Web Services Reference for Oracle Billing Insight

Version 7.0, Rev. A
March, 2016

ORACLE

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.

Contents

Chapter 5: Examples of Web Services

Index

1 What's New in This Release

What's New in Web Services Reference for Oracle Billing Insight, Version 7.0, Rev. A

Table 1 lists changes in this revision of the documentation to support this release of the software.

Table 1. What's New in Web Services Reference for Oracle Billing Insight, Version 7.0, Rev. A

Topic	Description
"About Web Services Security" on page 10	Modified topic. Describes the use of Basic authentication with Oracle Billing Insight and how to configure token-based authentication if perferred.

What's New in Web Services Reference for Oracle Billing Insight, Version 7.0

Table 2 lists changes in this revision of the documentation to support this release of the software.

Table 2. What's New in Web Services Reference for Oracle Billing Insight, Version 7.0

Topic	Description
"Accessing API Resources Used in Earlier Product Versions" on page 12	New topic. Describes how to access URIs from earlier product product versions.
"Discovering Oracle Billing Insight RESTful Services Using a Web Browser" on page 13 "Discovering Oracle Billing Insight RESTful Services Using the Web Application Definition Language (WADL)" on page 14	Modified topics. The URI for discovering RESTful services has been updated.
Chapter 4, "RESTful API Resource Reference"	Modified topics. New Web services have been added. Some existing services have been refactored, resulting in the deprecation of resource URIs. The input format has been modified for several account, service agreement, and payment resources.
Web Service API Resources for Batch Analytics on page 22	New topic. API resources for the batch analytics resource (batchrpts) have been moved to a separate topic.

Table 2. What's New in Web Services Reference for Oracle Billing Insight, Version 7.0

Topic	Description
"Web Service API Resources for Hierarchies" on page 31 "Web Service API Resources for Notifications" on page 33	New topics. Added new hierarchy resource URIs for managing notifications and hierarchies.
Chapter 5, "Examples of Web Services"	Modified topics. Web service examples have been updated for refactored URIs. Also, new examples have been added for disputes, notifications, companies, and payments.
"Discovering Oracle Self-Service E-Billing RESTful Services Using the RESTful Services Client"	Deleted topic. Instructions for discovering the RESTful Services Client have been removed. This feature has been deprecated in Oracle Billing Insight version 7.0.

Additional Changes

This book also contains the following name changes in this version:

- The product name changed to Oracle Billing Insight.
- The Customer Service Representative application name changed to the Assisted Service application.
- The Billing and Payment application name changed to the Self-Service application.

2 Overview of Web Services

This chapter provides an overview of Oracle Billing Insight Web services. It includes the following topics:

About Oracle Billing Insight Web Services

Oracle Billing Insight Web services provide a programmatic interface over the Web that developers and system integrators can use to build applications to integrate with Oracle Billing Insight applications.

A *Web service* is a discrete piece of business logic, accessible through Internet protocols, that allows businesses to communicate with each other and with other clients, without detailed knowledge of each other's IT systems.

Figure 1 shows how Web services interact with a user and the Oracle Billing Insight database.

Figure 1. Web Services Communication with Oracle Billing Insight

The user communicates with Oracle Billing Insight over the Internet using HTTP and HTTPS requests. Oracle Billing Insight Web services use the Java API for RESTful Web services (JAX-RS) Specification.JAX-RS is a Java programming language API that provides support for creating Web services according to the Representational State Transfer (REST) architectural style.

The Web services APIs allow you to build programs to integrate with Oracle Billing Insight. Some common examples of client integrations include the following:

- **Web-based portal applications.** You can create customized Web-based applications, using Active Server Pages (ASPs), Java Server Pages (JSPs), or similar Web technologies that access Oracle Billing Insight through a Web services interface. For example, a portal site can create a portlet that invokes the Oracle Billing Insight Web services APIs to get a billing account summary or display the billing trend for the current user.
- **Native mobile applications.** Your native mobile applications can invoke Oracle Billing Insight Web services to perform functions, such as retrieving account balances and returning analytic reports. All Oracle Billing Insight Web services can be invoked by mobile applications.
- **CRM, provisioning, and back-office applications.** You can request a provisioning service to create companies, accounts, or services. The provisioning system can then call Oracle Billing Insight Web services to create those business objects in the Oracle Billing Insight application, which allows users to view their service information immediately. Similarly, an account receivable system that received a payment from a customer for an account could call an Oracle Billing Insight Web service to update the account balance in Oracle Billing Insight, enabling a user to view her updated account balance immediately.
- **Add-on read operations.** You can extend Oracle Billing Insight functionality to add additional reports that query data from Oracle Billing Insight database tables or any customized or additional database tables. These reports can be configured and function as part of the RESTful Web services.

Oracle Billing Insight provides a separate EAR file with Web-services functionality, which you deploy on a separate application server.

About Web Services Security

Oracle Billing Insight Web services uses Basic authentication. Custom token authentication is supported, but requires the configuration, as described in "Configuring Token-Based Authentication" on page 11.

HTTP Basic authentication (BA) implementation is the simplest method for enforcing access controls to Web resources because cookies, session identifiers, and login pages are not required. Instead, HTTP Basic authentication uses standard fields in the HTTP header, eliminating the need for handshakes. It is preferable to use HTTPS over or with Basic authentication. With Basic authentication, transmitted credentials are encoded with Base64 in transit, and are not encrypted or hashed.

With custom token authentication, the authentication API generates a unique token for each registered API user, and then the token for the user is included with each request to the service.

A *token* is a secure random text string with a default length of 48. The following string is an example of a token:

```
Dlc7lkpeVp9InmOUB82dJMg6LF7WQ6ZnujTHq8zP94uCWtjg
```

When a token is created, it stays on the server temporarily and expires automatically after a certain period. The default value is 20 minutes, and it is preconfigured. The valid token must be passed in an HTTP header for each subsequent Web service request. If a request is made with an invalid token, then an exception with status code 401, which is a standard code for unauthorized access, is returned.

User roles determine which functionality and data a user can access. For details about permissions assigned to each user role, see *Implementation Guide for Oracle Billing Insight*.

Using Basic Authentication

For basic authentication, you send the authentication credentials to the server for each service request.

To use basic authentication

- Place the authentication credentials in the Authorization field of the request header.
- Place the authorization method and a space (such as "Basic ") before the encoded user and password string. Combine the username and password into a string separated by a colon, as in *username:password*, where, the resulting string is encoded using the RFC2045-MIME variant of Base64, except it is not limited to 76 characters per line.

  For example, for user agent name ftown and password Password1234, the field is formed as follows:

  ```
  Authorization: Basic ZnRvd246UGFzc3dvcmQxMjM0
  ```

Configuring Token-Based Authentication

Oracle Billing Insight uses Basic authentication by default. If you want to use token-based authentication, you must modify the configuration in the spring-security.xml file.

To configure Oracle Billing Insight to use token-based authentication

1. Edit the spring-security.xml file located in the *EDX_HOME*/config/security/rs/ directory.
2. Uncomment the configuration XML for token-based authentication in the section:

   ```
   <security:http> for token based authentication
   ```

3. Comment out the configuration XML for Basic authentication in the section below the one you uncommented:

   ```
   <security:http> create-session=″stateless: use-expressions=″false″
   ```

4. Save the file.

Types of Web Services

Oracle Billing Insight uses the following types of Web services:

- **Transactional data access services.** Transactional data access services provide access to Oracle Billing Insight business objects and provide Create, Read, Update, and Delete (CRUD) operations to these objects. A list of these types of Web services can be is in *rsclient,* which is a Web-service test-harness application provided to help you find services. The Web service information in the rsclient test-harness application includes the name of the service, input, and output.
- **Analytics report services.** Analytics report services generate analytics reports from Oracle Billing Insight for various business needs. These services provide read-only operations. You can extend these services to retrieve additional information from Oracle Billing Insight by using the report XML file configuration.

About Web Services Localization

The locale of the content for requests and responses is typically based on the authenticated user's locale preference, which is set on the User Profile Preferences page in the Oracle Billing Insight Web application.

Each API call has a passed-in token that represents the authenticated user. The locale values are specified in the HTML header of requests and responses. In Web service requests, you can retrieve the locale from the Accept-Language attribute. Oracle Billing Insight supports one acceptable language for each request. In Web service responses, the locale value is in the Content-Language attribute.

The Date and Number values are displayed in the format that is specific to the locale. The currency amount is based on the billing account statement, not on the authenticated user's preferred locale. For more information about localizing your implementation, see "About Customizing Localization Values" on page 116.

Accessing API Resources Used in Earlier Product Versions

API resources that were available in earlier versions of Oracle Billing Insight are backward-compatible, and can still be accessed with one modification in their usage.

To access an API resource from an earlier product version

- In the URL, replace the `ebillingrs/rs/` path before the resource with `rs/api/v1/`.

  For example, in version 6.2, the following sample URL got a list of billing hierarchies for a user:

  http://myhost.example.com:7017/ebillingrs/rs/hierarchies/billing

  To use the 6.2 hierarchies/billing resource in version 7.0, change the URL as follows:

  http://myhost.example.com:7017/rs/api/v1/hierarchies/billing

  NOTE: Version 7.0 APIs are accessed with `rs/api/v2` in the URL path.

3 Discovering RESTful Web Service Resources

This chapter describes the various methods that you can use to view Oracle Billing Insight Web service resources. It includes the following topics:

Discovering Oracle Billing Insight RESTful Services Using a Web Browser

You can access read-only RESTful service resources, such as GET operations, using a Web browser.

To discover RESTful services using a Web browser

1. Start the RESTful Service server domain.
2. Start a new browser using the following URL, and enter a valid user name and password:

   http://hostname:port/rs/rsPostLogin.action

   This action returns a secure token for the session.
3. Edit the same URL with the name of an individual resource, and enter the edited URL in a different window of the same browser.

   For example, the following URL returns an account summary for all accounts that the login user can access:

   http://hostname:port/rs/api/v1/accounts

   http://hostname:port/rs/api/v2/accounts

   http://hostname:port/rs/api/latest/accounts

   The URL response returns link elements that you can use to drill down further. From the drill-down link, you can view all the GET Web services using a browser session.
4. To find response HTML header information, use third-party browser plug-ins to view the HTML header in a separate window.

   The following Web sites provide a toolbar for viewing an HTTP header for Firefox and describe how to add it to your Firefox browser:

   https://addons.mozilla.org/en-US/firefox/addon/live-http-headers/

   http://www.addictivetips.com/internet-tips/view-http-headers-of-any-web-page-in-firefox/

Discovering Oracle Billing Insight RESTful Services Using the Web Application Definition Language (WADL)

You can use Web Application Definition Language (WADL) to discover all supported Oracle Billing Insight RESTful service resources. WADL lets you view the overall XML schema.

To discover Oracle Billing Insight RESTful services using WADL

- Use the following URL:

  http://hostname.example.com:7017/rs/api/v1/application.wadl

  http://hostname.example.com:7017/rs/api/v2/application.wadl

4 RESTful API Resource Reference

This chapter describes the RESTful API resources for using Web services with Oracle Billing Insight. It includes the following topics:

Web Service API Resources for Authentication

To use custom token authentication, use the resource described in Table 3 to authenticate an Oracle Billing Insight user.

Table 3. API Resource for Authentication

Operation	Resource URI	API Input Parameters	Description
POST	/authentication	None	Returns an authenticated user token.
POST	/authentication/impersonation	csr_id, target_user_id	Returns an impersonation token.

Web Service API Resources for Statements

You use the statements resource described in Table 4 to manage statement information in Oracle Billing Insight.

Table 4. API Resources for Statements

Operation	Resource URI	API Input Parameters	Description
GET	/statements/	accountKey	Returns a list of statements available for the account, including statement date and statement number.
GET	/statements/latest	accountKey	Returns the latest statement summary for the account.
GET	/statements/{statementKey}/summary	accountKey, reportPeriodKey	Returns a statement summary.
GET	/statements/{statementKey}/accountSummary	accountKey, reportPeriodKey	Returns a summary for an account on a statement, including a list of service numbers.
GET	/statements/{statementKey}/serviceAgreements/{serviceKey}/summary	accountKey, reportPeriodKey	Returns a summary of a service for account on a statement.
GET	/statements/{statementKey}/serviceAgreements/{serviceKey}/usage	accountKey, reportPeriodKey	Returns a statement usage summary.

Table 4. API Resources for Statements

Operation	Resource URI	API Input Parameters	Description
GET	/statements/{statementKey}/serviceAgreements/{serviceKey}/usageDetails	accountKey, reportPeriodKey	Returns a summary of statement usage detail.
GET	/statements/{statementKey}/serviceAgreements/{serviceKey}/transactionDetails	serviceDetailKey	Returns the statement transaction details for a particular service agreement. The serviceKey can be retrieved from a call to the /statements/{statementKey}/accountSummary resource.
GET	/statements/{statementKey}/contentRefs	None	Returns a list of content IDs for the statement.
POST	/statements/{statementKey}/contentRefs	None	Adds a PDF reference to the statement, in which the contentRef is the ID for the PDF on the content management server. This resource can also add for DAISY and BRAILLE, content ID is in the payload.

Web Service API Resources for Analytics Based on the Billing Account

You use the analytics resources described in Table 5 to manage analytics that are based on the billing account.

Table 5. API Resources for Analytics That Are Based on the Billing Account

Operation	Resource URI	API Input Parameters	Description
GET	/analytics/accounts	period	Returns the overview summary of the account that the current user can access for a particular period. If no period ID is passed, then the data for the latest period that has data loaded is returned. The period must be the period name returned from a call to the /periods service.
GET	/analytics/accounts/{accountId}/summary	parameter(chartId), reportPeriodStart, reportPeriodEnd	Returns the account summary. The account_id can be retrieved from the response of the /analytics/accounts service.
POST	/analytics/accounts/{accountId}/summary/batchrpts	parameter(chartId), reportPeriodStart, reportPeriodEnd	Creates a batch report schedule request for the Account Billing Overview report.
GET	/analytics/accounts/{accountId}/amountDue	Amount Due(AmountDue)	Returns the current amount due for an account, which can be different from the statement due. The accountId can be retrieved from the response of the /analytics/accounts service.
GET	/analytics/accounts/{accountId}/billingTrend	reportPeriodStart, reportPeriodEnd	Returns the account billing trend for the period specified. The value of accountId can be retrieved from the response of the /analytics/accounts service.
POST	/analytics/accounts/{accountId}/billingTrend/batchrpts	reportPeriodStart, reportPeriodEnd	Creates a batch report schedule request for the Account Billing Trend report.
GET	/analytics/accounts/{accountId}/unbilledSummary	None	Returns an unbilled usage summary for an account.

