have been set to just allow specific auditors.
- Rows of free text may be added to the table.

File Structure:

The main elements of the AuditFullDTO root element are shown in the following table:

Summary of AuditFullDTO Elements

Element Name	Description
announced costChargedToSupplier auditTemplateSnapshot	Values from the audit/visit template, required to control the record's behavior when it is progressed within Brand Compliance.
code	The audit/visit's business key.
dueDate fromDate toDate leadTechnical Manager recordType	Values that may be mandatory for the audit/visit to be created.
nonConformances	Container for the non conformance data - a separate element for each non conformance/issue.
supplier	ID of the supplier that the audit/visit is associated to.
site	ID of the site that the audit/visit is associated to.
status	The audit/visit's status.
template	ID of the audit/visit template that the audit/visit is associated to.

Example Request XML

This example shows the minimum requirement to be able to create an In Progress audit or visit.

```
<ns0:auditFullDTO
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full"
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple">
   <ns0:announced>true</ns0:announced>
```

```
 <ns0:costChargedToSupplier>false</ns0:costChargedToSupplier>
 <ns0:auditTemplateSnapshot>
     <ns0:allowWorkingOffline>false</ns0:allowWorkingOffline>
     <ns0:allowableScoreOptions>
         <ns1:code>AMBER</ns1:code>
     </ns0:allowableScoreOptions>
     <ns0:allowableScoreOptions>
         <ns1:code>GREEN</ns1:code>
     </ns0:allowableScoreOptions>
     <ns0:auditType>INTERNAL</ns0:auditType>
     <ns0:auditVisibility>OPTIONAL</ns0:auditVisibility>
     <ns0:autoSchedule>false</ns0:autoSchedule>
     <ns0:buyerInformationRequired>true</ns0:buyerInformationRequired>
     <ns0:categoriesRequired>2</ns0:categoriesRequired>
     <ns0:categoryRequired>false</ns0:categoryRequired>
     <ns0:certificationBodiesSelectable>ALL</ns0:certificationBodiesSelectable>
     <ns0:closureDeadline>56</ns0:closureDeadline>
     <ns0:closureDeadlineNoOfDays>0</ns0:closureDeadlineNoOfDays>
         <ns0:commentsNotVisibleToOrganisationRequired>true
     </ns0:commentsNotVisibleToOrganisationRequired>
     <ns0:costChargedToSupplier>false</ns0:costChargedToSupplier>
     <ns0:defaultFrequency>12</ns0:defaultFrequency>
     <ns0:defaultIssueCompletion>28</ns0:defaultIssueCompletion>
     <ns0:extendedDeadline>56</ns0:extendedDeadline>
     <ns0:extendedDeadlineNoOfDays>0</ns0:extendedDeadlineNoOfDays>
     <ns0:internalAuditorsSelectable>ALL</ns0:internalAuditorsSelectable>
     <ns0:isActive>true</ns0:isActive>
     <ns0:localeData>
         <ns0:name>Checklist Audit</ns0:name>
         <ns0:id>49</ns0:id>
     </ns0:localeData>
     <ns0:mayExternalManagersEditConfig>MAY_NOT_EDIT
     </ns0:mayExternalManagersEditConfig>
     <ns0:nonConformanceCreationMethod>VIA_CHECKLISTS
     </ns0:nonConformanceCreationMethod>
     <ns0:organisationType>SITE</ns0:organisationType>
     <ns0:recordType>AUDIT</ns0:recordType>
     <ns0:scoreAudit>true</ns0:scoreAudit>
     <ns0:scoreRequired>true</ns0:scoreRequired>
     <ns0:stampOrgWithLastAuditDate>true</ns0:stampOrgWithLastAuditDate>
     <ns0:supplierGenerated>false</ns0:supplierGenerated>
     <ns0:useAuditVisitStandards>false</ns0:useAuditVisitStandards>
     <ns0:id>6</ns0:id>
</ns0:auditTemplateSnapshot>
<ns0:dueDate>2017-04-27</ns0:dueDate>
<ns0:fromDate>2017-04-27</ns0:fromDate>
<ns0:leadTechnicalManager>
    <ns1:code>PROD-TECH</ns1:code>
    <ns1:disabled>false</ns1:disabled>
    <ns1:externalAuthenticationUser>false</ns1:externalAuthenticationUser>
    <ns1:timeZone>Europe/London</ns1:timeZone>
    <ns1:userType>RETAILER</ns1:userType>
    <ns1:id>12</ns1:id>
</ns0:leadTechnicalManager>
<ns0:nonConformances>
    <ns0:description>&lt;p>Test&lt;/p></ns0:description>
    <ns0:reference>AR01</ns0:reference>
    <ns0:status>
        <ns1:status>OPEN</ns1:status>
    </ns0:status>
```

```
      <ns0:type>
          <ns1:code>MAJOR</ns1:code>
      </ns0:type>
      <ns0:code>AR01-11</ns0:code>
   </ns0:nonConformances>
   <ns0:recordType>AUDIT</ns0:recordType>
   <ns0:site>
       <ns1:id>27</ns1:id>
   </ns0:site>
   <ns0:status>
       <ns1:status>IN PROGRESS</ns1:status>
   </ns0:status>
   <ns0:supplier>
       <ns1:id>21</ns1:id>
   </ns0:supplier>
   <ns0:template>
       <ns1:id>8</ns1:id>
   </ns0:template>
   <ns0:toDate>2017-04-27</ns0:toDate>
   <ns0:code>AR1</ns0:code>
</ns0:auditFullDTO>
```

Where the record is linked to another record, such as the Template, the business key or ID must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record. The supplier and site elements can contain just the ID of the respective record links.

A unique code for the audit/visit must be assigned.

Templates:

The audit/visit is linked to its template using the template and auditTemplateSnapshot elements. The template element can contain just the code or ID of the associated Template record.

The auditTemplateSnapshot element must be completed to provide the parameters that control the audit/visits' behavior as it progresses through its Brand Compliance workflow. If scoring is used, the allowable options can be specified using the code or ID of the glossary record; if none are specified, all options will be available when the user selects the score.

Non Conformances:

If the audit/visit is to be created with non conformances/issues already raised against it, they can be added within a nonConformances element - a separate element for each. A unique code must be assigned to each. The status can be set using its code or ID.

For further information about the process for creating audit, non conformances and checklists, see "Dependencies."

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a AuditVisitLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Audit/Visit record's internal ID
recordLink	string	URI to the newly created Audit/Visit record, for use in a GET request
code	string	Audit/Visit's code business key
dueDate	string	Audit/Visit's due date
siteCode	string	Audit/Visit's site code
supplierCode	string	Audit/Visit's supplier code
templateCode	string	Audit/Visit's template code

Error Responses

If the supplied data does not result in a valid Audit or Visit (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
peoplePresent	WARNING: class com.micros.creations.core.domain.Audit.peoplePresent - There must be at least one entry in the People Present table	People Present not provided
recordType	IllegalArgumentException: No enum constant com.micros.creations.core.type.AuditRecordType. <###>	Record Type not found
supplier/id code site/id code supplierId supplierCode	IllegalStateException: Cannot locate keyword for class com.micros.creations.core.domain.Site using id: <###>, code:<###>, supplierCode:<###> and supplierId:<###>	Site Id not found Site Code not found Supplier Id not found Supplier Code not found
template	EntityNotFoundException: Unable to find com.micros.creations.core.domain.AuditTemplate with id <###>	Template not found
fromDate	ERROR: class com.micros.creations.core.domain.Audit.fromDate - The condition is invalid	From Date not provided
code	ERROR: class com.micros.creations.core.domain.Audit.code - The condition is invalid	Audit/Visit code not provided
code	ERROR: class com.micros.creations.core.domain.Audit.code - Another object exists with this value	Audit/Visit code already exists

Update Record (Audit)

Description

Updates an existing Audit or Visit record. Use this function to update an audit or visit's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/audit/{id}
HTTP method: PUT

Request Details

The body of the request contains an AuditFullDTO element to specify the updates to the Audit or Visit record, which is based on the AuditFullDTO type returned when retrieving an audit/visit.