Table 5. API Resources for Analytics That Are Based on the Billing Account

Operation	Resource URI	API Input Parameters	Description
GET	/analytics/accounts/{accountId}/unbilledSummaryByService	None	Returns an unbilled usage summary for an account by service
GET	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/unbilledByPlan	None	Returns the unbilled usage plan for a service.
GET	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/unbilledDetails	parameter(TariffKey), parameter(Usage TypeKey)	Returns the unbilled usage transaction details for a tariff.
GET	/analytics/accounts/{accountId}/serviceAgreements	period	Returns a list of service agreements for a particular account. If no period is passed, the data for the latest period that has billing data loaded is returned.
GET	/analytics/accounts/{accountId}/serviceAgreements/overview	period	Returns the Service Billing Overview for all service agreements within the account.
POST	/analytics/accounts/{accountId}/serviceAgreements/overview/batchrpts	period	Creates a batch report schedule request for the Service Billing Overview for all service agreements in the account report.
GET	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/overview	reportPeriodStart, reportPeriodEnd	Returns a Service Overview for a given service number.
POST	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/overview/batchrpts	reportPeriodStart, reportPeriodEnd	Creates a batch report schedule request for the Service Overview for a given service number.

Table 5. API Resources for Analytics That Are Based on the Billing Account

Operation	Resource URI	API Input Parameters	Description
GET	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/details	reportPeriodStart, reportPeriodEnd, parameter (usageType)	Returns service details.
POST	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/details/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (usageType)	Creates a batch report schedule request for the Service Details report.
GET	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/transactionDetails	serviceDetailKey	Returns the transaction details for a service agreement. The value of {service_number} is available from the call to the /analytics/accounts/{accountId}/serviceAgreements service.
GET	/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/usageTrend	reportPeriodStart, reportPeriodEnd	Returns the usage trend for a service agreement. The value of {service_number} is available from the call to the /analytics/accounts/{accountId}/serviceAgreements service.
GET	/analytics/accounts/{accountId}/highestSpendingServices	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the highest spending services for an account.
POST	/analytics/accounts/{accountId}/highestSpendingServices/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Highest Spending Services report.
GET	/analytics/accounts/{accountId}/mostExpensiveCalls	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most expensive calls for an account.
POST	/analytics/accounts/{accountId}/mostExpensiveCalls/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Most Expensive Calls report.

Table 5. API Resources for Analytics That Are Based on the Billing Account

Operation	Resource URI	API Input Parameters	Description
GET	/analytics/accounts/{accountId}/longestCalls	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the longest calls for an account.
POST	/analytics/accounts/{accountId}/longestCalls/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Longest Calls report.
GET	/analytics/accounts/{accountId}/mostFrequentlyCalledNumber	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most frequently called numbers for an account.
POST	/analytics/accounts/{accountId}/mostFrequentlyCalledNumbers/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Most Frequently Called Numbers report.
GET	/analytics/accounts/{accountId}/mostFrequentlyCalledDestinations	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most frequently called destinations for an account.
POST	/analytics/accounts/{accountId}/mostFrequentlyCalledDestinations/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Most Frequently Called Destinations report.
GET	/analytics/accounts/{accountId}/mostFrequentlyCalledCountries	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most frequently called countries for an account.
POST	/analytics/accounts/{accountId}/mostFrequentlyCalledCountries/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for Most Frequently Called Countries report.

Web Service API Resources for Batch Analytics

You can use the batch report (batchrpts) resource to manage batch analytic reporting for Oracle Billing Insight as described in Table 6.

Table 6. API Resources for Batch Analytics

Operation	Resource URI	API Input Parameters	Description
GET	/batchrpts	status:{completed, pending, failed}; hierarchyType: {"BILLING", "BUSINESS", or custom} If hierarchyType is null, it returns batchRpt for both types. The hierarchyType parameter applies only to B2B users; it is ignored for B2C users.	Returns a list of batch reports that the user can access.
GET	/batchrpts/scheduled	None	Returns a list of scheduled batch reports that the user can access.
GET	/batchrpts/scheduled/{batrptId}	None	Returns a recurring batch report request that is scheduled.
DELETE	/batchrpts/scheduled/{batrptId}	None	Deletes a recurring batch report request that is scheduled.
GET	/batchrpts/{batrptId}	None	Returns a batch report request.
DELETE	/batchrpts/{batrptId}	None	Deletes a batch report.
GET	/batchrpts/{batchrptId}/content	None	Returns batch report result content with pagination on.
GET	/batchrpts/{batchrptId}/download	downloadFormat: {xml, csv, pdf}	Downloads a batch report in the specified format.

Web Service API Resources for Analytics Based on the Billing Hierarchy (Business Edition Only)

You can use the analytics and dashboard resources to manage statement information for Oracle Billing Insight as described in Table 7.

The number of returned entries is based on the calling user's access within the company's billing hierarchy.

Table 7. API Resources for Analytics That Are Based on the Billing Hierarchy (Business Edition Only)

Operation	Resource URI	API Input Parameters	Description
GET	/dashboard/ accountBillingOverview	reportPeriodStart, reportPeriodEnd	Returns the account billing overview for all accounts that a user can access from her position in the company's billing hierarchy.
GET	/dashboard/ accountBillingTrend	reportPeriodStart, reportPeriodEnd	Returns the account billing overview for all accounts that a user can access from her position in the company's billing hierarchy.
GET	/dashboard/serviceTrend	reportPeriodStart, reportPeriodEnd	Returns the service agreement trend for all service agreements that a user can access from her position in the company's billing hierarchy.
GET	/analytics/ highestSpendingServices	reportPeriodStart, reportPeriodEnd, parameter(num Results)	Returns the highest-spending services of all services that a user can access from her position in the company's billing hierarchy.
POST	/analytics/ highestSpending Services/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Highest Spending Services report.
GET	/analytics/ mostExpensiveCalls	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most expensive calls of all service agreements that a user can access from her position in the company's billing hierarchy.
POST	/analytics/ mostExpensiveCalls/ batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Most Expensive Calls report.

Table 7. API Resources for Analytics That Are Based on the Billing Hierarchy (Business Edition Only)

Operation	Resource URI	API Input Parameters	Description
GET	/analytics/longestCalls	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the longest calls of all service agreements that a user can access from her position in the company's billing hierarchy.
POST	/analytics/longestCalls/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Longest Calls report.
GET	/analytics/mostFrequentlyCalledNumbers	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most frequently called numbers of all service agreements that a user can access from her position in the company's billing hierarchy.
POST	/analytics/mostFrequentlyCalledNumbers/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Most Frequently Called Numbers report.
GET	/analytics/mostFrequentlyCalledDestinations	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most frequently called destinations of all service agreements that a user can access from her position in the company's billing hierarchy.
POST	/analytics/mostFrequentlyCalledDestinations/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Most Frequently Called Destinations report.
GET	/analytics/mostFrequentlyCalledCountries	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Returns the most frequently called destinations of all service agreements that a user can access from her position in the company's billing hierarchy.
POST	/analytics/most_mostFrequentlyCalledCountries/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for Most Frequently Called Countries report.

Web Service API Resources for Analytics Based on the Business Hierarchy (Business Edition Only)

You use the analytics resource to manage analytics that are based on the business hierarchy as described in Table 8. The number of entries returned is based on the calling user's access to the company's specific business hierarchy.

Table 8. API Resources for Analytics That Are Based on the Business Hierarchy (Business Edition Only)

Operation	Resource URI	API Input Parameters	Description
GET	/analytics/ highestSpendingServices	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Returns the highest-spending service of all service agreements under a particular business hierarchy position.
POST	/analytics/ highestSpendingServices/ batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Creates a batch report schedule request for the Highest Spending Services report.
GET	/analytics/ mostExpensiveCalls	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Returns the most expensive calls of all service agreements under a particular business hierarchy position.
POST	/analytics/ mostExpensiveCalls/ batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults)	Creates a batch report schedule request for the Most Expensive Calls report.
GET	/analytics/longestCalls	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Returns the longest calls of all services agreements under a particular business hierarchy position.

Table 8. API Resources for Analytics That Are Based on the Business Hierarchy (Business Edition Only)

Operation	Resource URI	API Input Parameters	Description
POST	/analytics/longestCalls/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Creates a batch report schedule request for the Longest Calls report.
GET	/analytics/mostFrequentlyCalledNumbers	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Returns the most frequently called numbers of all services agreements under a particular business hierarchy position.
POST	/analytics/mostFrequentlyCalledNumbers/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Creates a batch report schedule request for the Most Frequently Called Numbers report.
GET	/analytics/mostFrequentlyCalledDestinations	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Returns the most frequently called destinations of all service agreements that a user can access from her position in the company's billing hierarchy.
POST	/analytics/mostFrequentlyCalledDestinations/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Creates a batch report schedule request for the Most Frequently Called Destinations report.
GET	/analytics/mostFrequentlyCalledCountries	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Returns the most frequently called countries of all service agreements that a user can access from her position in the company's billing hierarchy.

Table 8. API Resources for Analytics That Are Based on the Business Hierarchy (Business Edition Only)

Operation	Resource URI	API Input Parameters	Description
POST	/analytics/mostFrequentlyCalledCountries/batchrpts	reportPeriodStart, reportPeriodEnd, parameter (numResults), hierarchyId, hierarchyNodeId	Creates a batch report schedule request for Most Frequently Called Countries report.
GET	/analytics/groupSpendingTrend	reportPeriodStart, reportPeriodEnd, hierarchyId, hierarchyNodeId	Returns the group-spending trend for all groups under a particular business hierarchy position.
POST	/analytics/groupSpendingTrend/batchrpts	reportPeriodStart, reportPeriodEnd, hierarchyId, hierarchyNodeId	Creates a batch report schedule request for the Group Spending Trend report.
GET	/analytics/serviceSpendingTrend	reportPeriodStart, reportPeriodEnd, hierarchyId, hierarchyNodeId	Returns the service-spending trend for all services under a particular business hierarchy position.
POST	/analytics/serviceSpendingTrend/batchrpts	reportPeriodStart, reportPeriodEnd, hierarchyId, hierarchyNodeId	Creates a batch report schedule request for Service Spending Trend report.

Web Service API Resources for Billing Accounts

You use the accounts resource for billing account resources as described in Table 9.

Table 9. API Resources for Billing Accounts

Operation	Resource URI	API Input Parameters	Description
GET	/accounts	None	Returns all accounts for the current period. This information is retrieved from the billing hierarchy for a B2B user, or from a user account link table for a B2C user.
POST	/accounts	billerId, type{b2b, b2c}	Creates an account. For B2B, it also adds the account to the company's latest period of the billing hierarchy. The billing file load can override any attribute values entered.
GET	/accounts/{accountId}	billerId	Returns an account.
PUT	/accounts/{accountId}	billerId	Updates an account.
PUT	/accounts/{accountId}	billerId, endDate= now	Cancels an account.
DELETE	/accounts/{accountId}	None	Deletes an account.
GET	/accounts/{accountId}/ balance	None	Returns the current account balance. The balance can differ from the most recent statement.
PUT	/accounts/{accountId}/ balance	None	Updates the current balance of the account. The current balance can differ from the most recent statement.
POST	/accounts/{accountId}/ contacts	None	Creates a contact for the account.
GET	/accounts/{accountId}/ contacts	None	Returns all contacts for the account.
PUT	/accounts/{accountId}/ contacts/{contactKey}	None	Updates the contact for the account. <See payload>>
DELETE	/accounts/{accountId}/ contacts/{contactKey}	None	Deletes the contact from the account.

Web Service API Resources for Billing Periods

You use the periods resource described in Table 10 to manage billing period-related information in Oracle Billing Insight.

Table 10. API Resources for Billing Periods

Operation	Resource URI	API Input Parameters	Description
GET	/periods	numResults	Returns a list of the available billing periods.
GET	/periods/{periodId}	None	Returns the single period object.
GET	/periods/current	None	Returns the current period for the system date.
GET	/periods/latest	None	Returns the latest period for which a user has the billing data loaded.

Web Service API Resources for Companies

You use the companies resource described in Table 11 to manage company information in Oracle Billing Insight.

Table 11. API Resources for Companies

Operation	Resource URI	API Input Parameters	Description
GET	/companies	None	Returns all companies.
POST	/companies	None	Creates a company, creates a billing hierarchy for the company, and publishes the billing hierarchy to the current period. The load contains User id and email (enrollment).
GET	/companies/{companyId}	None	Returns a company's information.
PUT	/companies/{companyId}	Can add additional attributes.	Updates a company's information.
DELETE	/companies/{companyId}	None	Deletes a company's information.

Web Service API Resources for Disputes

You use the disputes resource described in Table 12 to manage disputes in Oracle Billing Insight.

Table 12. API Resources for Disputes

Operation	Resource URI	API Input Parameters	Description
GET	/disputes	itemType: {account, statement, transaction}, itemId	Returns a list of disputes by item type and ID. The response includes the internal Oracle Billing Insight ID and external reference number.
POST	/disputes	itemType: {account, statement, transaction}	Creates a dispute on a given item. The payload includes item id.
GET	/disputes/{disputeId}	None	Returns detailed information for a dispute.
PUT	/disputes/{disputeId}	None	Updates information for a dispute, including status. The payload includes the internal id, external system reference number, and other related information. The DisputeId is the internal database ID.
GET	/disputes/{disputeId}/comments	None	Returns dispute comments.
POST	/disputes/{disputeId}/comments	None	Creates a dispute comment.

Web Service API Resources for Hierarchies

You use the hierarchies resource described in Table 13 to manage hierarchy information in Oracle Billing Insight.

Table 13. API Resources for Hierarchies

Operation	Resource URI	API Input Parameters	Description
GET	/hierarchies	hierarchyType: {BILLING, or BUSINESS}	Returns a list of all hierarchies that the current user can access, including IDs and names.
POST	/hierarchies	None	Creates a new hierarchy with the type specified in the payload; hierarchyType: {BILLING, BUSINESS, or a custom value}.
POST	/hierarchies	None	Copies a hierarchy, using the original hierarchy type. The payload includes an existing hierarchyId. The hierType element is ignored.
POST	/hierarchies	None	Imports a hierarchy from an XML payload, using the type defined in importXML. Uses the importXML, startPeriod, and endPeriod elements. The payload does not include hierarchyId. The importXML element must be in a single line. Do not include “\n”. in JSON format, all double quotes (") must be preceded with a slash, as in \". In XML format, the importXML string must be placed inside <![CDATA[…]]>). The hierType element is ignored.
GET	/hierarchies/{hierarchyId}	None	Returns information about a particular hierarchy.
PUT	/hierarchies/{hierarchyId}	None	Updates a hierarchy’s information, including expire.
DELETE	/hierarchies/{hierarchyId}	None	Deletes a hierarchy.