> **Note:** The request content is similar to that for creating an audit or visit.
>
> As a minimum, the values specified as mandatory for the Create Record function (see above) must be included. However, when updating records, all values must be included.
>
> If a value or element is omitted from the request, the field contents will be cleared on the Audit/Visit record.

If the audit/visit has any non conformances/issues raised against it, they appear within a nonConformances element - a separate element for each. The non conformances/issues can be updated or created when updating the audit/visit. If creating a new Non Conformance record, assign a unique code and omit the id element.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an AuditVisitLink element. The AuditVisitLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Audit/Visit record's internal ID
recordLink	string	URI to the Audit/Visit record, for use in a GET request
code	string	Audit/Visit's code business key
dueDate	string	Audit/Visit's due date
siteCode	string	Audit/Visit's site code
supplierCode	string	Audit/Visit's supplier code
templateCode	string	Audit/Visit's template code

Error Responses

If the supplied data does not result in a valid Audit or Visit (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

List of Values (Checklist)

Description

Retrieves a list of Checklists in a paged list. Use this function to locate Checklist record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/audit/{auditId}/checklist
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
auditId	Optional	long	No	105
offset	Optional	int	No	60
pageSize	Optional	int	No	30

Example URL

.../services/rest/audit/105/checklist/?offset=2&pageSize=20

Response Details

For a successful response, XML is returned with an AuditChecklistLinkList root element containing an entries element for each matched audit or visit. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Checklist record's internal ID
recordLink	string	URI to the AuditRestService Retrieve record service for this checklist
parentId	string	The parent Audit/Visit's internal ID
auditCode	string	The parent Audit/Visit's code business key
globalId	string	Checklist's code business key

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID (Checklist)

Description

Retrieves a single Checklist record's details using the record's internal unique ID (which is not visible in the UI) and its parent audit/visit's internal unique Id. Use this function to retrieve the full details of an individual audit or visit.

Endpoint address: /services/rest/audit/{auditId}/checklist/{id}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {auditId} parameter that identifies the parent Audit/Visit record, and the {id} parameter that determines the record to retrieve.

Example URL

…/services/rest/audit/105/checklist/7

Response Details

For a successful response, XML is returned with an auditChecklistFullDTO root element containing the individual attributes of the requested Checklist record. If a Checklist ID is not specified, a list of all the audit/visit's checklists is returned (as per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
auditId	HTTP 404 Not found	Invalid {auditId} - not numeric
auditId	IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {auditId} - not found
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key (Checklist)

Description

Retrieves a single Checklist record's details using its business key (code) and its parent audit/visit's code. Use this function to retrieve the full details of an individual checklist using its key code.

Endpoint address:
/services/rest/audit/byKey/{auditCode}/checklist/{checklistCode}
HTTP method: GET

Request Details

There are no request parameters, but the URL contains the {auditCode} parameter that identifies the parent Audit/Visit record, and the {checklistCode} parameter that determines the record to retrieve.

Example URL

…/services/rest/audit/byKey/110/checklist/7374be1e-2a6c-11e7-8fda-0021f60dd918

Response Details

For a successful response, XML is returned with an auditChecklistFullDTO root element containing the individual attributes of the requested Audit or Visit record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
auditCode	HTTP 404 Not Found	Invalid {auditCode} - blank or not found
checklistCode	HTTP 404 Not Found	Invalid {checklistCode} - blank or not found

Create Record (Checklist)

Description

Creates a new Checklist record. Use this function to create a new checklist in Brand Compliance based on data sourced from the external system.

Dependencies: To create an audit with a checklist, the parent Audit record must be created using the Create Record (Audit/Visit) function, before creating the associated Checklist records.

Therefore, when creating the checklist, the parent Audit/Visit record must be present in the application and its record ID obtained. A checklist's questions and the permitted answers are determined by the checklist template that has been assigned to the audit/visit's template. The IDs of the Checklist Template record must also be provided when creating a template. For more information, see "Dependencies".

Endpoint address: /services/rest/audit{auditId}/checklist
HTTP method: POST

Request Details

The body of the request contains an auditChecklistFullDTO element to specify the detail of the checklist to create, which is based on the auditChecklistFullDTO element returned when retrieving a checklist. As a minimum, the fields shown in the following table must be populated.

Checklist Mandatory Fields

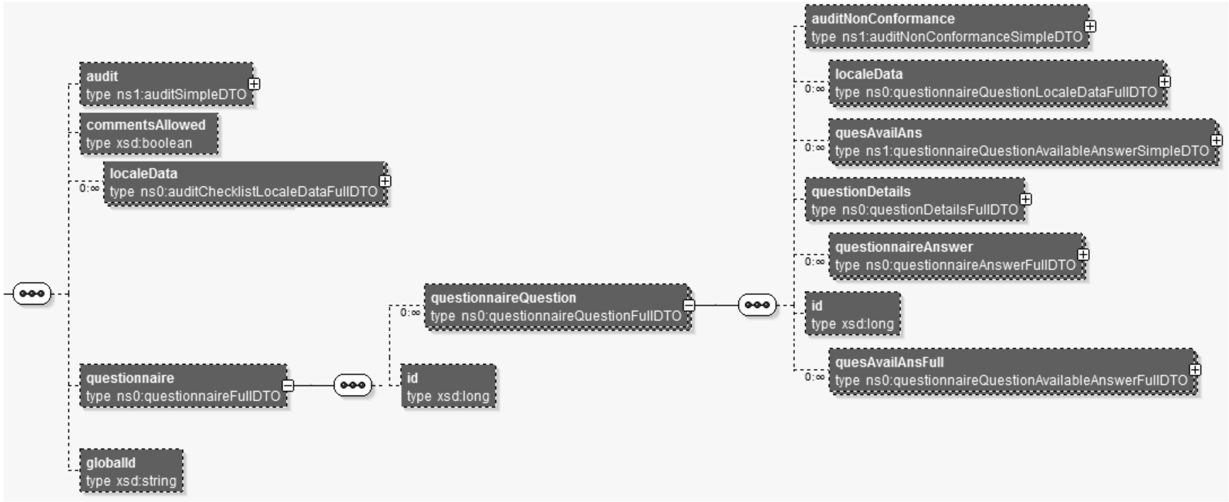
Field Name	Element Name
Audit/Visit Id	audit / id See note 1.
Audit/Visit Code	audit /code See note 1.
Checklist Template	localeData / id
Checklist Comments Allowed	commentsAllowed
Questionnaire Questions	questionnaire / questionnaireQuestion / questionDetails
Question Answers	questionnaire / questionnaireQuestion / questionnaireAnswer questionnaire / questionnaireQuestion / quesAvailAnsFull

Notes:

1. Either code or ID may be specified to identify the audit/visit.

File Structure:

The following diagram shows the structure of the auditChecklistFullDTO request file structure:


audit
type ns1:auditSimpleDTO
commentsAllowed
type xsd:boolean
0:∞
localeData
type ns0:auditChecklistLocaleDataFullDTO
questionnaire
type ns0:questionnaireFullDTO
globalId
type xsd:string
0:∞
questionnaireQuestion
type ns0:questionnaireQuestionFullDTO
id
type xsd:long
auditNonConformance
type ns1:auditNonConformanceSimpleDTO
0:∞
localeData
type ns0:questionnaireQuestionLocaleDataFullDTO
0:∞
quesAvailAns
type ns1:questionnaireQuestionAvailableAnswerSimpleDTO
questionDetails
type ns0:questionDetailsFullDTO
0:∞
questionnaireAnswer
type ns0:questionnaireAnswerFullDTO
id
type xsd:long
0:∞
quesAvailAnsFull
type ns0:questionnaireQuestionAvailableAnswerFullDTO

Summary of AuditChecklistFullDTO Elements

Element Name	Description
audit	Identifies the parent audit record by code or id.
commentsAllowed	A value from the checklist template, required to control the record's behavior when it is progressed within Brand Compliance.
localeName	Identifies the checklist template by name and id.
questionnaire	Groups the questions and their answers.
questionnaireQuestion	Groups the checklist questions and their answers - repeated for each question in the checklist. reference is the question reference.

Element Name	Description
questionDetails	Individual questions. question is the question text. answerComment is the answer's comments (if configured). answerFieldType is code of answer type. commentAllowed is whether comments are allowed.
questionnaireAnswer	Groups the answer details.
questionnaireAvailableAnswer	Answers for the individual questions. code is the question's unique key. commentMandatory is whether comments are mandatory or not. generateNonConformance is whether an answer will generate an issue.
quesAvailAnsFull	The available answers - repeated for each answer. code is the question's unique key. commentMandatory is whether comments are mandatory or not. id and keywordTypeEnum are the keys to the Issues glossary.

Process for Creating Checklist Audits

The general process for creating checklist audits:

1. Create the Audit/Visit. While it is possible to create a checklist audit using the Create Audit function, it is recommended that they be created within the Brand Compliance application.

   A method of creating a new audit:

   a. Retrieve an existing audit using a GET function.

   b. Omit the id and statusChangeHistory elements from the retrieved file.

   c. Omit localData elements from status and allowableAuditStandards.

   d. Assign a new unique code.

   e. Link to the Supplier, Site, and Audit Template records using the relevant IDs.

   f. Change other content to reflect the new audit.

   g. Submit using the POST function.

   h. If non conformances are to be created at the time of creating the audit/visit, include them as nonConformances elements.

   > **Note:** The API does not automatically create a non conformance based on the answer of a checklist question. Non conformances must be assigned when either creating or updating the Audit/Visit record.

2. Retrieve the code or ID of the newly created Audit, and its template detail using a GET function.

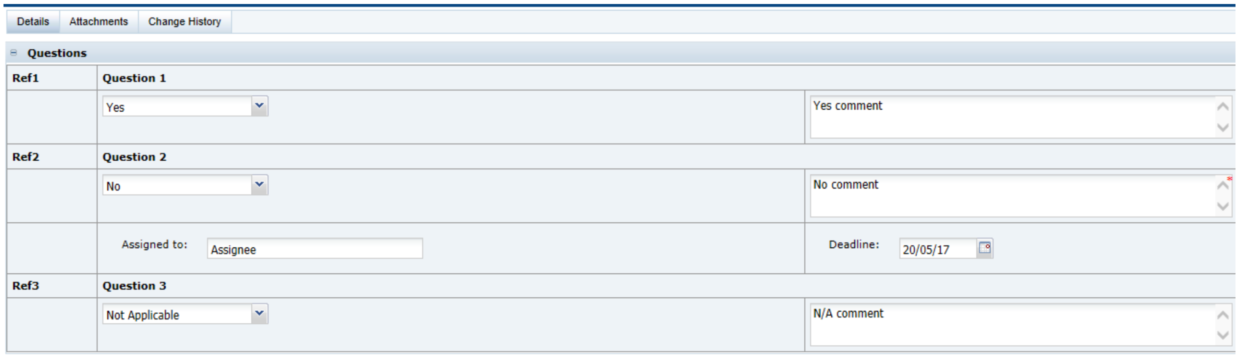
3. Create the checklist using the POST function (repeat for each checklist):

   - Identify the parent audit/visit by setting the code or id within audit element.

- Can retrieve an existing checklist using a GET function, in which case the id and questionnaireQuestionAvailableAnswer elements must be omitted. Also omit any id elements within the questionnaire element, except those that relate to the auditNonConformance element.

Example Request XML

This example shows the minimum requirement to be able to create an audit checklist. This example contains three questions, each with the available answers of Yes, No, or Not Applicable.



Ref1	Question 1	
	Yes	Yes comment
Ref2	Question 2	
	No	No comment
	Assigned to: Assignee	Deadline: 20/05/17
Ref3	Question 3	
	Not Applicable	N/A comment

```
<ns0:auditChecklistFullDTO
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
    <ns0:audit>
        <ns1:code>AUD09</ns1:code>
        <ns1:id>132</ns1:id>
    </ns0:audit>
    <ns0:commentsAllowed>false</ns0:commentsAllowed>
    <ns0:localeData>
        <ns0:name>
            <![CDATA[ API Test Template ]]>
        </ns0:name>
        <ns0:id>563</ns0:id>
    </ns0:localeData>
    <ns0:questionnaire>
        <ns0:questionnaireQuestion>
            <ns0:localeData>
                <ns0:reference>
                    <![CDATA[Ref1]]>
                </ns0:reference>
            </ns0:localeData>
            <ns0:questionDetails>
                <ns0:answerComment>
                    <![CDATA[Yes comment]]>
                </ns0:answerComment>
                <ns0:answerFieldType>SINGLE_SELECTION</ns0:answerFieldType>
                <ns0:commentAllowed>true</ns0:commentAllowed>
                <ns0:localeData>
                    <ns0:question>
                        <![CDATA[Question 1]]>
                    </ns0:question>
                </ns0:localeData>
            </ns0:questionDetails>
            <ns0:questionnaireAnswer>
                <ns0:questionnaireAvailableAnswer>
                    <ns1:code>QQAA7236</ns1:code>
                    <ns1:commentMandatory>false</ns1:commentMandatory>
                    <ns1:generateNonConformance>false
                    </ns1:generateNonConformance>
                    <ns1:localeData>
                        <ns0:answer>Yes</ns0:answer>
                    </ns1:localeData>
                </ns0:questionnaireAvailableAnswer>
                <ns0:questionnaireQuestion>
                    <ns1:localeData>
                        <ns0:reference>
                            <![CDATA[Ref1]]>
                        </ns0:reference>
                    </ns1:localeData>
                </ns0:questionnaireQuestion>
            </ns0:questionnaireAnswer>
            <ns0:quesAvailAnsFull>
                <ns0:code>QQAA7236</ns0:code>
                <ns0:commentMandatory>false</ns0:commentMandatory>
                <ns0:generateNonConformance>false</ns0:generateNonConformance>
                <ns0:localeData>
                    <ns0:answer>Yes</ns0:answer>
                </ns0:localeData>
            </ns0:quesAvailAnsFull>
            <ns0:quesAvailAnsFull>
                <ns0:code>QQAA7237</ns0:code>
                <ns0:commentMandatory>true</ns0:commentMandatory>
                <ns0:generateNonConformance>true</ns0:generateNonConformance>
                <ns0:localeData>
                    <ns0:answer>No</ns0:answer>
                </ns0:localeData>
            </ns0:quesAvailAnsFull>
            <ns0:quesAvailAnsFull>
                <ns0:code>QQAA7238</ns0:code>
                <ns0:commentMandatory>false</ns0:commentMandatory>
                <ns0:generateNonConformance>false</ns0:generateNonConformance>
                <ns0:localeData>
                    <ns0:answer>Not Applicable</ns0:answer>
                </ns0:localeData>
            </ns0:quesAvailAnsFull>
        </ns0:questionnaireQuestion>
        <ns0:questionnaireQuestion>
            <ns0:localeData>
                <ns0:reference>
                    <![CDATA[Ref2]]>
                </ns0:reference>
            </ns0:localeData>
            <ns0:questionDetails>
                <ns0:answerComment>
                    <![CDATA[No comment]]>
                </ns0:answerComment>
                <ns0:answerFieldType>SINGLE_SELECTION</ns0:answerFieldType>
                <ns0:commentAllowed>true</ns0:commentAllowed>
                <ns0:localeData>
                    <ns0:question>
                        <![CDATA[Question 2]]>
                    </ns0:question>
                </ns0:localeData>
            </ns0:questionDetails>
            <ns0:questionnaireAnswer>
                <ns0:questionnaireAvailableAnswer>
```

```
                <ns1:code>QQAA7240</ns1:code>
                <ns1:commentMandatory>true</ns1:commentMandatory>
                <ns1:generateNonConformance>true</ns1:generateNonConformance>
            <ns1:localeData>
                <ns0:answer>No</ns0:answer>
            </ns1:localeData>
        </ns0:questionnaireAvailableAnswer>
        <ns0:questionnaireQuestion>
            <ns1:localeData>
                <ns0:reference>
                    <![CDATA[Ref2]]>
                </ns0:reference>
            </ns1:localeData>
        </ns0:questionnaireQuestion>
    </ns0:questionnaireAnswer>
    <ns0:quesAvailAnsFull>
        <ns0:code>QQAA7239</ns0:code>
        <ns0:commentMandatory>false</ns0:commentMandatory>
        <ns0:generateNonConformance>false</ns0:generateNonConformance>
        <ns0:localeData>
            <ns0:answer>Yes</ns0:answer>
        </ns0:localeData>
    </ns0:quesAvailAnsFull>
    <ns0:quesAvailAnsFull>
        <ns0:code>QQAA7240</ns0:code>
        <ns0:commentMandatory>true</ns0:commentMandatory>
        <ns0:generateNonConformance>true</ns0:generateNonConformance>
        <ns0:localeData>
            <ns0:answer>No</ns0:answer>
        </ns0:localeData>
    </ns0:quesAvailAnsFull>
    <ns0:quesAvailAnsFull>
        <ns0:code>QQAA7241</ns0:code>
        <ns0:commentMandatory>false</ns0:commentMandatory>
        <ns0:generateNonConformance>false</ns0:generateNonConformance>
        <ns0:localeData>
            <ns0:answer>Not Applicable</ns0:answer>
        </ns0:localeData>
    </ns0:quesAvailAnsFull>
</ns0:questionnaireQuestion>
<ns0:questionnaireQuestion>
    <ns0:localeData>
        <ns0:reference>
            <![CDATA[Ref3]]>
        </ns0:reference>
    </ns0:localeData>
    <ns0:questionDetails>
        <ns0:answerComment>
            <![CDATA[N/A comment]]>
        </ns0:answerComment>
        <ns0:answerFieldType>SINGLE_SELECTION</ns0:answerFieldType>
        <ns0:commentAllowed>true</ns0:commentAllowed>
        <ns0:localeData>
            <ns0:question>
                <![CDATA[Question 3]]>
            </ns0:question>
        </ns0:localeData>
    </ns0:questionDetails>
    <ns0:questionnaireAnswer>
        <ns0:questionnaireAvailableAnswer>
```

```
                <ns1:code>QQAA7244</ns1:code>
                <ns1:commentMandatory>false</ns1:commentMandatory>
                <ns1:generateNonConformance>false</ns1:generateNonConformance>
                <ns1:localeData>
                    <ns0:answer>Not Applicable</ns0:answer>
                </ns1:localeData>
             </ns0:questionnaireAvailableAnswer>
             <ns0:questionnaireQuestion>
                 <ns1:localeData>
                     <ns0:reference>
                         <![CDATA[Ref3]]>
                     </ns0:reference>
                 </ns1:localeData>
             </ns0:questionnaireQuestion>
         </ns0:questionnaireAnswer>
         <ns0:quesAvailAnsFull>
             <ns0:code>QQAA7242</ns0:code>
             <ns0:commentMandatory>false</ns0:commentMandatory>
             <ns0:generateNonConformance>false</ns0:generateNonConformance>
             <ns0:localeData>
                 <ns0:answer>Yes</ns0:answer>
             </ns0:localeData>
         </ns0:quesAvailAnsFull>
         <ns0:quesAvailAnsFull>
             <ns0:code>QQAA7243</ns0:code>
             <ns0:commentMandatory>true</ns0:commentMandatory>
             <ns0:generateNonConformance>true</ns0:generateNonConformance>
             <ns0:localeData>
                 <ns0:answer>No</ns0:answer>
             </ns0:localeData>
         </ns0:quesAvailAnsFull>
         <ns0:quesAvailAnsFull>
             <ns0:code>QQAA7244</ns0:code>
             <ns0:commentMandatory>false</ns0:commentMandatory>
             <ns0:generateNonConformance>false</ns0:generateNonConformance>
             <ns0:localeData>
                 <ns0:answer>Not Applicable</ns0:answer>
             </ns0:localeData>
         </ns0:quesAvailAnsFull>
     </ns0:questionnaireQuestion>
</ns0:questionnaire>
<ns0:globalId>7470e250-36cc-4dea-a0f1-eb6040185865</ns0:globalId>
</ns0:auditChecklistFullDTO>
```

For further information about the process for creating audit, non conformances, and checklists, see "Dependencies."

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an AuditChecklistLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Checklist record's internal ID
recordLink	string	URI to the newly created Checklist record, for use in a GET request
parentId	long	The parent Audit/Visit's internal ID
auditCode	string	The parent Audit/Visit's code business key
globalId	string	Checklist's code business key

Error Responses

If the supplied data does not result in a valid Checklist (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
audit / id audit / code	IllegalStateException: The checklists audit cannot be located	Audit/Visit id/code not found
id	IllegalStateException: Cannot find matching relation to update on <###> looking for <###>	Invalid link to related record

Update Record (Checklist)

Description

Updates an existing Checklist record. Use this function to update a checklist's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/rest/audit/{auditId}/checklist/{id}
HTTP method: PUT

Request Details

The body of the request contains an auditChecklistFullDTO element to specify the updates to the Audit or Visit record, which is based on the auditChecklistFullDTO type which is returned when retrieving an audit/visit.

> **Note:** The request content is similar to that for creating a checklist.
>
> As a minimum, the values specified as mandatory for the Create Record function (see above) must be included. However, when updating records, all values must be included.
>
> If a value or element is omitted from the request, the field contents will be cleared on the Checklist record.