Table 13. API Resources for Hierarchies

Operation	Resource URI	API Input Parameters	Description
GET	/hierarchies/{hierarchyId}/myRootNodes	period	Returns the list of root nodes of the hierarchy that the current user can access, for the latest period. You can optionally specify a different period.
GET	/hierarchies/{hierarchyId}/linkTargetTypes	None	Returns a list of valid link target types for a hierarchy.
GET	/hierarchies/{hierarchyId}}/nodes/{nodeId}/children	period	Returns a list of children nodes for a particular node with node index, for the latest period.
GET	/hierarchies/{hierarchyId}/nodes/{nodeId}/linkTargets	period; linkTargetType: {account,serviceAgreement, group, company, custom type}	Returns a list of link target objects under a particular node within the hierarchy, for the latest period.
POST	/hierarchies/{hierarchyId}/nodes/{nodeId}/linkTargets	nodeName; linkTargetType; linkTargetURI; linkTargetExternalKey	Adds a linkTarget object to a node as a child node.
GET	/hierarchies/{hierarchyId}/linkTargets/{linkTargetURI}	period	Returns the node that represent a particular linktarget URI, for the latest period.
DELETE	/hierarchies/{hierarchyId}/linkTargets/{linkTargetURI}	period	Removes a particular linkTarget from the hierarchy, for the latest period.
DELETE	/hierarchies/{hierarchyId}/nodes/{nodeId}	period	Deletes the node and its child node from a heirarchy, for the latest period.
GET	/hierarchies/{hierarchyId}/nodes/{nodeId}/users	status:{"assigned","unassigned", "authorized", "unauthorized"}	Returns a list of users assigned to this node. If no status is specified, the default is assigned.
POST	/hierarchies/{hierarchyId}/nodes/{nodeId}/users	None	Assigns a user to a node with userid in the payload.

Table 13. API Resources for Hierarchies

Operation	Resource URI	API Input Parameters	Description
DELETE	/hierarchies/{hierarchyId}/nodes/{nodeId}/users/{userId}	None	Removes an assigned user from the node.
GET	/hierarchies/{hierarchyId}/periods	None	Returns a list of available periods for a hierarchy.
POST	/hierarchies/{hierarchyId}/periods	startPeriod, endPeriod	Publishes a hierarchy for specific periods.

Web Service API Resources for Notifications

You use the notifications resources described in Table 14 to manage notifications in Oracle Billing Insight.

Table 14. API Resources for Notifications

Operation	Resource URI	API Input Parameters	Description
GET	/notifications	None	Returns a list of notifications that a user can choose to receive.
PUT	/notifications	None	Updates the notification preferences with the list of selected notifications in the payLoad XML.

Web Service API Resources for Service Agreements

You use the serviceagreements resources described in Table 15 to manage service agreements in Oracle Billing Insight.

Table 15. API Resources for Service Agreements

Operation	Resource URI	API Input Parameters	Description
GET	/serviceAgreements	accountId	Returns all service agreements for the current period.
POST	/serviceAgreements	None	Creates a service agreement and adds it to the billing hierarchy. The payload contains serviceAgreementId or (billerId, accountNumber saNumber).
GET	/serviceAgreements/{serviceAgreementId}	None	Returns information for a service agreement.
PUT	/serviceAgreements/{serviceAgreementId}	None	Updates information for a service agreement.
PUT	/serviceAgreements/{serviceAgreementId}	None	Cancels a service agreement. The payload must contain endDate=now to cancel a service agreement.

Web Service API Resources for Users

You use the users resources described in Table 16 to manage user information in Oracle Billing Insight.

Table 16. API Resources for Users

Operation	Resource URI	API Input Parameters	Description
GET	/users	None	Returns a list of users that the authenticated user can access.
POST	/users	None	Creates a user for synchronization only.
GET	/users/{userId}	None	Returns a user.
PUT	/users/{userId}	None	Updates a user's information.
DELETE	/users/{userId}	None	Deletes a user (soft delete).

Web Service API Resources for Payments

You use the payment resources described in Table 17 to manage payment information in Oracle Billing Insight.

Table 17. API Resources for Payments

Operation	Resource URI	API Input Parameters	Description
GET	/payment/transactions	billingAccountid:{}, status:{2000, 2002, 2004, 2005}. Status values are from column ID of EDX_PMT_PAYMENT_STATUS_DEF table), Optional: startDate and endDate. If the start date is empty, the default is 30 days prior to today's date.	Returns all payment transactions made for a given billing account, if provided. If billingAccountid is not provided, all payments viewable by the current user are returned.
POST	/payment/transactions	None.	Makes a payment for a given amount on the account using the existing specified payment account. The billing account ID, payment account ID, and amount are in the payload and must be I18n compatible. If payment account is null, the default payment account is used.
GET	/payment/transactions/{paymentId}	None	Returns information about a particular payment. For credit card payments, only the last 4 digits of the card number are returned.
GET	/payment/transactions/external	None	Returns all external payment transactions made in an external system for a given billing account.
POST	/payment/transactions/external	None	Posts a payment transaction for a particular billing account, made externally, to Oracle Billing Insight.

Table 17. API Resources for Payments

Operation	Resource URI	API Input Parameters	Description
GET	/payment/transactions/external/{trans_id}	None	Returns a payment transaction made in an external system.
PUT	/payment/transactions/external/{trans_id}	None	Updates a payment transaction made in an external system.
GET	/payment/recurringPmts	billingAccountid: {}	Returns a list of recurring payments created for a given billing account, if provided. If billingAccountid is not provided, all recurring payments created by the authenticated user are returned.
POST	/payment/recurringPmts	None	Create a recurring payment for a particular account. The billing account is included in the payload.
GET	/payment/recurringPmts/{recurringPmtId}	None	Returns a given recurring payment.
PUT	/payment/recurringPmts/{recurringPmtId}	None	Updates a given recurring payment.
DELETE	/payment/recurringPmts/{recurringPmtId}	None	Deletes a given recurring payment.
GET	/payment/pmtAccounts	None	Returns a list of payment accounts for the current authenticated user.
POST	/payment/pmtAccounts	None	Creates a payment account. The payload includes accountType {bank, creditcard, debitcard}.
GET	/payment/pmtAccounts/{pmtAccountId}	None	Returns a payment account.
PUT	/payment/pmtAccounts/{pmtAccountId}	None	Updates a payment account.
DELETE	/payment/pmtAccounts/{pmtAccountId}	None	Deletes a payment account.

Table 17. API Resources for Payments

Operation	Resource URI	API Input Parameters	Description
PUT	/payment/ pmtAccounts/ {pmtAccountId}/ billingAccounts	None	Updates shared billing accounts.
PUT	/payment/ pmtAccounts/ defaultPmtAccount/ {billingAcctId}/ {pmtAccountId}	None	Updates the default payment account for a billing account.
GET	/payment/ pmtAccounts/ billingAccount/ {billingAcctId}	None	Returns the payment accounts for a billing account.

5 Examples of Web Services

This chapter contains examples of how to use Oracle Billing Insight Web services. It includes the following topics:

Examples of the Authentication Web Service

This topic includes the following examples of how to use the authenticate Web service with Oracle Billing Insight:

Example of Authenticating a User

This example shows you how to use the authenticate Web service to authenticate and return a valid token for a particular user. The token in the XML response must be passed as a parameter in subsequent REST services requests.

Operation

Use the following operation to authenticate a user:

```
POST /authentication
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/authentication

Sample XML Input

The following is an example of XML input:

```
<credential>
  <username>ftown</username>
  <password>example</password>
</credential>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"username":"csradmin",
"password":"Password"
}
```

XML Response

The following is an example of the XML response:

```
<token> gI59AFXTa0p6XFgvMzPNOGMMNhYOhKKbcjGN0K8es6fYM5Po</token>
```

Example of Impersonating a User

This example shows you how to use the impersonation Web service to return a valid impersonation token for a particular user. The token in the XML response must be passed as a parameter in subsequent REST services requests.

Operation

Use the following operation to impersonate a user:

```
POST /authentication/impersonation
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/authentication/
impersonation?target_user_id=ftown&csr_id=$trust_user_name

Sample XML Input

The following is an example of XML input:

```
<credential>
  <username>$trust_user_name</username>
  <password>$trust_user_password</password>
</credential>
```

Sample JSON Input

The following is an example of JSON input:

```
{
   "username": "$trust_user_name",
   "password": "$trust_user_password"
}
```

XML Response

The following is an example of the XML response:

```
<token>fqKHXSNFh1NHbt9A5aWiCuPgFcorcvpD558Ws6RwOo8kgQe4</token>
```

Examples of the Analytics Web Services

This topic includes the following examples of how to use the analytics and billingrs Web services to handle account information in Oracle Billing Insight:

Example of Getting a Summary of Accounts

This example shows you how to use the analytics Web service to get a summary of the accounts that the authenticated user can access for a particular period.

Operation

Use the following operation to get a summary of accounts:

```
GET /analytics/accounts
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/analytics/accounts

XML Response

The following is an example of the XML response:

```
<accountList>
  <count>2</count>
  <account>
  <accountId>BS1|7836380B2B1</accountId>
     <billerId>BS1</billerId>
     <accountNumber>7836380B2B1</accountNumber>
     <link>/analytics/BS1%7C7836380B2B1
     summary?reportPeriodStart=93&reportPeriodEnd=93</link>
  </account>
  <account>
     <accountId>BS1|7836380DEMO1</accountId>
     <billerId>BS1</billerId>
     <accountNumber>7836380DEMO1</accountNumber>
     <link>/analytics/BS1%7C7836380DEMO1/
     summary?reportPeriodStart=93&reportPeriodEnd=93</link>
  </account>
</accountList>
```

Example of Getting Service Agreement Details

This example shows you how to use the analytics Web service to get service agreement details that an authenticated user can access for a particular period.

Operation

Use the following operation to get service agreement details:

```
GET /analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/details
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/analytics/accounts/BS1|7836380DEMO1/
serviceAgreements/878-443-DEM1/details?reportPeriodStart=93&reportPeriodEnd=93

XML Response

The following is an example of the XML response:

```
<report>
  <serviceList>
     <count>12</count>
     <totalPages>6</totalPages>
     <firstPage>/analytics/BS1%7C7836380DEMO1/service_agreements/878-443-DEM1
     /details?reportPeriodEnd=93&reportPeriodStart=93&page=1</firstPage>
     <currentPage>/analytics/BS1%7C7836380DEMO1/service_agreements/878-443-DEM1/
     details?reportPeriodEnd=93&reportPeriodStart=93&page=1</currentPage>
     <nextPage>/analytics/BS1%7C7836380DEMO1/service_agreements/878-443-DEM1
     /details?reportPeriodEnd=93&reportPeriodStart=93&page=2</nextPage>
     <lastPage>/analytics/BS1%7C7836380DEMO1/service_agreements/878-443-DEM1
     /details?reportPeriodEnd=93&reportPeriodStart=93&page=6</lastPage>
     <header><Total>USD</Total></header>
  <service>
     <Call_date pattern="MM/dd/yyyy">08/15/2012</Call_date>
     <Call_time>13 : 12</Call_time>
     <service_number>878-443-DEM1</service_number>
     <service_name>btwob ligan</service_name>
     <Called_Number>650-359-8601</Called_Number>
     <category>Business</category>
     <memo_img/><memo/>
     <dispute/>
     <DISPUTE_NUM/>
     <Dispute_detail_desc/>
     <Usage_Type>Voice</Usage_Type>
     <Call_Type>Roaming -Vodafone</Call_Type>
     <DESTINATION_NAME>Foster City, TN USA</DESTINATION_NAME>
     <Duration>32 Minutes</Duration>
     <Total>3.00</Total>
     <link>/analytics/BS1%7C7836380DEMO1/service_agreements/878-443-DEM1
     /transaction_details?serviceDetailKey=263</link>
  </service>
  <service>
     <Call_date pattern="MM/dd/yyyy">08/14/2012</Call_date>
     <Call_time>21 : 11</Call_time>
     <service_number>878-443-DEM1</service_number>
     <service_name>BusinessUser</service_name>
     <Called_Number/>
     <category>Business</category>
     <memo_img/><memo/>
     <dispute/>
     <DISPUTE_NUM/>
     <Dispute_detail_desc/>
     <Usage_Type>Data</Usage_Type>
     <Call_Type>Data Transfers - Uploads</Call_Type>
     <DESTINATION_NAME>Los Angeles, CA USA</DESTINATION_NAME>
     <Duration>175 Kilobytes</Duration>
     <Total>1.40</Total>
     <link>/analytics/BS1%7C7836380DEMO1/service_agreements/878-443-DEM1
     transaction_details?serviceDetailKey=257</link>
```

```
  </service>
</serviceList>
</report>
```

Example of Creating a Batch Report Schedule Request for an Account Billing Overview

This example shows you how to use the analytics Web service to schedule a batch report request for an Account Billing Overview report.

Operation

Use the following operation to get a batch report summary:

```
GET /analytics/accounts/{account_id}/summary/batchrpts
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/analytics/accounts/BS1%7CAC-B2B-123361-01/summary/batchrpts

Sample XML Input

The following is an example of XML input:

```
<batchRequestInfo>
  <reportName>AccountBillingOverview</reportName>
  <runType>SINGLE</runType>
  <csv>Y</csv>
  <xml>Y</xml>
  <privateFlag>N</privateFlag>
</batchRequestInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
   "reportName": "AccountBillingOverview",
   "runType": "SINGLE",
   "csv": "Y",
   "xml": "Y",
   "privateFlag": "N"
}
```

XML Response

The following is an example of the XML response:

Batch report is created successfully!

Visit <link>http://myhost.example.com:7017/rs/api/v2/batchrpts/4</link>

Example of Getting a Batch Report Request

This example shows you how to use the batchrpts Web service to get a batch report request.

Operation

Use the following operation to get a batch report request:

```
GET /batchrpts/{batchrptId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/batchrpts/2

XML Response

The following is an example of the XML response:

```
<batchRequestInfo>
  <batchId>2</batchId>
  <status>PENDING</status>
  <reportName>AccountBillingOv</reportName>
  <reportId>telco_std_r1</reportId>
  <origReportName>Account Billing Overview</origReportName>
  <userId>41</userId>
  <userName>ftown</userName>
  <companyName>Cuelle Mobile Corporation</companyName>
  <hierarchyId>8000</hierarchyId>
  <hierarchyName>Billing Hierarchy</hierarchyName>
  <nodeId>80000</nodeId>
  <position>Cuelle Mobile Corporation</position>
  <privateFlag>N</privateFlag>
  <csv>Y</csv>
  <xml>N</xml>
  <pdf>N</pdf>
  <createDate>08/23/2013</createDate>
  <batchParamMap>
  <entry>
     <key>form.reportPeriodEnd</key>
     <value>105</value>
  </entry>
  <entry>
     <key>form.accountKey</key>
     <value>145</value>
  </entry>
  <entry>
```

```
      <key>costReallocForm.active</key>
      <value>N</value>
    </entry>
    <entry>
      <key>form.reportPeriodStart</key>
      <value>105</value>
    </entry>
    <entry>
      <key>form.parameter(AccountKey)</key>
      <value>145</value>
    </entry>
    </batchParamMap>
  </batchRequestInfo>
```

Example of Getting Batch Report Content With Pagination On

This example shows you how to use the batchrpts Web service to get batch report result content with pagination on.

Operation

Use the following operation to return batch report result content with pagination on:

```
GET /batchrpts/{batrpt_id}/content
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/batchrpts/content/3

XML Response

The following is an example of the XML response:

```
<report>
  <serviceList>
  <totalPages>2</totalPages>
  <firstPage>/batchrpts/content/8?page=1</firstPage>
  <currentPage>/batchrpts/content/8?page=1</currentPage>
  <nextPage>/batchrpts/content/8?page=2</nextPage>
  <lastPage>/batchrpts/content/8?page=2</lastPage>
  <count>11</count>
  <header>
  <Total>USD</Total>
  </header>
  <service>
    <Call_date pattern="MM/dd/yyyy">09/24/2013</Call_date>
    <Call_time>06:40</Call_time>
    <service_number>999-888-9903</service_number>
```

```
   <service_name>Marisa ODoole</service_name>
   <Called_Number>999-888-9905</Called_Number>
   <category>Personal</category>
   <memo_img/>
   <memo/>
   <dispute/>
   <DISPUTE_NUM/>
   <Dispute_detail_desc/>
   <Usage_Type>Message</Usage_Type>
   <Call_Type>Text Message Sent</Call_Type>
   <DESTINATION_NAME>New York, NY USA</DESTINATION_NAME>
   <Duration>1 Messages</Duration>
   <Total>0.02</Total>
</service>
<service>
   <Call_date pattern="MM/dd/yyyy">09/21/2013</Call_date>
   <Call_time>18:40</Call_time>
   <service_number>999-888-9903</service_number>
   <service_name>Marisa ODoole</service_name>
   <Called_Number>999-888-9901</Called_Number>
   <category>Personal</category>
   <memo_img/>
   <memo/>
   <dispute/>
   <DISPUTE_NUM/>
   <Dispute_detail_desc/>
   <Usage_Type>Message</Usage_Type>
   <Call_Type>Text Message Received</Call_Type>
   <DESTINATION_NAME>Anytown, CA USA</DESTINATION_NAME>
   <Duration>1 Messages</Duration>
   <Total>0.02</Total>
</service>
<service>
   <Call_date pattern="MM/dd/yyyy">09/19/2013</Call_date>
   <Call_time>20:11</Call_time>
   <service_number>999-888-9903</service_number>
   <service_name>Marisa ODoole</service_name>
   <Called_Number>*9977</Called_Number>
   <category>Business</category>
   <memo_img/>
   <memo/>
   <dispute/>
   <DISPUTE_NUM/>
   <Dispute_detail_desc/>
   <Usage_Type>Data</Usage_Type>
   <Call_Type>Data Downloaded</Call_Type>
   <DESTINATION_NAME>Los Angeles, CA USA</DESTINATION_NAME>
   <Duration>8 Kilobytes</Duration>
   <Total>2.00</Total>
</service>
<service>
   <Call_date pattern="MM/dd/yyyy">09/17/2013</Call_date>
   <Call_time>19:40</Call_time>
   <service_number>999-888-9903</service_number>
```

```
      <service_name>Marisa ODoole</service_name>
      <Called_Number>999-888-9902</Called_Number>
      <category>Personal</category>
      <memo_img/>
      <memo/>
      <dispute/>
      <DISPUTE_NUM/>
      <Dispute_detail_desc/>
      <Usage_Type>Message</Usage_Type>
      <Call_Type>Text Message Sent</Call_Type>
      <DESTINATION_NAME>New York, NY USA</DESTINATION_NAME>
      <Duration>1 Messages</Duration>
      <Total>0.02</Total>
   </service>
   <service>
      <Call_date pattern="MM/dd/yyyy">09/17/2013</Call_date>
      <Call_time>09:19</Call_time>
      <service_number>999-888-9903</service_number>
      <service_name>Marisa ODoole</service_name>
      <Called_Number>923-783-8934</Called_Number>
      <category>Business</category>
      <memo_img/>
      <memo/>
      <dispute/>
      <DISPUTE_NUM/>
      <Dispute_detail_desc/>
      <Usage_Type>Voice</Usage_Type>
      <Call_Type>Peak call incoming</Call_Type>
      <DESTINATION_NAME>Los Angeles, CA USA</DESTINATION_NAME>
      <Duration>8 Minutes</Duration>
      <Total>1.60</Total>
   </service>
      </serviceList>
</report>
```

Example of Downloading a Batch Report

This example shows how to use the batchrpts Web service to download a batch report in CSV file format.

Operation

Use the following operation to get a batch report:

```
GET /batchrpts/download/csv/{batrpt_id}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/batchrpts/3/download?downloadFormat=csv

Example of Deleting a Batch Report Request

This example shows how to use the batchrpts Web service to delete a batch report request.

Operation

Use the following operation to delete a batch report request:

```
DELETE /batchrpts/{batchrptId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/batchrpts/3

Response

Batch request deleted.

Example of Getting a Summary of Unbilled Usage for an Account

This example shows you how to use the analytics Web service to get a summary of the unbilled usage for an account.

Operation

Use the following operation to get a summary of unbilled usage for an account:

```
GET /analytics/accounts/{accountId}/unbilledSummary
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/analytics/accounts/BS1%7CAC-B2B-123361-01/unbilledSummary

XML Response

The following is an example of the XML response:

```
<report>
<usageTypeList>
  <count>1</count>
  <header></header>
  <usageType>
     <LastStmtDate pattern="MM/dd/yyyy">10/21/2013</LastStmtDate>
     <DaysUsed>1</DaysUsed>
     <DaysInCycle>31</DaysInCycle>
```

```
    </usageType>
  </usageTypeList>
  <usageTypeList>
    <count>3</count>
    <header></header>
    <usageType>
       <LastStmtDate pattern="MM/dd/yyyy">10/21/2013</LastStmtDate>
       <UsageCode>VOICE</UsageCode>
       <UsageName>Voice</UsageName>
       <UnitName>Minutes</UnitName>
       <Usage>880.00</Usage>
       <Allowance>2,500.00</Allowance>
       <DaysUsed>1</DaysUsed>
       <DaysInCycle>31</DaysInCycle>
       <alertThreshold>500.0</alertThreshold>
    </usageType>
    <usageType>
       <LastStmtDate pattern="MM/dd/yyyy">10/21/2013</LastStmtDate>
       <UsageCode>MESSAGE</UsageCode>
       <UsageName>Message</UsageName>
       <UnitName>Messages</UnitName>
       <Usage>27.00</Usage>
       <Allowance>2.00</Allowance>
       <DaysUsed>1</DaysUsed>
       <DaysInCycle>31</DaysInCycle>
       <alertThreshold>0.4</alertThreshold>
    </usageType>
    <usageType>
       <LastStmtDate pattern="MM/dd/yyyy">10/21/2013</LastStmtDate>
       <UsageCode>DATA</UsageCode>
       <UsageName>Data</UsageName>
       <UnitName>Kilobytes</UnitName>
       <Usage>225.00</Usage>
       <Allowance>0.00</Allowance>
       <DaysUsed>1</DaysUsed>
       <DaysInCycle>31</DaysInCycle>
       <alertThreshold>0.0</alertThreshold>
    </usageType>
  </usageTypeList>
  </report>
```

Example of Getting a Summary of Unbilled Usage for an Account by Service

This example shows you how to use the analytics Web service to get a summary of the unbilled usage for an account by service.

Operation

Use the following operation to get a summary of unbilled usage data for an account by service:

```
GET /analytics/accounts/{accountId}/unbilledSummaryByService
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/analytics/accounts/BS1%7CAC-B2B-123361-01/unbilledSummaryByService

XML Response

The following is an example of the XML response:

```
<report>
  <productList>
     <count>1</count>
     <header></header>
     <product><Product_Name>Business Circle 2500</Product_Name></product>
  </productList>
  <serviceList>
     <count>3</count>
     <header><Voice>Minutes</Voice><Message>Messages</Message><Data>Kilobytes<
     /Data>
     </header>
  <service>
     <service_num>555-444-3301</service_num>
     <Voice>523</Voice>
     <Message>10</Message>
     <Data>64</Data>
     <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301/
     unbilledByPlan</link>
  /service>
  <service>
     <service_num>555-444-3302</service_num>
     <Voice>375</Voice>
     <Message>8</Message>
     <Data>0</Data>
     <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3302/
     unbilledByPlan</link>
     </service>
  <service>
     <service_num>555-444-3303</service_num>
     <Voice>602</Voice>
     <Message>9</Message>
     <Data>161</Data>
     <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3303/
     unbilledByPlan</link>
  </service>
  </serviceList>
</report>
```

Example of Getting the Unbilled Usage Plan for a Service

This example shows you how to use the analytics Web service to get the unbilled usage plan for a service.

Operation

Use the following operation to get the unbilled usage plan for a service:

```
GET /analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/
unbilledByPlan
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/analytics/accounts/BS1%7CAC-B2B-123361-01/serviceAgreements/555-444-3301/unbilledByPlan

XML Response

The following is an example of the XML response:

```
<report>
  <productList>
     <count>1</count>
     <header></header>
     <product>
        <Product_Name>Business Circle 2500 - 555-444-3301</Product_Name>
     </product>
  </productList>
  <planList>
     <groupName>Data .25 / Kilobytes</groupName>
     <count>1</count>
     <header></header>
     <plan>
        <tariff_name>Data Downloaded</tariff_name>
        <txn_count>3</txn_count>
        <total_usage>64</total_usage>
        <unit_name>Kilobytes</unit_name>
        /analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
     /unbilledDetails?parameter(TariffKey)=102&parameter(UsageTypeKey)=103
     </plan>
  </planList>
  <planList>
     <groupName>Message 2 Messages</groupName>
     <count>3</count>
     <header></header>
     <plan>
        <tariff_name>Text Message Received</tariff_name>
        <txn_count>2</txn_count>
        <total_usage>2</total_usage>
        <unit_name>Messages</unit_name>
        <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
```

```
      /unbilledDetails?parameter(TariffKey)=4</link>
   </plan>
   <plan>
      <tariff_name>Text Message Sent</tariff_name>
      <txn_count>6</txn_count>
      <total_usage>6</total_usage>
      <unit_name>Messages</unit_name>
      <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
      /unbilledDetails?parameter(TariffKey)=3</link>
   </plan>
   <plan>
      <tariff_name>Text Message Allowance</tariff_name>
      <txn_count>2</txn_count>
      <total_usage>2</total_usage>
      <unit_name>Messages</unit_name>
      <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
      /unbilledDetails?parameter(TariffKey)=113</link>
   </plan>
</planList>
<planList>
   <groupName>Voice 2500 Minutes</groupName>
   <count>4</count>
   <header></header>
   <plan>
      <tariff_name>Roaming Verizon</tariff_name>
      <txn_count>2</txn_count>
      <total_usage>74</total_usage>
      <unit_name>Minutes</unit_name>
      <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
      /unbilledDetails?parameter(TariffKey)=104</link>
   </plan>
   <plan>
      <tariff_name>Family Circle Minutes</tariff_name>
      <txn_count>10</txn_count>
      <total_usage>152</total_usage>
      <unit_name>Minutes</unit_name>
      <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
      /unbilledDetails?parameter(TariffKey)=1</link>
   </plan>
   <plan>
      <tariff_name>Peak call incoming</tariff_name>
      <txn_count>5</txn_count>
      <total_usage>63</total_usage>
      <unit_name>Minutes</unit_name>
      <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
      /unbilledDetails?parameter(TariffKey)=106</link>
   </plan>
   <plan>
      <tariff_name>Peak call outgoing</tariff_name>
      <txn_count>2</txn_count>
      <total_usage>20</total_usage>
      <unit_name>Minutes</unit_name>
      <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
      /unbilledDetails?parameter(TariffKey)=107</link>
```

```
      </plan>
    </planList><planList>
      <groupName>Voice Unlimited Minutes</groupName>
      <count>2</count>
      <header></header>
      <plan>
        <tariff_name>Rollover Available</tariff_name>
        <txn_count>3</txn_count>
        <total_usage>145</total_usage>
        <unit_name>Minutes</unit_name>
        <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
        /  unbilledDetails?parameter(TariffKey)=114</link>
      </plan>
      <plan>
        <tariff_name>Night / Weekend Minutes</tariff_name>
        <txn_count>3</txn_count>
        <total_usage>69</total_usage>
        <unit_name>Minutes</unit_name>
        <link>/analytics/BS1%7CAC-B2B-123361-01/service_agreements/555-444-3301
        /unbilledDetails?parameter(TariffKey)=2</link>
      </plan>
    </planList>
  </report>
```

Example of Getting the Unbilled Usage Transaction Details for a Tariff

This example shows you how to use the analytics Web service to get the unbilled usage transaction details for a tariff.

Operation

Use the following operation to get the unbilled usage transaction details for a tariff:

```
GET /analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/
unbilledDetails
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/analytics/accounts/BS1%7CAC-B2B-123361-01/serviceAgreements/555-444-3301/unbilledDetails?parameter(TariffKey)=1&parameter(UsageTypeKey)=103

XML Response

The following is an example of the XML response:

```
<report>
  <transactionList>
  <count>10</count>
  <header></header>
     <transaction>
        <called_num>555-444-3305</called_num>
        <date pattern="MM/dd/yyyy">11/14/2013</date>
        <time>11:02</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>5</total_usage>
     </transaction>
     <transaction>
        <called_num>555-444-3302</called_num>
        <date pattern="MM/dd/yyyy">11/14/2013</date>
        <time>15:10</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>12</total_usage>
     </transaction><transaction>
        <called_num>555-444-3305</called_num>
        <date pattern="MM/dd/yyyy">11/14/2013</date>
        <time>11:02</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>25</total_usage>
     </transaction
     <transaction>
        <called_num>555-444-3304</called_num>
        <date pattern="MM/dd/yyyy">11/16/2013</date>
        <time>08:30</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>3</total_usage>
     </transaction>
     <transaction>
        <called_num>555-444-3303</called_num>
        <date pattern="MM/dd/yyyy">11/16/2013</date>
        <time>08:10</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>17</total_usage>
     </transaction>
     <transaction>
        <called_num>555-444-3302</called_num>
        <date pattern="MM/dd/yyyy">11/14/2013</date>
        <time>15:10</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>32</total_usage>
     </transaction>
     <transaction>
        <called_num>555-444-3305</called_num>
        <date pattern="MM/dd/yyyy">11/14/2013</date>
        <time>12:02</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>26</total_usage>
     </transaction>
     <transaction>
        <called_num>555-444-3304</called_num>
```

```
        <date pattern="MM/dd/yyyy">11/05/2013</date>
        <time>06:30</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>11</total_usage>
      </transaction>
      <transaction>
        <called_num>555-444-3303</called_num>
        <date pattern="MM/dd/yyyy">11/26/2013</date>
        <time>18:10</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>9</total_usage>
      </transaction>
      <transaction>
        <called_num>555-444-3302</called_num>
        <date pattern="MM/dd/yyyy">11/10/2013</date>
        <time>12:10</time>
        <area_cd>Anytown, CA</area_cd>
        <total_usage>12</total_usage>
      </transaction>
    </transactionList>
  </report>
```

Examples of the Accounts Web Service

This topic includes the following examples of how to use the account Web service to handle account information in Oracle Billing Insight:

Example of Getting a List of Billing Accounts for a User

This example shows you how to use the accounts Web service to get a list of the billing accounts that a user can access.

Operation

Use the following operation to get a list of accounts for a user:

```
GET /accounts
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/accounts

XML Response

The following is an example of the XML response:

```
<accountList>
  <count>2</count>
  account currency="USD">
     <accountId>BS1|7836380B2B1</accountId>
     <billerId>BS1</billerId>
     <accountNumber>7836380B2B1</accountNumber>
     <currBalance>999</currBalance>
     <companyId>Dutch Home Insurance</companyId>
     <contactName>John SMITH updated</contactName>
  </account>
  <account>
     <accountId>BS1|7836381B2B1</accountId>
     <billerId>BS1</billerId>
     <accountNumber>7836381B2B1</accountNumber>
     <accountName>7836381B2B1</accountName>
     <companyId>Dutch Home Insurance</companyId>
     <contactName>xxx</contactName>
     <address1>…</address1>
     <address2>…</address2>
     <address3>…</address3>
     <city>…</city>
     <state>…</state>
     <country>…</country>
     <postalCode>…</postalCode>
  </account>
</accountList>
```

Example of Creating a B2B Billing Account

This example shows you how to use the accounts Web service to create a B2B billing account and add it to the company's billing hierarchy.