The link to the parent Audit/Visit record is defined by the code or ID within the audit element.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an auditChecklistLink element. The auditChecklistLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Checklist record's internal ID
recordLink	string	URI to the Checklist record, for use in a GET request
parentId	long	The parent Audit/Visit's internal ID
auditCode	string	The parent Audit/Visit's code business key
globalId	string	Checklist's code business key

Error Messages

If the supplied data does not result in a valid Audit or Visit (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Delete Record (Checklist)

Deletes an existing Checklist record. Use this function to delete a checklist that has been created in error from the external system

Endpoint address: /services/rest/audit/{auditId}/checklist/{id}
HTTP method: DELETE

Request Details

There are no request parameters, but the URL contains the {auditId} parameter that identifies the parent Audit/Visit record, and the {id} parameter that determines the record to delete.

> **Note:** It is not possible to manually delete checklists from an audit/visit within the application; this delete function has been provided solely for the deletion of checklists that have been created in error by the API.

Example URL

…/services/rest/audit/105/checklist/7

Response Details

If successful, an HTTP 200 response is sent.

Error Messages

Element	Message	Meaning
auditId	HTTP 404 Not Found	Invalid {auditId} - not numeric
auditId	IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {auditId} - not found
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	IllegalStateException: Cannot find the Audit Checklist to return with id:<###>	Invalid {id} - not found

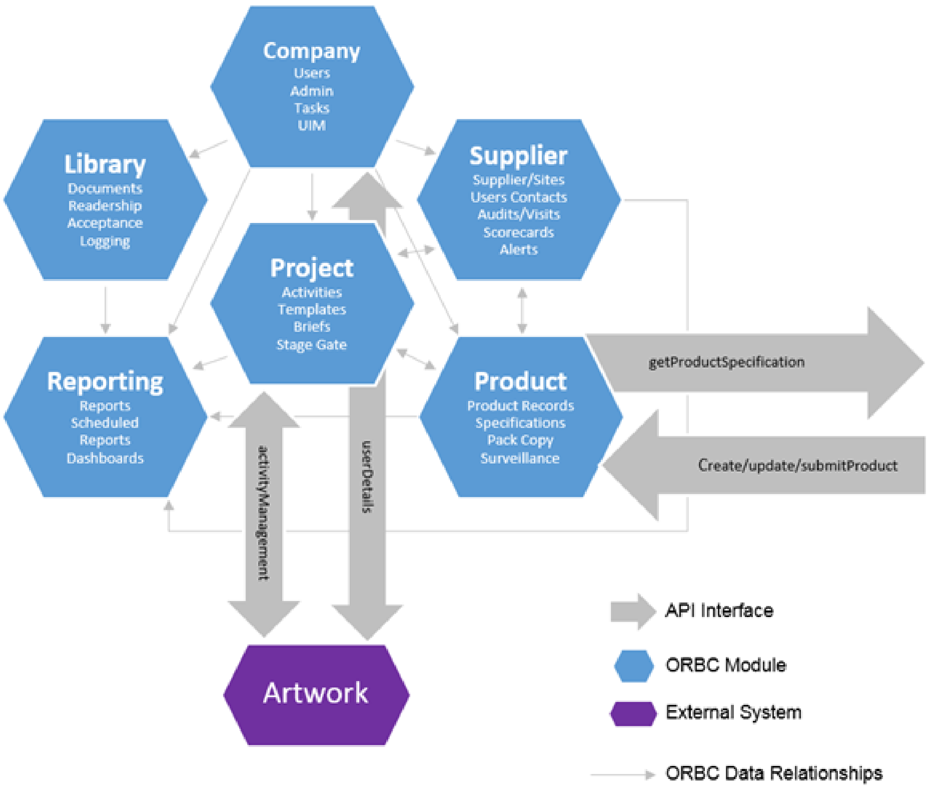
5

SOAP APIs

The following SOAP APIs are available:

- User Details: Users
- Product Service: Product Records
- Retrieve Product Specifications: Product Specifications
- Project Activities: Project Activities

Figure 5–1 Overview of SOAP APIs



Parameters and Filtering

Various parameters can be included in calls to the APIs, generally to define what data is to be returned, but can also control how records are returned. The following table lists some commonly used parameters.

Common Parameters

Parameter	Type	Description
offset	int	Used with pageSize to control the paging of a returned list of records. Specifies the starting point for the retrieval of records. If not specified, zero is assumed. For example, to retrieve 150 records: 1. A call with offset = 0 and pageSize = 50 returns the first 50 records. 2. Then, a call with offset = 50 and pageSize = 50 returns the next 50. 3. Then, a call with offset = 100 and pageSize = 50 returns the final 50.
batchSize	int	Used with offset to control the paging of a returned list of records. Specifies the number of entries in each page of returned list of records. If not specified, 30 is assumed. The maximum is 100.

Note: Date/Time parameters must be provided in the YYYY-MM-DD hh:mm:ss format.

When retrieving records, the returned XML only includes elements that actually contain data; empty elements are omitted.

The following example provides the first 50 Product Specifications that changed status to either Active or Off Range in May 2016. Searching with either fromDate or toDate specified without specStatus, will filter based on the last amended date of the Product Specification.

```
<soapenv:Envelope xmlns:soapenv="<<hostname>>/soap/envelope/"
xmlns:v1="<<servicename>>/">
<soapenv:Header/>
<soapenv:Body>
       <v1:getProductSpecification>
              <request>
                     <batchsize>50</batchsize>
                     <offset>0</offset>
                     <specStatus>ACTIVE</specStatus>
                     <specStatus>OFF_RANGE</specStatus>
                     <fromDate>2016-05-01T00:00:00</fromDate>
                     <toDate>2016-05-31T24:00:00</toDate>
              </request>
       </v1:getProductSpecification>
</soapenv:Body>
```

User Details

This section describes the API for managing single sign-on between Brand Compliance and the JasperReports Server used by the Reports module. The following function is available:

- userDetails: retrieves user information for the user currently logged into Brand Compliance

userDetails

Description

This API is used in the Reports Single Sign-On integration with the TIBCO JasperSoft JasperReports Server to allow the JasperReports Server to retrieve the user information for the user that is logged into Brand Compliance. A oneTimePad is given to the JasperReports Server for use in calling this API. For each oneTimePad, a single call is permitted. Attempting to use the same oneTimePad multiple times results in failures.

This API is used in the Reporting Single Sign-On integration. The oneTimePad is automatically generated when the user clicks the Reports link; it is not feasible to manually call this service.

The API can be tested by clicking the navigation link to the Reports module.

> **Note:** This service is superseded by the SSO Restful Service.

Endpoint address: /services/userDetails
HTTP method: getUser

Request Details

The request contains a single value, oneTimePad, which is the identifier passed to the JasperReports Server when the Reports tab is opened.

Response Details

If the oneTimePad value matches a value stored in the application, the details of the currently logged in user are returned to the caller.

Error Messages

Element	Message	Meaning
oneTimePad	Value not recognized: <###>.	The oneTimePad value is not found or not specified
oneTimePad	userId cannot be null	The user is not found

Product Service

This section describes the API managing Product Records. The following functions are available:

- createProduct: creates a new Product Record
- updateProduct: update an existing Product Record
- submitProduct: automatically detects whether a submitted Product Record is to be updated or created based on whether it already exists or not

> **Note:** The SOAP version of the Product Record API is only compatible with portals that are operating in single variant mode, where the Product Record represents just a single variant of the product.

createProduct

Description

Creates a new Product Record. Use this function to create new Product Records in Brand Compliance based on the data sourced from, the external system.

Dependencies: If linked to a supplier or site, the Supplier/Site must be present in the application and its record ID obtained. If assigning a Business Category, Product Technologist, or Other Contact the record must be present in the application and its record ID obtained. Product Technologist and Other Contacts may be omitted to default to TBC if the TBC user is present in the application. The status must be Active in order to be linked to a specification; for the status to be Active the Product Record must be linked to a supplier. For more information, see "Dependencies".

Endpoint address: /services/productService
HTTP method: createProduct

Request Details

The body of the request contains a createProductRequest element to specify the details of the Product Record to create. Only the attributes that are to be populated on the created Product Record need to be included. As a minimum, this must include the fields shown in the following table.

Product Record Mandatory Fields

Field Name	Element Name
Product Title	ProductName
Status	Status
Variant Name	ProductName
Quantity	Quantity
Specification Type	SpecificationType - see below
Specification Type Format	SpecificationTypeFormat - see below
Technologist	ProductTechnologist - see below
Other Contacts	OtherContacts - see below
Supplier Details	supplier
Site Details	site

Example Request XML

This example shows the minimum requirement to be able to create a Product Record.

```
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ext="external.service.creations.micros.com"
xmlns:sch="http://www.micros.com/schema">
   <soapenv:Header/>
   <soapenv:Body>
      <ext:createProductRequest>
         <product>
            <sch:ProductNumber>ABC124</sch:ProductNumber>
            <sch:ProductName>AR Product</sch:ProductName>
            <sch:SupplierCode>A0001</sch:SupplierCode>
            <sch:Status>DRAFT</sch:Status>
            <sch:ProductTechnologist>TBC</sch:ProductTechnologist>
            <sch:OtherContacts>
               <sch:Contact>
                  <sch:Name>TBC</sch:Name>
                  <sch:Role>BUYER</sch:Role>
               </sch:Contact>
               <sch:Contact>
                  <sch:Name>TBC</sch:Name>
                  <sch:Role>PRODUCT DEVELOPMENT MANAGER</sch:Role>
               </sch:Contact>
            </sch:OtherContacts>
            <sch:Quantity>100g</sch:Quantity>
         </product>
      </ext:createProductRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

If a specification type has more than one format, such as a Food specification having separate formats for Food - UK and Food - US, the SpecificationTypeFormat element is used to specify the specification format and SpecificationType is used to identify the specification type, for example:

```
<sch:SpecificationType>FOOD</sch:SpecificationType>
<sch:SpecificationTypeFormat>FOODUK</sch:SpecificationTypeFormat>
```

The code of the specification type/format must be used rather than the description.

ProductName populates the product name/title and the variant name.

ProductNumber populates the retailer's product code, such as SKU in the Product Coverage table.

ProductBarcode and OuterBarcode populate the EAN Barcodes and Shipping Case Code barcodes on the Product Record, and are cascaded to the Product Specification when it is linked to the Product Record.

ProductTechnologist and OtherContacts populate the retailer contacts. If not specified, TBC will be defaulted (if the necessary User record has been configured).

Depending on the setup of Brand Compliance, there can be zero to many additional roles for which users must be nominated on a Product Record. For each Role of users, a Contact sub-element is added to OtherContacts which maps to the application's Product Record based on the Role element's value matching the Code of the user role in the application. The users are then loaded using the name supplied. For example, if the system has been set up to include the Buyer role in the Product Record's list of roles:

```
<sch:OtherContacts>
    <sch:Contact>
        <sch:Name>John Smith</sch:Name>
        <sch:Role>BUYER</sch:Role>
    </sch:Contact>
</sch:OtherContacts>
```

See the associated WSDL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a createProductResponse root element, consisting of a Status element populated with SUCCEEDED.

If an error occurs, the createProductResponse root element contains the elements shown in the following table.

Returned Elements

Element	Type	Description
Status	string	Contains FAILED to indicate an error
Message/Message/ Severity	string	Contains ERROR to indicate an error
Message/Message/ Description	string	The error message, for example: Specification Type FOOD does not map to a single Specification Type Format in Brand Compliance, product not added

Note: Even if the request succeeds, additional warning messages may be included. Check the Status element for SUCCEEDED or FAILED to determine the success or failure of the call.

Errors are returned with a Severity element set to either ERROR (action failed) or WARNING (action completed with the invalid data omitted) and a Description element containing the error message.

Error Messages

Element	Message	Meaning
	Error: Contact TBC not found in Brand Compliance Management Cloud Service, product not added.	User TBC not found; required as the default for contacts if no valid user is specified
SpecificationType	Error: Specification Type <###> does not map to a single Specification Type Format in Brand Compliance Management Cloud Service, product not added.	Specification type not found
SpecificationTypeFormat	Error: Specification Type Format <###> does not exist in Brand Compliance Management Cloud Service, product not added.	Specification format not found
ProductNumber	Error: Product Number <###> already exists in Brand Compliance Management Cloud Service, product not added.	Product Record already exists for the product number and supplier combination

Element	Message	Meaning
SupplierCode	Error: Supplier <###> is not found in Brand Compliance Management Cloud Service, product with product number null not added.	Supplier not found
SiteCode	Error: Site with code <###> not found in Brand Compliance Management Cloud Service, product not added.	Site not found
ProductBarcode	Warning: EAN barcode <###> is invalid.	Invalid barcode format - ignored
OuterBarcode	Warning: Shipping Case code <###> is invalid.	Invalid barcode format - ignored
ProductTechnologist	Error: Contact <###> not found in Brand Compliance Management Cloud Service.	Product Technologist not found
ProductTechnologist	Error: Contact <###> does not have PRODUCT TECHNOLOGIST role, product not added.	User does not have the Product Technologist role
OtherContacts	Error: <###> role is not a valid product record contact role, product not added.	Role not found
OtherContacts	Warning: Contact <###> not found in Brand Compliance Management Cloud Service.	User not found - ignored
LeadBusinessCategory BusinessCategories	Warning: Business Category <###> not found in Brand Compliance Management Cloud Service.	Business category not found - ignored
CountryWhereSold	Warning: Country Where Sold with code <###> not found in Brand Compliance Management Cloud Service.	Business category not found - ignored

updateProduct

Description

Updates an existing Product Record. Use this function to update a Product Record's details in Brand Compliance based on data sourced from the external system.

Endpoint address: /services/productService
HTTP method: updateProduct

Request Details

The body of the request contains an updateProductRequest element to specify the updates to the Product Record. As a minimum, the values specified as mandatory for the createProduct function (see above) must be included.

The request content is similar to that for creating a Product Record. After the call, the Product Record is updated to match the request.

See the associated WSDL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an updateProductResponse root element, consisting of a Status element populated with SUCCEEDED.

If an error occurs, the updateProductResponse root element contains the elements shown in the following table.

Returned Elements

Element	Type	Description
Status	string	Contains FAILED to indicate an error
Message/Message/Severity	string	Contains ERROR to indicate an error
Message/Message/Description	string	The error message, for example: Specification Type FOOD does not map to a single Specification Type Format in Brand Compliance, product not added

> **Note:** Even if the request succeeds, additional warning messages may be included. Check the Status element for SUCCEEDED or FAILED to determine the success or failure of the call.

Error messages are the same as createProduct, plus the messages shown in the following table.

Error Messages

Element	Message	Meaning
	Error: API does not support multiple variants per Product Record.	Updates may only be applied if the portal is operating in the mode where a separate Product Record is created for individual product variants (such as sizes)
ProductNumber	Error: Product Record not found in Brand Compliance Management Cloud Service (based on Product Number and Supplier Code). Product Record status must be Draft or Active.	Product Record not found

submitProduct

Description

Updates an existing Product Record or creates a new Product Record if it does not already exist. Use this function to pass data from the external system and have the API automatically determine whether a Product Record needs to be updated or created based on whether it already exists in Brand Compliance.

The Product Record to be updated is identified by matching Product Number, Product Name, and Supplier Code as a minimum.

Endpoint address: /services/productService
HTTP method: submitProduct

Request Details

The body of the request contains a submitProduct element to specify the Product Record details. As a minimum, the values specified as mandatory for the createProduct function (see above) must be included.

This service determines if the specified product already exists and if so, updates it (per updateProduct above). Otherwise, it creates the product (per createProduct above).

See the associated WSDL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 3 - Product* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a submitProductResponse root element, consisting of a Status element populated with SUCCEEDED.

Error Messages

If an error occurs, the submitProductResponse root element contains the elements shown in the following table.

Returned Elements

Element	Type	Description
Status	string	Contains FAILED to indicate an error
Message/Message/Severity	string	Contains ERROR to indicate an error
Message/Message/Description	string	The error message, for example: Specification Type FOOD does not map to a single Specification Type Format in Brand Compliance, product not added

Note: Even if the request succeeds, additional warning messages may be included. Check the Status element for SUCCEEDED or FAILED to determine the success or failure of the call.

Retrieve Product Specifications

This section describes the API for Product Specifications. The following function is available:

- getProductSpecificationV1: retrieves a list of Product Specifications in a page list

getProductSpecificationV1

Description

Retrieves a list of Product Specification in a paged list. Use this to extract specifications from Brand Compliance for use in the external system, using parameters to filter the returned specifications to meet the desired selection criteria.

Endpoint address: /services/getProductSpecificationV1
HTTP method: getProductSpecification

Request Details

Parameters are passed as URI parameters

URI Parameters

Parameter	Mandatory/ Optional	Type	Multiple Value Separator (~) Supported?	Example	Validation
offset	Mandatory	int	No	60	>=0
batchSize	Mandatory	int	No	30	>0 and <=100
specType	Optional	string	Yes	FOOD	FOOD FNF CNF PRODUCE BWS
language	Optional	string	Yes	en_GB	The Code for the required Language, such as en_GB. Specifying a preferred language will return only values in that language. If not specified, the base language is assumed

Parameter	Mandatory/ Optional	Type	Multiple Value Separator (~) Supported?	Example	Validation
specStatus	Optional	string	Yes	RETAILER_ DRAFT	SUPPLIER_DRAFT RETAILER_DRAFT COLLABORATIVE_DRAFT GATE_STEP PART_PACK_COPY_SENT PACK_COPY_SENT PACK_COPY_READY READY_FOR_ AUTHORISATION SUPPLIER_AUTHORISED ACTIVE OFF_RANGE DE_LISTED SUPERSEDED NOT_PROGRESSED APPROVE_FOR_ LABELLING PRODUCE_DRAFT PRODUCE_PACK_COPY PRODUCE_APPROVED PRODUCE_ARCHIVED
countryWhereSold	Optional	string	Yes	FOOD_ COUNTRY_ WHERE_ SOLD	The Code for the required Country Where Sold.
productNumber	Optional	string	Yes	12345	Filters on the Product Number column of the Product Coverage table.

Parameter	Mandatory/ Optional	Type	Multiple Value Separator (~) Supported?	Example	Validation
sectionType	Optional	string	Yes	RECIPE AND RAW MATERIALS	MAIN DETAILS RECIPE AND RAW MATERIALS NUTRITION ALLERGY AND DIETARY ADVICE PACKAGING FINISHED PRODUCT STANDARDS STORAGE OTHER LABELLING COPY CLAIMS SUBSTANTIATION PROCESS CONTROLS BATCH CODING PRODUCT APPROVAL REQUIREMENTS POST LAUNCH INFORMATION COMPONENTS FNF STORAGE PRODUCT REQUIREMENTS COUNTER TICKET PRODUCT CHARACTER COMP
productCoverage	Optional	string	Yes	Chicken (500g)	Filters on the Product Name column of the Product Coverage table.
supplier	Optional	string	Yes	A0001	Filters on the Primary Sites supplier code for Produce Specifications or the Supplier code for all other types of Specification.
site	Optional	string	Yes	A0001-001	Filters on the code of the Sites in the Primary Sites table of the Product Specification. Use the full site code incorporating the Supplier Code, such as A0001-001.
fromDate	Optional	string	No	2013-06-10T09:00:00	Can be used in conjunction with toDate to form a date range or can be specified individually.
toDate	Optional	string	No	2013-06-10T09:00:00	Can be used in conjunction with fromDate to form a date range or can be specified individually.

For example, to retrieve the Main Details and Nutrition sections for active Food specifications where the product number is ABC001 from a specific supplier/site, the request would be:

```
<request>
  <batchsize>100</batchsize>
  <offset>0</offset>
  <specStatus>ACTIVE</specStatus>
  <specType>FOOD</specType>
  <countryWhereSold>UK</countryWhereSold>
  <productNumber>ABC001</productNumber>
  <language>en_GB</language>
  <sectionType>MAIN DETAILS</sectionType>
  <sectionType>NUTRITION</sectionType>
  <supplier>A0001</supplier>
  <site>A0001-0001</site>
  <fromDate>2016-01-01T01:00:00</fromDate>
  <toDate>2016-10-01T01:00:00</toDate>
</request>
```

If no filtering is required, omit the parameter from the call. For example, if all specification types are to be included, omit the specType element (rather than including it with no value specified).

If multiple values for a parameter are to be included, repeat the parameter in the call. For example if Main Details and Nutrition section type are to be returned, include two specSection elements in the call, one for MAIN DETAILS and one for NUTRITION.

Searching with either fromDate or toDate specified, as well as one or more specStatus, filters the specifications on the Status Change History. Use this to retrieve specifications that changed to a specific status during a date range.

Searching with either fromDate or toDate specified without specStatus filters on the specification's last amended date.

Response Details

If successful, an HTTP 200 response is sent with a body containing details of the specifications that match the selection criteria.

See the associated WSDL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 4 - Product (Food Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 5 - Product (CNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 6 - Product (FNF Specification)*, *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 7 - Product (BWS Specification)*, and *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 8 - Product (Produce Specification)* for details of their mapping to the fields within the Brand Compliance UI.

Error Messages

Element	Message	Meaning
offset	Unmarshalling Error: Exception Description: An error occurred unmarshalling the document Internal Exception: org.xml.sax.SAXParseException; cvc-maxInclusive-valid: Value '9999999999999999' is not facet-valid with respect to maxInclusive '2147483647' for type 'int'.	Invalid offset value
offset batchSize	Unmarshalling Error: Exception Description: An error occurred unmarshalling the document Internal Exception: org.xml.sax.SAXParseException; cvc-datatype-valid.1.2.1: '?' is not a valid value for 'integer'.	Non-numeric offset value Non-numeric batch size value
batchSize	<errorCode>INVALIDBATCHSIZE <errorMessage>The Batch Size must be between 1 and 100	Invalid batch size No batch size provided No batch size tag
specType	<errorCode>INVALIDSPECTYPE <errorMessage>The Spec Type <###> does not exist	Specification type not found
specStatus	<errorCode>INVALIDSPECSTATUS <errorMessage>The Status <###> does not exist	Spec Status not found
countryWhereSold	<errorCode>INVALIDCOUNTRYWHERESOLD <errorMessage>The Country Where Sold <###> does not exist	Country not found
language	<errorCode>INVALIDLANGUAGE <errorMessage>The Language <###> does not exist	Language not found
sectionType	<errorCode>INVALIDSECTIONTYPE <errorMessage>The Section Type does not exist	Section type not found
fromDate toDate	Unmarshalling Error: Exception Description: An error occurred unmarshalling the document Internal Exception: org.xml.sax.SAXParseException; cvc-datatype-valid.1.2.1: '?' is not a valid value for 'dateTime'.	Invalid date format

Project Activities

This section describes the API for Artwork integration. The following function is available:

- Activity Management: updates existing Artwork activities

Activity Management

Description

Updates the sub status of existing Artwork Project Activities. This function is used for the integration of Brand Compliance with an external Artwork Management system, to update the status of an artwork activity from the external system.

> **Note:** This service has been superseded by the Artwork Restful Service.

Endpoint address: /services/activityManagement
HTTP method: updateActivityStatus

Request Details

The body of the request contains an updateActivityStatus/activityUpdateRequest element consisting of the elements shown in the following table.

Request Elements

Element	Type	Description
activityName	string	Project activity's name
activityRecordId	string	Project activity record's internal ID
projectId	string	ID business key to the Project record
projectRecordId	string	Project record's internal ID
subStatusCode	string	Code business key to the activity's sub status

After the call, the Project Activity record's sub status is updated to match the request.

See the associated WSDL for a full list of the attributes, and the *Oracle Retail Brand Compliance Management Cloud Service Data Dictionary, Volume 9 - Project* for details of their mapping to the fields within the Brand Compliance UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an updateActivityStatusResponse root element, consisting of a return element populated with SUCCESS.

Error Messages

If an error occurs, the updateActivityStatusResponse root element's return element will contain FAILURE.

A

Appendix: Secure Development Guide

The web service APIs provided by Brand Compliance are designed to be secure by default. When consuming these services, there are recommended best practices for the clients which call them. These development and usage guidelines are provided to maintain secure use, and preserve integrity throughout the system domain.

The Brand Compliance web services are secured using HTTPS Basic Authentication. In order to use them, calling clients require an External System logon with which to authenticate. Transport encryption is enabled using TLS v1.2 as a minimum standard.

Best Practices for Web Service Use

This section describes some best practices.

Store External System Credentials Securely

In order to grant access to web services, credentials must be set up within Brand Compliance in an External System record. These credentials, used by the client to call web services, should be managed securely in the client's domain. It is advised *not* to store passwords in plaintext, either in a file or a database table.

Configure External Systems Permissions Appropriately

The permissions of External Systems are configurable, and determine which services may be accessed. The permissions model governs not only the service which is accessible (for example, The Product Service), but also the endpoints (or actions) permissible on that service (for example, Read a Product or Add a Product).

It is recommended that External Systems are set up with the minimum authorities required to perform their required function. External System records should have granular permissions wherever feasible, rather than be granted access to many disparate functions. This minimizes the level of disclosure possible if credentials are compromised.

Maintain Strong Passwords for External System Records

Configurable parameters within Brand Compliance govern the strength of passwords used for External Systems and the rules surrounding their expiry. These parameters are set to values which strike a suitable balance between usability and security by default.

It is advised that should a client wish to change any of these parameter values, it should only be to increase security. Changes which slacken security are *not* recommended.

Those responsible for maintaining clients which consume services should note that passwords for External Systems expire in the same manner as those for regular application users. A process for updating the credentials used by the client should be put in place. To facilitate this, Brand Compliance implements functionality for notifying elected users of pending password expiry for an External System. For full details of this functionality, and of External Systems in general, see the *Oracle Retail Brand Compliance Management Cloud Service Administration Guide*.

Make Use of the Web Service Log

Intrusion detection mechanisms are employed within the hosted environment of Brand Compliance to capture to prevent malicious behavior at the network level. In addition to this, protections against XML injection attacks exist at the logical tier.

The application also provides a Web Service Log for users with sufficient permissions to see what calls have been made to its services. This provides a user-centric audit of the web service activity within the application, and can help clients monitor usage, or gather metrics to understand usage behavior further.

Ensure Your Client Supports TLS v1.2 or Higher

All Brand Compliance deployments require communication over HTTPS, and use TLS v1.2 as a minimum encryption protocol. Clients of the web service must be developed within a technology which supports TLS v 1.2 or higher. For example, Java 7 does not support TLS v1.2 out of the box.

Limit the Size of Web Service Payloads

The web services are designed for continued, real-time use. It is recommended that calls to the services are made in a manner to retrieve or send only what is required.

Where larger result sets are necessary however, provision is made within the web service APIs to allow paging of data. By providing an offset and a page size, it is possible to retrieve a page of results at a time.

Glossary

API

Application Programming Interface, a type of web service. It provides a way of communicating with a particular computer program or internet service, such as the interfaces between Brand Compliance and other systems, by exchanging data as XML files.

DELETE

A web service call used to delete an existing resource (not generally used in the APIs exposed in Brand Compliance).

GET

A web service call used to retrieve data without modifying it, for example, to request details of specifications.

HEAD

Identical to GET except that the server does not return a message-body in the response. The header information retrieved is the same as for a GET request and obtains meta-information without transferring the entity-body itself.

HTTP

Hyper Text Transfer Protocol, an application protocol used as the foundation for data communication on the World Wide Web.

HTTPS

Communication over Hyper Text Transfer Protocol which is protected with an additional security layer. Used to authenticate the site visited and protect the exchanged data.

oneTimePad

A secure encryption technique which uses a string of randomly generated numbers to produce a key for encoding and decoding a message. Both the sender and receiver of the message have a copy of the key and each key (or section of the key) is used only once.

POST

A web service call used to create new resources, for example, to create a new supplier or new user record.

PUT

A web service call used to update an existing resource, for example, to update/amend an existing supplier or user record.

REST / RESTful

Representational State Transfer, an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed system.

In Brand Compliance, most new services or new versions of services will use RESTful architecture.

SOAP

Simple Object Access Protocol, an XML-based protocol for exchanging data over HTTP. In Brand Compliance, SOAP services are developed for specific, definite requirements or tend to be associated with historical services.

URI

Uniform Resource Identifier, a string of characters used to identify the name of a resource and enable it to be accessed over a network such as the World Wide Web. A common example of a URI is the Uniform Resource Locator (URL), or web address.

WADL

Web Application Description Language, an XML-based interface description language for describing the functionality offered by a RESTful web service.

WSDL

Web Services Description Language, an XML-based interface description language for describing the functionality offered by a SOAP web service.

XML

Extensible Markup Language, a language used to encode a document in a way which is both human readable and machine readable.

Markup symbols are used to define the content of the document and to indicate what that content is associated to in the sending or receiving system.

XSD

XML Schema Definition, defines the structure and data type of the elements in an XML document. XSD is used to validate the content of an XML document.