Operation

Use the following operation to create a B2B account:

```
POST /accounts
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/accounts

Sample XML Input

The following is an example of XML input:

```
<accountInfo>
  <billerId>BS1</billerId>
  <accountNumber>AC-B2B-123368-99</accountNumber>
  <accountName>Account Name</accountName>
  <contactName>Account owner</contactName>
  <accountType>UNK</accountType>
  <companyId>CUELLE</companyId>
  <billCycleEndDay>21</billCycleEndDay>
  <startDate pattern="MM/dd/yyyy">06/12/2015</startDate>
  <billType>BF</billType>
  <paperOnFlag>Y</paperOnFlag>
  <billableFlag>Y</billableFlag>
  <b2bFlag>Y</b2bFlag>
  <flexField1>flex1</flexField1>
  <flexField2>flex2</flexField2>
  <flexField3>flex3</flexField3>
  <flexField4>flex4</flexField4>
  <flexField5>flex5</flexField5>
  <flexField6>flex6</flexField6>
  <flexField7>flex7</flexField7>
  <flexField8>flex8</flexField8>
  <flexField9>flex9</flexField9>
  <flexField10>flex10</flexField10>
</accountInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
   "billerId": "BS1",
   "accountNumber": "AC-B2B-123368-99",
   "accountName": "Account Name",
   "contactName": "Account owner",
   "accountType": "UNK",
   "companyId": "CUELLE",
   "billCycleEndDay": "21",
   "startDate": {
      "@pattern": "MM/dd/yyyy",
      "#text": "06/12/2015"
   },
   "billType": "BF",
   "paperOnFlag": "Y",
   "billableFlag": "Y",
   "b2bFlag": "Y",
   "flexField1": "flex1",
   "flexField2": "flex2",
   "flexField3": "flex3",
   "flexField4": "flex4",
   "flexField5": "flex5",
   "flexField6": "flex6",
```

```
   "flexField7": "flex7",
   "flexField8": "flex8",
   "flexField9": "flex9",
   "flexField10": "flex10"
}
```

Response Message

The following is an example of the response message:

Account Create Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/accounts/BS1%7C7836380B2B1</link>

Example of Creating a B2C Billing Account

This example shows you how to use the accounts Web service to create a B2C billing account.

Operation

Use the following operation to create a B2C account:

```
POST /accounts
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/accounts

Sample XML Input

The following is an example of XML input:

```
<accountInfo>
  <accountId>BS2|7836380B2C1</accountId >
  <billerId>BS2</billerId>
  <accountNumber>7836380B2C1</accountNumber>
  <accountType>PREPAY</accountType>
  <accountName>7836380B2C1</accountName>
  <contactName>xxx</contactName>
  <address1>…</address1>
  <address2>…</address2>
  <address3>…</address3>
  <city>…</city>
  <state>…</state>
  <country>…</country>
  <postalCode>…</postalCode>
</accountInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"userId":"rsamb2c2",
"role":"User",
"firstName":"Rob",
"lastName":"Sunny",
"email":"RSunny@example.com",
"language":"en_US",
"billerId":"BS2",
"accountNumber":"AC-B2C-7836380-1",
"userType":"b2c"
}
```

Response Message

The following is an example of the response message:

Account Create Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/accounts/BS2%7C7836380B2C1</link>

Example of Getting Billing Account Information

This example shows you how to use the accounts Web service to get the information for a particular billing account.

Operation

Use the following operation to get information for a billing account:

```
GET /accounts/{accountId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/ebillingrs/rs/api/v2/accounts/{accountId}

XML Response

The following is an example of the XML response:

```
<accountInfo currency="USD">
  <accountId>BS1|7836380B2B1</accountId>
  <billerId>BS1</billerId>
  <accountNumber>7836380B2B1</accountNumber>
  <currBalance>786.75</currBalance>
  <companyId>Dutch Home Insurance</companyId>
  <contactName>John Smith</contactName>
</accountInfo>
```

Example of Updating an Existing Billing Account

This example shows you how to use the accounts Web service to update the information for an existing billing account.

Operation

Use the following operation to update an existing billing account:

```
PUT /accounts/{accountId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/accounts/7836380B2B1

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- Tags that are not in bold are optional.
- If tags are not present, then the corresponding fields are not updated.
- If tags are present with an empty value, then the corresponding fields are set to empty if a null value is allowed.

Sample XML Input

The following is an example of XML input:

```
<accountInfo>
  <accountId>BS1|7836380B2B1</accountId>
  <billerId>BS1</billerId>
  <accountNumber>7836380B2B1</accountNumber>
  <accountName>7836380B2B1</accountName>
  <companyId>Dutch Home Insurance</companyId>
  <contactName>xxx</contactName>
  <address1>…</address1>
  <address2>…</address2>
  <address3>…</address3>
  <city>…</city>
  <state>…</state>
  <country>…</country>
  <postalCode>…</postalCode>
</accountInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
   "accountId": "BS1|7836380B2B1",
   "billerId": "BS1",
   "accountNumber": "7836380B2B1",
   "accountName": "7836380B2B1",
   "companyId": "Dutch Home Insurance",
   "contactName": "xxx",
   "address1": "…",
   "address2": "…",
   "address3": "…",
   "city": "…",
   "state": "…",
   "country": "…",
   "postalCode": "…"
}
```

Response Message

The following is an example of the response message:

Account Update Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/accounts/BS1%7C7836380B2B1</link>

Example of Getting a Billing Account Balance

This example shows you how to use the accounts Web service to return the balance amount for a billing account.

Operation

Use the following operation to get a billing account balance:

```
GET /accounts/{accountId}/balance
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/accounts/7836380B2B1/balance

XML Response

The following is an example of the XML response:

```
<accountInfo currency="USD">
  <accountId>BS1|7836380B2B1</accountId>
  <billerId>BS1</billerId>
  <accountNumber>7836380B2B1</accountNumber>
  <currBalance>999</currBalance>
```

```
  <companyId>Dutch Home Insurance</companyId>
  <contactName>John Smith</contactName>
</accountInfo>
```

Example of Updating a Billing Account Balance

This example shows you how to use the accounts Web service to update a billing account balance with additional information.

Operation

Use the following operation to update a billing account balance:

```
PUT /accounts/{account_id}/balance
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/accounts/836380B2B1/balance

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- Tags that are not in bold are optional.
- If tags are not present, then the corresponding fields are not updated.
- If tags are present with an empty value, then the corresponding fields are set to empty if a null value is allowed.
- The amount uses the localized format and is parsed in the locale that is passed in the Accept Language HTML attribute.

Sample XML Input

The following is an example of XML input:

```
<paymentAccountActivity>
  <billerId>BS1</billerId>
  <accountNumber>7836380B2B1</accountNumber>
  <currBalance>786.00</currBalance>
  <lastPaymentReceivedAmount>120.00</lastPaymentReceivedAmount>
  <lastPaymentReceivedDate pattern="MM/dd/yyyy">08/13/2012<
  lastPaymentReceivedDate>
</paymentAccountActivity>
```

Response Message

The following is an example of the response message:

Update Success

Examples of the Companies Web Service

This topic includes the following examples of how to use the companies Web service to handle company information in Oracle Billing Insight:

Example of Getting a List of Companies

This example shows you how to use the companies Web service to get a list of all companies.

Operation

Use the following operation to get a list of companies:

```
GET /companies
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/companies

XML Response

The following is an example of the XML response:

```
<companyList>
  <count>1</count>
  <company>
  <companyId>Dutch Home Insurance</companyId>
  <companyName>Dutch Home Insurance Corporation</companyName>
  <taxId>003</taxId>
  <address1>UNIV BLVD</address1>
  <city>DALLAS</city>
  <state>TEXAS</state>
  <country>USA</country>
  <postalCode>78042</postalCode>
  </company>
  </companyList>
```

Example of Creating a New Company and Primary Contact

This example shows you how to use the companies Web service to create a new company and add a primary contact for initial enrollment.

Operation

Use the following operation to update a company's information:

```
POST /companies/
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/companies

Sample JSON Input

The following is an example of JSON input:

```
{
"companyId":"Company3",
"displayName":"Company3 Mobile Corporation",
"corpAccountNumber":"012",
"taxId":"022",
"street":"24 Tuvell St",
"city":"Portland",
"state":"CA",
"country":"USA",
"postalCode":"92901",
"primaryContactUsername":"newUserABC",
"primaryContactFirstname":"First",
"primaryContactLastname":"Last",
"primaryContactEmail":"Name@example.com",
"startDate":{
"pattern":"MM/dd/yyyy",
"value":"06/12/2015"
}
}
```

Response Message

The following is an example of the response message:

Company successfully created

Visit <link>http://hostname.example.com:7017/rs/api/v2/companies/Company3</link>

Example of Creating a Company and a Billing Hierarchy

This example shows you how to use the companies Web service to create a company and a billing hierarchy for the company.

Operation

Use the following operation to create a company and a billing hierarchy:

```
POST /companies
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/companies

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required.
- Corresponding fields cannot be updated.

Sample XML Input

The following is an example of XML input:

```
<companyInfo>
  <companyId>Dutch Home Insurance</companyId>
  <companyName>Dutch Home Insurance Corporation</companyName>
  <taxId>…</taxId>
  <address1>…</address1>
  <city>…</city>
  <state>…</state>
  <country>…</country>
  <postalCode>…</postalCode>
</companyInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"companyId": "Dutch Home Insurance",
"companyName": "Dutch Home Insurance Corporation",
"taxId": "…",
"address1": "…",
"city": "…",
"state": "…",
"country": "…",
"postalCode": "…"
}
```

Response Message

The following is an example of the response message:

Company Create Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/companies/Dutch%20Home%20Insurance</link>

Example of Getting a Company's Information

This example shows you how to use the companies Web service to get information about a company.

Operation

Use the following operation to get a company's information:

```
GET /companies/{company_id}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/companies/Dutch Home Insurance

XML Response

The following is an example of the XML response:

```
<companyInfo>
  <companyId>Dutch Home Insurance</companyId>
  <companyName>Dutch Home Insurance Corporation</companyName>
  <taxId>…</taxId>
  <address1>…</address1>
  <city>…</city>
  <state>…</state>
  <country>…</country>
  <postalCode>…</postalCode>
</companyInfo>
```

Example of Updating a Company's Information

This example shows you how to use the companies Web service to update a company's information.

Operation

Use the following operation to update a company's information:

```
PUT /companies/{companyId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/companies/Dutch Home Insurance

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- If tags are not present, then the corresponding fields are not updated.
- If tags are present with an empty value, then the corresponding fields are set to empty if a null value is allowed.

Sample XML Input

The following is an example of XML input:

```
<companyInfo>
  <companyId>Dutch Home Insurance</companyId>
  <companyName>Dutch Home Insurance Corporation UPDATED</companyName>
  <taxId>…</taxId>
  <address1>…</address1>
  <city>…</city>
  <state>…</state>
  <country>…</country>
  <postalCode>…</postalCode>
</companyInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
   "companyId": "Dutch Home Insurance",
   "companyName": "Dutch Home Insurance Corporation UPDATED",
   "taxId": "…",
   "address1": "…",
   "city": "…",
   "state": "…",
   "country": "…",
   "postalCode": "…"
}
```

Response Message

The following is an example of the response message:

Company Update Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/companies/Dutch%20Home%20Insurance</link>

Examples of the Disputes Web Service

This topic includes the following examples of how to use the disputes Web service to handle dispute information in Oracle Billing Insight:

Example of Getting a List of Disputes for a Service Transaction

This example shows you how to use the disputes Web service to get a list of the disputes created for a service transaction.

Operation

Use the following operation to get list of disputes:

```
GET /disputes
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/disputes?itemType=transaction&itemId=4710

XML Response

The following is an example of the XML response:

```
<disputeList>
  <links>
     <rel>self</rel>
     <href>http://myhost.example.com:7017/rs/api/v2/
     disputes?itemType=transaction&amp;itemId=4710</href>
  </links>
     <count>1</count>
     <items>
  <links>
     <rel>self</rel>
     <href>http://myhost.example.com:7017/rs/api/v2/disputes/100002</href>
     </links>
  <disputeId>100002</disputeId>
  <disputeRefNumber>1000017</disputeRefNumber>
  </items>
</disputeList>
```

Example of Getting a List of Dispute Comments

This example shows you how to use the disputes Web service to get a list of the disputes created for a service transaction.

Operation

Use the following operation to get list of disputes:

```
GET /disputes/{disputeId}/comments
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/disputes/90004/comments

XML Response

The following is an example of the XML response:

```
<commentsInfo>
  <links>
     <rel>self</rel>
     <href>http:// myhost.example.com:7017/rs/api/v2/disputes/90004/comments
     </href>
  </links>
  <count>2</count>
<commentInfo>
     <disputeCommentId>100000</disputeCommentId>
     <createdDate pattern="MM/dd/yyyy">08/26/2015</createdDate>
     <createdBy>ftown</createdBy>
     <statusCode>process</statusCode>
     <comment>asdfasf</comment>
</commentInfo>
<commentInfo>
     <disputeCommentId>90005</disputeCommentId>
     <createdDate pattern="MM/dd/yyyy">08/18/2015</createdDate>
     <createdBy>ftown</createdBy>
     <statusCode>open</statusCode>
</commentInfo>
</commentsInfo>
```

Example of Updating a Dispute

This example shows you how to use the disputes Web service to update a dispute.

Operation

Use the following operation to update a dispute:

```
PUT /disputes/{disputeId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/disputes/90004

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- If tags are not present, then the corresponding fields are not updated.
- If tags are present with an empty value, then the corresponding fields are set to empty if a null value is allowed.

Sample XML Input

The following is an example of XML input:

```
<disputeInfo currency="USD">
  <disputeStatus>credit</disputeStatus>
</disputeInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"disputeStatus": "credit"
}
```

Response Message

The following is an example of the response message:

Dispute Update Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/disputes/90004</link>

Example of Creating a Dispute

This example shows you how to use the disputes Web service to create a dispute.

Operation

Use the following operation to create a dispute comment:

```
POST /disputes
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/disputes

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- If tags are not present, then the corresponding fields are not updated.
- If tags are present with an empty value, then the corresponding fields are set to empty if a null value is allowed.
- The amount uses the localized format and is parsed in the locale that is passed in the Accept Language HTML attribute.

Sample XML Input

The following is an example of XML input:

```
<?xml version="1.0" encoding="UTF-8"?>
<disputeInfo currency="USD">
  <accountNumber>AC-B2B-123361-01</accountNumber>
  <statementTxnDate pattern="MM/dd/yyyy">08/28/2015</statementTxnDate>
  <disputeAmount>34</disputeAmount>
  <createUser>ftown</createUser>
  <disputeReason>1</disputeReason>
  <disputeStatus>pending</disputeStatus>
  <itemIDs>4705</itemIDs>
<flexibleField1>flexibleField1</flexibleField1>
<flexibleField2>flexibleField2</flexibleField2>
<flexibleField3>flexibleField3</flexibleField3>
<flexibleField4>flexibleField4</flexibleField4>
<flexibleField5>flexibleField5</flexibleField5>
<flexibleFieldNumber1>22</flexibleFieldNumber1>
<flexibleFieldNumber2>23</flexibleFieldNumber2>
<flexibleFieldNumber3>24</flexibleFieldNumber3>
<flexibleFieldDate1 pattern="MM/dd/yyyy">08/28/2015</flexibleFieldDate1>
<flexibleFieldDate2 pattern="MM/dd/yyyy">08/28/2015</flexibleFieldDate2>
<flexibleFieldDate3 pattern="MM/dd/yyyy">08/28/2015</flexibleFieldDate3>
</disputeInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"currency": "USD",
"accountNumber": "AC-B2B-123361-01",
  "statementTxnDate": {
     "pattern": "MM/dd/yyyy",
     "value": "08/26/2015"
},
"disputeAmount": "34",
"createUser": "ftown",
```

```
"disputeReason": "1",
"disputeStatus": "open",
"itemType": "transaction",
"itemIDs": "4705",
"flexibleField1": "flexibleField1",
"flexibleField2": "flexibleField2",
"flexibleField3": "flexibleField3",
"flexibleField4": "flexibleField4",
"flexibleField5": "flexibleField5",
"flexibleFieldNumber1": 22,
"flexibleFieldNumber2": 23,
"flexibleFieldNumber3": 24,
"flexibleFieldDate1": {
     "pattern": "MM/dd/yyyy",
      "value": "08/28/2015"
},
"flexibleFieldDate2": {
     "pattern": "MM/dd/yyyy",
     "value": "08/28/2015"
},
"flexibleFieldDate3": {
     "pattern": "MM/dd/yyyy",
     "value": "08/28/2015"
}
}
```

Response Message

The following is an example of the response message:

Dispute Create Success

Visit <link>http://10.240.66.69:7050/rs/api/v2/disputes/100022</link>

Example of Creating a Dispute Comment

This example shows you how to use the disputes Web service to create a dispute comment.

Operation

Use the following operation to create a dispute comment:

```
POST /disputes/{disputeId}/comments
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/disputes/9004

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- If tags are not present, then the corresponding fields are not updated.
- If tags are present with an empty value, then the corresponding fields are set to empty if a null value is allowed.

Sample XML Input

The following is an example of XML input:

```
<disputeCommentInfo>
  <createdDate pattern="MM/dd/yyyy">08/29/2015</createdDate>
  <createdBy>ftown</createdBy>
  <statusCode>reject</statusCode>
  <comment>the dispute has been rejected</comment>
  </disputeCommentInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"disputeCommentId": 100000,
"createdDate": {
  "pattern": "MM/dd/yyyy",
  "value": "08/26/2015"
},
"createdBy": "ftown",
"statusCode": "reject",
"comment": "the dispute has been rejected"

}
```

Response Message

The following is an example of the response message:

Dispute Comment Create Success

Visit <link> http://myhost.example.com:7017/rs/api/v2/disputes/90004/comments</link>

Examples of the Hierarchy Web Service

This topic includes the following examples of how to use the hierarchy Web service to handle hierarchy information in Oracle Billing Insight:

Example of Getting a List of Hierarchies

This example shows you how to use the hierarchy Web service to get a list of hierarchies that the authenticated user can access.

Operation

Use the following operation to get a list of hierarchies for a user:

```
GET /hierarchies
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies

XML Response

The following is an example of the XML response:

```
<hierarchyList>
  <count>2</count>
  <hierarchy>
     <hierarchyId>91295</hierarchyId>
     <hierName>Billing Hierarchy</hierName>
     <hierType>BILLING</hierType>
     <company>Dutch Home Insurance</company>
  </hierarchy>
  <hierarchy>
     <hierarchyId>102963</hierarchyId>
     <hierName>Hierarchy43358</hierName>
     <hierType>BUSINESS</hierType>
     <company>Dutch Home Insurance</company>
  </hierarchy>
</hierarchyList>
```

Example of Getting a List of Billing Hierarchies

This example shows you how to use the hierarchy Web service to get a list of billing hierarchies, including IDs and names, that the authenticated user can access.

Operation

Use the following operation to get a list of billing hierarchies for a user:

```
/hierarchies?hierarchyType=BILLING
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies?hierarchyType=BILLING

XML Response

The following is an example of the XML response:

```
<hierarchyList>
  <count>1</count>
  <hierarchy>
     <hierarchyId>91295</hierarchyId>
     <hierName>Billing Hierarchy</hierName>
     <hierType>BILLING</hierType>
     <company>Dutch Home Insurance</company>
  </hierarchy>
</hierarchyList>
```

Example of Getting a List of Business Hierarchies

This example shows you how to use the hierarchy Web service to get a list of business hierarchies, including IDs and names, that the authenticated user can access.

Operation

Use the following operation to get a list of business hierarchies for a user:

```
GET /hierarchies?hierarchyType=BUSINESS
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies?hierarchyType=BUSINESS

XML Response

The following is an example of the XML response:

```
<hierarchyList>
  count>1</count>
  <hierarchy>
     <hierarchyId>102963</hierarchyId>
     <hierName>Hierarchy43358</hierName>
     <hierType>BUSINESS</hierType>
     <company>Dutch Home Insurance</company>
  </hierarchy>
</hierarchyList>
```

Example of Getting a List of Hierarchies by Type

This example shows you how to use the hierarchy Web service to get a list of hierarchies by hierarchy type, that the authenticated user can access.

Operation

Use the following operation to get a list of hierarchies by type, for a user:

GET /hierarchies/{hierarchyID/{hierarchyId}/nodes/{nodeId}

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies?hierarchyType=BILLING</href>

XML Response

The following is an example of the XML response:

```
<hierarchyList>
  <links>
     <rel>self</rel>
     <href>http://hostname.example.com:7017/rs/api/v2
     /hierarchies?hierarchyType=BILLING</href>
  </links>
  <count>1</count>
  <items>
     <links>
        <rel>rootNodes</rel>
        <href>http://hostname.example.com:7017/rs/api/v2/hierarchies
        /8000/myRootNodes</href>
     </links>
     <links>
        <rel>self</rel>
        <href>http://hostname.example.com:7017/rs/api/v2/hierarchies/8000</href>
     </links>
     <hierarchyId>8000</hierarchyId>
     <hierName>Billing Hierarchy</hierName>
     <displayName>hierarchy.label.name.billingHierarchy</displayName>
     <description>Cuelle Mobile Corporation</description>
     <hierType>BILLING</hierType>
     <company>CUELLE</company>
  </items>
</hierarchyList>
```

Example of Getting a List of Hierarchy Root Nodes

This example shows you how to use the hierarchy Web service to return a list of root nodes of the hierarchy that the authenticated user can access, for the latest period.

Operation

Use the following operation to get a list of hierarchy root nodes for a user:

```
GET /hierarchies/{hierarchyId}/myRootNodes
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/myRootNodes

XML Response

The following is an example of the XML response:

```
<hierarchyNodeList>
  <count>1</count>
  <hierarchyNode>
  <nodeId>91296</nodeId>
  <hierarchyId>91295</hierarchyId>
```

```
  <nodeName>Dutch Home Insurance</nodeName>
  <linkTargetType>edx:omf:company:</linkTargetType>
  <linkTargetURI>edx:omf:company:Dutch Home Insurance</linkTargetURI>
  <linkTargetExternalKey>Dutch Home Insurance</linkTargetExternalKey>
  </hierarchyNode>
</hierarchyNodeList>
```

Example of Getting a List of Link Targets for a Node

This example shows you how to use the hierarchy Web service to return a list of link targets for a particular node.

Operation

Use the following operation to get a list of link targets for a particular node:

```
GET /hierarchies/{hierarchyId}/linkTargetTypes
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2//hierarchies/91295/linkTargetTypes

XML Response

The following is an example of the XML response:

```
<linkTargetTypeList>
  <count>3</count>
  <linkTargetType><linkTargetType>edx:amf:billingaccount:
  </linkTargetType></linkTargetType>
  <linkTargetType><linkTargetType>edx:omf:company:</linkTargetType>
  </linkTargetType>
  <linkTargetType><linkTargetType>edx:omf:serviceagreement:</linkTargetType>
  </ linkTargetType>
</linkTargetTypeList>
```

Example of Getting a List of Hierarchy Service Agreements

This example shows you how to use the hierarchy Web service to returns a list of service agreements for the hierarchy that the authenticated user can access.

Operation

Use the following operation to get a list of hierarchy service agreements for a user:

```
GET /hierarchies/{hierarchyId}/nodes/{nodeId}/
linkTargets?linkTargetType=serviceAgreement
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/nodes/91296?linkTargets/linkTargetType=serviceAgreement

XML Response

The following is an example of the XML response:

```
<hierarchyNodeList>
  <count>2</count>
  <hierarchyNode>
     <nodeId>100025</nodeId>
     <hierarchyId>91295</hierarchyId>
     <parentId>91296</parentId>
     <nodeName>7836380B2B1</nodeName>
     <linkTargetType>edx:amf:billingaccount:</linkTargetType>
     <linkTargetURI>edx:amf:billingaccount:BS1|7836380B2B1</linkTargetURI>
     <linkTargetExternalKey>BS1|7836380B2B1</linkTargetExternalKey>
  </hierarchyNode>
  <hierarchyNode>
     <nodeId>91298</nodeId>
     <hierarchyId>91295</hierarchyId>
     <parentId>91296</parentId>
     <nodeName>7836380DEMO1</nodeName>
     <linkTargetType>edx:amf:billingaccount:</linkTargetType>
     <linkTargetURI>edx:amf:billingaccount:BS1|7836380DEMO1</linkTargetURI>
     <linkTargetExternalKey>BS1|7836380DEMO1</linkTargetExternalKey>
  </hierarchyNode>
</hierarchyNodeList>
```

Example of Getting a List of Hierarchy Groups

This example shows you how to use the hierarchy Web service to returns a list of groups for the hierarchy that the authenticated user can access.

Operation

Use the following operation to get a list of hierarchy groups for a user:

```
GET hierarchies/103041/nodes/103042/linkTargets?linkTargetType=groups
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/103041/nodes/103042/linkTarget/linkTargetType/groups

XML Response

The following is an example of the XML response:

```
<hierarchyNodeList>
  <count>2</count>
  <hierarchyNode>
     <nodeId>103042</nodeId>
     <hierarchyId>103041</hierarchyId>
     <nodeName>Del24410</nodeName>
     <linkTargetType>edx:hierarchy:folder:</linkTargetType>
     <linkTargetURI>edx:hierarchy:folder:Dutch Home Insurance_Del24410
     </ linkTargetURI>
     <linkTargetExternalKey>Dutch Home Insurance_Del24410</linkTargetExternalKey>
  </hierarchyNode>
  <hierarchyNode>
     <nodeId>103043</nodeId>
     <hierarchyId>103041</hierarchyId>
     <parentId>103042</parentId>
     <nodeName>GName</nodeName>
     <linkTargetType>edx:hierarchy:folder:</linkTargetType>
     <linkTargetURI>edx:hierarchy:folder:GID1</linkTargetURI>
     <linkTargetExternalKey>GID1</linkTargetExternalKey>
  </hierarchyNode>
</hierarchyNodeList>
```

Example of Getting a List of Child Nodes

This example shows you how to use the hierarchy Web service to returns a list of child nodes for a particular node, and a node index.

Operation

Use the following operation to get a list of child nodes for a node:

```
GET /hierarchies/{hierarchyId}/nodes/{nodeid}/children
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/103041/nodes/103042/children

XML Response

The following is an example of the XML response:

```
<hierarchyNodeList>
  <count>2</count>
  <hierarchyNode>
     <nodeId>103042</nodeId>
     <hierarchyId>103041</hierarchyId>
     <nodeName>Del24410</nodeName>
     <linkTargetType>edx:hierarchy:folder:</linkTargetType>
     <linkTargetURI>edx:hierarchy:folder:Dutch Home Insurance_Del24410
     </linkTargetURI>
     <linkTargetExternalKey>Dutch Home Insurance_Del24410
     </linkTargetExternalKey>
  </hierarchyNode>
  <hierarchyNode>
     <nodeId>103043</nodeId>
     <hierarchyId>103041</hierarchyId>
     <parentId>103042</parentId>
     <nodeName>GName</nodeName>
     <linkTargetType>edx:hierarchy:folder:</linkTargetType>
     <linkTargetURI>edx:hierarchy:folder:GID1</linkTargetURI>
     <linkTargetExternalKey>GID1</linkTargetExternalKey>
  </hierarchyNode>
</hierarchyNodeList>
```

Example of Getting a List of Link Target Types for a Node

This example shows you how to use the hierarchy Web service to returns a list of link targets for a particular node.

Operation

Use the following operation to get a list of link target types for a particular node:

```
GET /hierarchies/{hierarchyId}/nodes/{nodeid}/linkTargets
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/nodes/91296/edx:amf:billingaccount/linkTargets

XML Response

The following is an example of the XML response:

```
<hierarchyNodeList>
  <count>2</count>
  <hierarchyNode>
     <nodeId>100025</nodeId>
     <hierarchyId>91295</hierarchyId>
     <parentId>91296</parentId>
     <nodeName>7836380B2B1</nodeName>
```

```
      <linkTargetType>edx:amf:billingaccount:</linkTargetType>
      <linkTargetURI>edx:amf:billingaccount:BS1|7836380B2B1</linkTargetURI>
      <linkTargetExternalKey>BS1|7836380B2B1</linkTargetExternalKey>
   </hierarchyNode>
   <hierarchyNode>
      <nodeId>91298</nodeId>
      <hierarchyId>91295</hierarchyId>
      <parentId>91296</parentId>
      <nodeName>7836380DEMO1</nodeName>
      <linkTargetType>edx:amf:billingaccount:</linkTargetType>
      <linkTargetURI>edx:amf:billingaccount:BS1|7836380DEMO1</linkTargetURI>
      <linkTargetExternalKey>BS1|7836380DEMO1</linkTargetExternalKey>
   </hierarchyNode>
   </hierarchyNodeList>
```

Example of Getting a List of Users Assigned to a Node

This example shows you how to use the hierarchy Web service to returns a list of users assigned to a node.

Operation

Use the following operation to get a list of users assigned to a node:

```
GET /hierarchies/{hierarchyId}/nodes/{nodeId}/users?status=assigned
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/nodes/91296/users?status=assigned

XML Response

The following is an example of the XML response:

```
<userList>
   <count>4</count>
   <user>
      <userId>B2BMANAGER</userId>
      <firstName>BBMANAGER</firstName>
      <lastName>BBMANAGER</lastName>
   </user>
   <user>
      <userId>B2B5420130517</userId>
      <firstName>Subscriber</firstName>
      <lastName>User</lastName>
   </user>
   <user>
      <userId>B2B44201313304</userId>
      <firstName>Business</firstName>
```

```
      <lastName>Changed</lastName>
    </user>
    <user>
      <userId>B2B44201315252</userId>
      <firstName>Business</firstName>
      <lastName>User</lastName>
    </user>
    <user>
      <userId>B2B442013152145</userId>
      <firstName>Business</firstName>
      <lastName>User</lastName>
    </user>
  </userList>
```

Example of Getting a List of Users Unassigned to a Node

This example shows you how to use the hierarchy Web service to return a list of users unassigned to a node.

Operation

Use the following operation to get a list of users unassigned to a nodes:

```
GET /hierarchies/{hierarchyId}/nodes/{nodeId}/users?status=unassigned
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/nodes/91296/users?status=unassigned

XML Response

The following is an example of the XML response:

```
<userList>
  <count>3</count>
  <user>
    <userId>unohu</userId>
    <firstName>Uno</firstName>
    <lastName>Hu</lastName>
  </user>
  <user>
    <userId>fedorastuart</userId>
    <firstName>Fedora</firstName>
    <lastName>Stuart</lastName>
  </user>
  <user>
    <userId>mdhoni</userId>
    <firstName>Mahendra</firstName>
```

```
      <lastName>Dhoni</lastName>
    </user>
  </userList>
```

Example of Getting a List of Users Authorized to a Node

This example shows you how to use the hierarchy Web service to returns a list of users authorized to a node.

Operation

Use the following operation to get a list of users authorized to a node:

```
GET /hierarchies/{hierarchyId}/nodes/{nodeId}/users?status=authorized
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/nodes/91296/users?status=authorized

XML Response

The following is an example of the XML response:

```
<userList>
  <count>3</count>
  <user>
     <userId>ftown</userId>
     <firstName>Frank</firstName>
     <lastName>Town</lastName>
     </user>
  <user>
     <userId>B2B57437</userId>
     <firstName>Business</firstName>
     <lastName>User</lastName>
  </user>
  <user>
     <userId>B2B442013134942</userId>
     <firstName>Business</firstName>
     <lastName>User</lastName>
  </user>
</userList>
```

Example of Getting a List of Users Unauthorized to a Node

This example shows you how to use the hierarchy Web service to returns a list of users unauthorized to a node.

Operation

Use the following operation to get a list of users unauthorized to a node:

```
GET /hierarchies/{hierarchyId}/nodes/{nodeId}/users?status=unauthorized
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/nodes/91296/users?status=unauthorized

XML Response

The following is an example of the XML response:

```
<userList>
  <count>3</count>
  <user>
     <userId>gracerichard</userId>
     <firstName>Grace</firstName>
     <lastName>Richard</lastName>
  </user>
  <user>
     <userId>lgreen</userId>
     <firstName>Lisa</firstName>
     <lastName>Green</lastName>
  </user>
  <user>
     <userId>klaracey</userId>
     <firstName>Kevin</firstName>
     <lastName>Laracey</lastName>
  </user>
</userList>
```

Example of Getting a Node ID

This example shows you how to use the hierarchy Web service to returns a node ID using the unique link target URI, without the hierarchy node type.

Operation

Use the following operation to get a node ID using the unique link target URI:

```
GET /hierarchies/{hierarchyId}/linkTargets/{linkTargetURI}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/hierarchies/91295/linkTargets/edx:omf:company:Dutch%20Home%20Insurance

XML Response

The following is an example of the XML response:

```
<hierarchyNodeInfo>
  <nodeId>91296</nodeId>
  <hierarchyId>91295</hierarchyId>
  <nodeName>Dutch Home Insurance</nodeName>
  <linkTargetType>edx:omf:company:</linkTargetType>
  <linkTargetURI>edx:omf:company:Dutch Home Insurance</linkTargetURI>
  <linkTargetExternalKey>Dutch Home Insurance</linkTargetExternalKey>
</hierarchyNodeInfo>
```

Examples of the Notifications Web Service

TThis topic includes the following examples of how to use the notifications Web service with Oracle Billing Insight:

- "Example of Getting a List of Notification Choices" on page 87
- "Example of Updating a Notification Choice" on page 91

Example of Getting a List of Notification Choices

This example shows you how to use the notification Web service to get a list of notification choices.

Operation

Use the following operation to get a node ID using the unique link target URI:

```
GET /notifications
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/notifications

XML Response

The following is an example of the XML response:

```
<notificationChoices>
  <choice>
     <id>newStmtAvailable</id>
     <description>My bill is ready for viewing</description>
     <email>
        <title>Send email if my bill is ready for viewing</title>
        <selected>true</selected>
     </email>
     <sms>
        <title>Send SMS if my bill is ready for viewing</title>
        <selected>false</selected>
     </sms>
     <enableSMS>Y</enableSMS>
  </choice>
  <choice>
     <id>newStmtAvailablePDF</id>
     <description>My bill summary is ready via PDF</description>
     <email>
        <title>Send email if my bill summary is ready via PDF</title>
        <selected>false</selected>
     </email>
     <sms>
        <title>Send SMS if my bill summary is ready via PDF</title>
        <selected>false</selected>
     </sms>
     <enableSMS>Y</enableSMS>
  </choice>
  <choice>
     <id>paymentAccountCreateDelete</id>
     <description>My payment account has been created,deleted or updated
     </description>
     <email>
        <title>Send email if my payment account has been created,deleted or updated
        </title>
        <selected>true</selected>
     </email>
     <sms>
        <title>Send SMS if my payment account has been created,deleted or updated
        </title>
        <selected>false</selected>
     </sms>
        <enableSMS>Y</enableSMS>
  </choice>
  <choice>
     <id>paymentDueXDays</id>
     <description>My payment is due in</description>
     <email>
        <title>Send email if my payment is due in</title>
        <selected>false</selected>
     </email>
     <sms>
```

```
      <title>Send SMS if my payment is due in</title>
      <selected>false</selected>
   </sms>
      <parameters>
      <parameter>
         <name>numDaysBeforeDueDate</name>
         <value>5</value>
      </parameter>
      </parameters>
   <enableSMS>Y</enableSMS>
</choice>
<choice>
   <id>instantPayment</id>
   <description>My payment was submitted, modified or deleted</description>
   <email>
      <title>Send email if my payment was submitted</title>
      <selected>false</selected>
   </email>
   <sms>
      <title>Send SMS if my payment was submitted</title>
      <selected>false</selected>
   </sms>
   <enableSMS>Y</enableSMS>
</choice>
<choice>
   <id>recurPaymentSetupModify</id>
   <description>My recurring payment is setup, modified or deleted</description>
   <email>
      <title>Send email if my recurring payment is setup or modified</title>
      <selected>false</selected>
   </email>
   <sms>
      <title>Send SMS if my recurring payment is setup or modified</title>
      <selected>false</selected>
   </sms>
   <enableSMS>Y</enableSMS>
</choice>
<choice>
   <id>recurPaymentThresholdExceed</id>
   <description>My recurring payment is less than the total amount due
   (threshold exceeded)</description>
   <email>
      <title>Send email if my recurring payment is less than the total amount due
      (threshold exceeded)</title>
      <selected>false</selected>
   </email>
   <sms>
      <title>Send SMS if my recurring payment is less than the total amount due
      (threshold exceeded)</title>
      <selected>false</selected>
   </sms>
   <enableSMS>Y</enableSMS>
</choice>
<choice>
```

```
      <id>paymentPosts</id>
      <description>My payment was made successfully</description>
      <email>
         <title>Send email if my payment was made successfully</title>
         <selected>false</selected>
      </email>
      <sms>
         <title>Send SMS if my payment was made successfully</title>
         <selected>false</selected>
      </sms>
      <enableSMS>Y</enableSMS>
   </choice>
   <choice>
      <id>paymentFailed</id>
      <description>My payment failed</description>
      <email>
         <title>Send email if my payment failed</title>
         <selected>false</selected>
      </email>
      <sms>
         <title>Send SMS if my payment failed</title>
         <selected>false</selected>
      </sms>
      <enableSMS>Y</enableSMS>
   </choice>
   <choice>
      <id>creditCardExpiration</id>
      <description>My credit card is about to expire</description>
      <email>
         <title>Send email if my credit card is about to expire</title>
         <selected>false</selected>
      </email>
         <sms>
            <title>Send SMS if my credit card is about to expire</title>
            <selected>false</selected>
         </sms>
      <enableSMS>Y</enableSMS>
   </choice>
   <choice>
      <id>batchReportReady</id>
      <description>My batch report is ready</description>
      <email>
         <title>Send email if my batch report is ready</title>
         <selected>false</selected>
      </email>
      <sms>
         <title>Send SMS if my batch report is ready</title>
         <selected>false</selected>
      </sms>
      <enableSMS>Y</enableSMS>
   </choice>
   <choice>
      <id>batchReportRequestExpiry</id>
      <description>My batch report request is expired</description>
```

```
      <email>
         <title>Send email if my batch report request is expired</title>
         <selected>false</selected>
      </email>
      <sms>
         <title>Send SMS if my batch report is expired</title>
         <selected>false</selected>
      </sms>
      <enableSMS>Y</enableSMS>
   </choice>
</notificationChoices>
```

Example of Updating a Notification Choice

This example shows you how to use the notification Web service to get a list of notification choices.

Operation

Use the following operation to update a notification choice:

```
PUT /notifications
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/notifications

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- If tags are not present, then the corresponding fields are not updated.
- If tags are present with an empty value, then the corresponding fields are set to empty if a null value is allowed.

Sample XML Input

The following is an example of XML input:

```
<notificationChoices>
<choice>
  <id>paymentDueXDays</id>
  <email>
     <selected>true</selected>
  </email>
  <sms>
     <selected>true</selected>
  </sms>
  <parameters>
```

```
      <parameter>
         <name>numDaysBeforeDueDate</name>
         <value>6</value>
      </parameter>
      </parameters>
</choice>
</notificationChoices>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"choice": [
  {
  "id": "paymentDueXDays",
  "email": {
     "selected": true
     },
     "sms": {
        "selected": true
      },
  "parameters": {
     "parameter": [
        "name": "numDaysBeforeDueDate",
        "value": "6"
   }
]

}
]
}
```

Response Message

Update Notification Choices SUCCESS

Examples of the Service Agreements Web Service

This topic includes the following examples of how to use the service_agreements Web service to handle service agreement information in Oracle Billing Insight:

Example of Getting a List of Service Agreements

This example shows you how to use the service_agreements Web service to get a list of all service agreements for the current period. The list is based on the billing hierarchy for B2B users and the account-service agreement relationship for B2C users.

Operation

Use the following operation to get a list of service agreements:

```
GET /serviceAgreements
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/serviceAgreements

XML Response

The following is an example of the XML response:

```
<serviceAgreementList>
  <count>2</count>
  <serviceAgreement>
     <serviceAgreementId>BS1|7836380B2B1|878-457-B2B1</serviceAgreementId>
     <billerId>BS1</billerId>
     <accountNumber>7836380B2B1</accountNumber>
     <saNumber>878-457-B2B1</saNumber>
  </serviceAgreement>
  <serviceAgreement>
     <serviceAgreementId>BS1|7836380DEMO1|878-342-DEM1</serviceAgreementId>
     <billerId>BS1</billerId>
     <accountNumber>7836380DEMO1</accountNumber>
     <saNumber>878-342-DEM1</saNumber>
  </serviceAgreement>
</serviceAgreementList>
```

Example of Getting Information for a Service Agreement

This example shows you how to use the service_agreements Web service to get information about a service agreement.

Operation

Use the following operation to get information for a service agreement:

```
GET /serviceAgreements/{serviceAgreementId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/serviceAgreements/BS1|7836380B2B1|878-457-B2B1

XML Response

The following is an example of the XML response:

```
<serviceAgreementInfo>
  <serviceAgreementId>BS1|7836380B2B1|878-457-B2B1</serviceAgreementId>
  <billerId>BS1</billerId>
  <accountNumber>7836380B2B1</accountNumber>
  <saNumber>878-457-B2B1</saNumber>
</serviceAgreementInfo>
```

Example of Creating a Service Agreement

This example shows you how to use the service_agreements Web service to create a service agreement.

Operation

Use the following operation to create a service agreement:

```
POST /serviceAgreements
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/serviceAgreements

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required.
- If tags are not present, then the corresponding fields are set to the default values.

- The value of serviceType must be in the SERVICE_TYPE_CD in the EDX_RPT_SERVICE_TYPE_DIM table in the Online Analytic Processing (OLAP) database.

Sample XML Input

The following is an example of XML input:

```
<serviceAgreementInfo>
  <serviceAgreementId>BS1|7836380B2B1|878-457-B2B1</serviceAgreementId>
  <billerId>BS1</billerId>
  <accountNumber>7836380B2B1</accountNumber>
  <saNumber>878-457-B2B1</saNumber>
  <serviceType>UNK</serviceType>
  <description>xxx</description>
  <subscriber>xxx</subscriber>
</serviceAgreementInfo>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"billerId":"BS1",
"accountNumber": "AC-B2B-64521A",
"saNumber": "123-111-1101",
"serviceType": "UNK",
"subscriber": "subscriber name",
"startDate":{
"pattern":"MM/dd/yyyy",
"value":"12/12/2014"
},
"zoning":"z",
"description": "desc"
}
```

Response Message

The following is an example of the response message:

Service Agreement Create Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/serviceAgreements/*BS1%7C7836380B2B1%7C878-457-B2B1</link>*

Examples of the Users Web Service

This topic includes the following examples of how to use the users Web service to handle user information in Oracle Billing Insight:

Example of Getting a List of Users

This example shows you how to use the users Web service to get a list of users. The list returned is based on the authenticated user's role.

Operation

Use the following operation to get a list of users:

```
GET /users
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/users

XML Response

The following is an example of the XML response:

```
<userList>
  count>2</count>
  <user>
     <userId>mdhoni</userId>
     <firstName>Mahendra</firstName>
     <lastName>Dhoni</lastName>
     <email>mdhoni@example.com</email>
     <link>/users/mdhoni</link>
  </user>
  <user>
     <userId>fedorastuart</userId>
     <firstName>Fedora</firstName>
     <lastName>Stuart</lastName>
     <email>fedorastuart@example.com</email>
     <link>/users/fedorastuart</link>
  </user>
</userList>
```

Example of Getting a User's Information

This example shows you how to use the users Web service to get information for a user.

Operation

Use the following operation to get a user's information:

```
GET /users/{userId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/users/mdhoni

XML Response

The following is an example of the XML response:

```
<userProfile>
  <userId>mdhoni</userId>
  <role>Manager</role>
  <firstName>Mahendra</firstName>
  <lastName>Dhoni</lastName>
  <addressLine1>j street apt 1900</addressLine1>
  <city>boston</city>
  <state>MA</state>
  <zip>04109</zip>
  <country>USA</country>
  <homePhone>555-111-1884</homePhone>
  <email>mdhoni@example.com</email>
  <paperOn>Y</paperOn>
  <isAccessibleUser>0<isAccessibleUser>
</userProfile>
```

Example of Creating a B2B User

This example shows you how to use the users Web service to create a B2B user.

Operation

Use the following operation to create a B2B user (for SSO, not enrollment):

```
POST /users
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/users

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- If tags are not present, then the corresponding fields are set to the default values.

Sample XML Input

The following is an example of XML input:

```
<userProfile>
  <userId>mdhoni</userId>
  <role>Manager</role>
  <firstName>Mahendra</firstName>
  <lastName>Dhoni</lastName>
  <addressLine1>j street apt 1900</addressLine1>
  <city>boston</city>
  <state>MA</state>
  <zip>04109</zip>
  <country>USA</country>
  <homePhone>555-111-1884</homePhone>
  <email>mdhoni@example.com</email>
  <paperOn>Y</paperOn>
  <language>en_US</language>
  <companyId>Dutch Home Insurance</companyId>
  <userType>b2c</userType>
</userProfile>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"userId":"rsamb2b5",
"role":"Manager",
"firstName":"Rob",
"lastName":"Sunny",
"email":"rs002@example.com",
"language":"en_US",
"companyId":"CUELLE",
"userType":"b2b"
}
```

Response Message

The following is an example of the response message:

User Create Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/users/mdhoni</link>

Example of Creating a B2C User

This example shows you how to use the users Web service to create a B2C user (for SSO, not enrollment).

Operation

```
POST /users
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/users

Sample XML Input Rules

The following are the XML input rules:

- Tags in bold are required, and the corresponding fields cannot be updated.
- If tags are not present, then the corresponding fields will be set to the default values.

Sample XML Input

The following is an example of XML input:

```
<userProfile>
  <userId>rsam</userId>
  <role>User</role>
  <firstName>Rich</firstName>
  <lastName>Sam</lastName>
  <email>mdhoni@example.com</email>
  <language>en_US</language>
  <billerId>BS2</billerId>
  <accountNumber>7836380B2C1</accountNumber>
  <userType>b2c</userType>
</userProfile>
```

Sample JSON Input

The following is an example of JSON input:

```
{
"userId":"rsamb2c2",
"role":"User",
"firstName":"Rob",
"lastName":"Sunny",
"email":"rs002@example.com",
"language":"en_US",
"billerId":"BS2",
"accountNumber":"AC-B2C-7836380-1",
"userType":"b2c"
}
```

Response Message

The following is an example of the response message:

User Create Success

Visit <link>http://myhost.example.com:7017/rs/api/v2/users/ rsam</link>

Examples of the Payments Web Service

This topic includes the following examples of how to use the payments Web service to handle payment information in Oracle Billing Insight:

Example of Posting a Payment Transaction

This example shows you how to use the payments Web service to post a payment transaction.

Operation

Use the following operation to post a payment transaction:

```
POST /payment/transactions
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/payment/transactions

Sample XML Input - Using a Default Payment Account

The following is an example of XML input for posting a payment transaction using a default payment account:

```
<paymentTxn>
  <billerId>BS1</billerId>
  <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
  <paymentAmount>110.50</paymentAmount>
  <paymentScheduledDate
```

```
  pattern="MM/dd/yyyy">02/17/2015</paymentScheduledDate>
  <flexfield1>aaa</flexfield1>
  <flexfield2>bbb</flexfield2>
  <flexfield3>ccc</flexfield3>
  <vertical1>ddd</vertical1>
  <vertical2>eee</vertical2>
  <vertical3>fff</vertical3>
  <vertical4>ggg</vertical4>
</paymentTxn>
```

Sample XML Input - Using an Existing Payment Account

The following is an example of XML input for posting a payment transaction using an existing payment account:

```
<paymentTxn>
  <billerId>BS1</billerId>
  <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
  <pmtAcctId>ccard_4</pmtAcctId>
  <paymentAmount>110.50</paymentAmount>
  <paymentScheduledDate pattern="MM/dd/yyyy">02/18/2015</paymentScheduledDate>
  <flexfield1>aaa</flexfield1>
  <flexfield2>bbb</flexfield2>
  <flexfield3>ccc</flexfield3>
  <vertical1>ddd</vertical1>
  <vertical2>eee</vertical2>
  <vertical3>fff</vertical3>
  <vertical4>ggg</vertical4>
</paymentTxn>
```

Sample XML Input - Using a New Bank Account (Without a Companion Savings Account)

The following is an example of XML input for posting a payment transaction using a new bank account, where there is no companion savings account:

```
<paymentTxn>
  <billerId>BS1</billerId>
  <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
  <paymentAmount>110.50</paymentAmount>
  <paymentScheduledDate pattern="MM/dd/yyyy">02/19/2015</paymentScheduledDate>
  <bankAccount>
     <paymentAccountName>checkAcctNew</paymentAccountName>
     <bankName>checkAcctNew</bankName>
     <bankAccountNumber>45465654</bankAccountNumber>
     <pmtMethod>Checking</pmtMethod>
     <routingNumber>091000022</routingNumber>
  </bankAccount>
  <flexfield1>aaa</flexfield1>
  <flexfield2>bbb</flexfield2>
  <flexfield3>ccc</flexfield3>
  <vertical1>ddd</vertical1>
  <vertical2>eee</vertical2>
```

```
  <vertical3>fff</vertical3>
  <vertical4>ggg</vertical4>
</paymentTxn>
```

Sample XML Input - Using a New Credit Card Account (With a Companion Savings Account)

The following is an example of XML input for posting a payment transaction with a new credit card account, when there is a companion savings account:

```
<paymentTxn>
  <billerId>BS1</billerId>
  <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
  <paymentAmount>110.50</paymentAmount>
  <paymentScheduledDate pattern="MM/dd/yyyy">07/15/2014</paymentScheduledDate>
  <ccardAccount>
     <paymentAccountName>OrbMasterAcct</paymentAccountName>
     <nameOnCard>OrbMaster</nameOnCard>
     <cardNumber>341134113411347</cardNumber>
     <pmtMethod>Amex</pmtMethod>
     <expirationDate pattern="MM/yyyy">12/2015</expirationDate>
     <cvvCode>1234</cvvCode>
     <street>4 Northeastern Blvd</street>
     <aptNumber>add2</aptNumber>
     <city>Salem</city>
     <country>ct</country>
     <state>ss</state>
     <zipCode>03105</zipCode>
  </ccardAccount>
  <isSaveAcct>true</isSaveAcct>
  <flexfield1>aaa</flexfield1>
  <flexfield2>bbb</flexfield2>
  <flexfield3>ccc</flexfield3>
  <vertical1>ddd</vertical1>
  <vertical2>eee</vertical2>
  <vertical3>fff</vertical3>
  <vertical4>ggg</vertical4>
</paymentTxn>
```

Sample JSON Input - Using a Default Payment Account

The following is an example of JSON input for posting a payment transaction using default payment account:

```
{
"billerId": "BS1",
“billingAccountNumber ": "AC-B2B-123361-01",
"paymentAmount": "110.50"
“paymentScheduledDate ": {
  "pattern": "MM/dd/yyyy",
  "value": "08/26/2015"
},
" flexfield1": "aaa",
" flexfield2": "bbb",
" flexfield3": "ccc",
```

```
" vertical1": "ddd",
" vertical2": "eee",
" vertical3": "fff",
" vertical4": "ggg"
}
```

Sample JSON Input - Using an Existing Payment Account

The following is an example of JSON input for posting a payment transaction using an existing payment account:

```
{
"billerId": "BS1",
" billingAccountNumber ": "AC-B2B-123361-01",
"pmtAcctId": "ccard_4",
"paymentAmount": "110.50"
" paymentScheduledDate ": {
"pattern": "MM/dd/yyyy",
"value": "08/26/2015"
},
" flexfield1": "aaa",
" flexfield2": "bbb",
" flexfield3": "ccc",
" vertical1": "ddd",
" vertical2": "eee",
" vertical3": "fff",
" vertical4": "ggg"
}
```

Sample JSON Input - Using a New Bank Account (Without a Companion Savings Account)

The following is an example of JSON input for posting a payment transaction with a new bank account, when there is no companion savings account:

```
{
"billerId": "BS1",
" billingAccountNumber ": "AC-B2B-123361-01",
"pmtAcctId": "ccard_4",
"paymentAmount": "110.50"
  " paymentScheduledDate ": {
  "pattern": "MM/dd/yyyy",
  "value": "08/26/2015"
},
"bankAccount": {
  "paymentAccountName":  "checkAcctNew",
  "bankName": "checkAcctNew",
  "bankAccountNumber": "45465654",
  "pmtMethod": "Checking",
  "routingNumber": "091000022"
},
" flexfield1": "aaa",
" flexfield2": "bbb",
" flexfield3": "ccc",
" vertical1": "ddd",
```

```
" vertical2": "eee",
" vertical3": "fff",
" vertical4": "ggg"
}
```

Sample JSON Input - Using a New Credit Card (With a Companion Savings Account)

The following is an example of JSON input for posting payment transaction with new credit card account and save account:

```
{
  "billerId": "BS1",
  " billingAccountNumber ": "AC-B2B-123361-01",
  "pmtAcctId": "ccard_4",
  "paymentAmount": "110.50"
  " paymentScheduledDate ": {
     "pattern": "MM/dd/yyyy",
     "value": "08/26/2015"
  },
"ccardAccount": {
  "paymentAccountName":  "OrbMasterAcct",
  "nameOnCard": "OrbMaster",
  "cardNumber": "341134113411347",
  "pmtMethod": "Amex",
  "expirationDate": {
     "pattern": "MM/dd/yyyy",
     "value": "08/26/2016"
},
"cvvCode": "1234",
"street": "street",
"aptNumber":  "t100",
"city": "Salem",
"country": "ct",
"state": "ss",
"zipCode": "12345"
},
"isSaveAcct": "ue",
" flexfield1": "aaa",
" flexfield2": "bbb",
" flexfield3": "ccc",
" vertical1": "ddd",
" vertical2": "eee",
" vertical3": "fff",
" vertical4": "ggg"
}
```

Response Message

The following is an example of the response message:

POST PAYMENT TRANSACTION SUCCESS

Visit <link>http://myhost.example.com:7017/rs/api/v2/payment/transactions/100401439318815371</link>

Example of Posting an External Payment Transaction

This example shows you how to use the payments Web service to post a payment transaction made externally from Oracle Billing Insight.

Operation

Use the following operation to post an external payment transaction:

```
POST /payment/transactions/external
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/payment/transactions/external

Sample XML Input

The following is an example of XML input:

```
<paymentExternalTransaction>
  <billerId>BS1</billerId>
  <billingAccountNumber>7836380B2B1</billingAccountNumber>
  <paymentAmount>111</paymentAmount>
  <paymentScheduledDate pattern="MM/dd/yyyy">03/21/2013</paymentScheduledDate>
  <paymentType>Mail In</paymentType>
  <status>paid</status>
  <transactionId>T123</transactionId>
  <flexible1>…</flexible1>
  <flexible2>…</flexible2>
  <flexible3>…</flexible3>
  <flexible4>…</flexible4>
  <flexible5>…</flexible5>
</paymentExternalTransaction>
```

Sample JSON Input

The following is an example of JSON input:

```
{
   "billerId": "BS1",
   "billingAccountNumber": "7836380B2B1",
   "paymentAmount": "111",
   "paymentScheduledDate": {
      "@pattern": "MM/dd/yyyy",
      "#text": "03/21/2013"
   },
   "paymentType": "Mail In",
   "status": "paid",
   "transactionId": "T123",
   "flexible1": "…",
   "flexible2": "…",
   "flexible3": "…",
```

```
    "flexible4": "…",
    "flexible5": "…"
}
```

Response Message

The following is an example of the response message:

POST PAYMENT SUCCESS

Visit <link>http://myhost.example.com:7017/rs/api/v2/payment/transactions/external/T123</link>

Example of Updating an External Payment Transaction

This example shows you how to use the payments Web service to update a payment made externally from Oracle Billing Insight.

Operation

Use the following operation to update an external payment transaction:

```
PUT /payments/transactions/external/{trans_id}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/payment/transactions/external/T123

Sample XML Input

The following is an example of XML input:

```
<paymentExternalTransaction>
<billerId>BS1</billerId>
  <billingAccountNumber>7836380B2B1</billingAccountNumber>
  <paymentAmount>222</paymentAmount>
  <paymentScheduledDate pattern="MM/dd/yyyy">04/21/2013</paymentScheduledDate>
  <paymentType>Mail In</paymentType>
  <status>paid</status>
  <transactionId>T123</transactionId>
  <flexible1>…</flexible1>
  <flexible2>…</flexible2>
  <flexible3>…</flexible3>
  <flexible4>…</flexible4>
  <flexible5>…</flexible5>
</paymentExternalTransaction>
```

Sample JSON Input

The following is an example of JSON input:

```
{
   "billerId": "BS1",
   "billingAccountNumber": "7836380B2B1",
   "paymentAmount": "222",
   "paymentScheduledDate": {
      "@pattern": "MM/dd/yyyy",
      "#text": "04/21/2013"
   },
   "paymentType": "Mail In",
   "status": "paid",
   "transactionId": "T123",
   "flexible1": "…",
   "flexible2": "…",
   "flexible3": "…",
   "flexible4": "…",
   "flexible5": "…"
}
```

Response Message

The following is an example of the response message:

POST PAYMENT SUCCESS

Visit <link>http://myhost.example.com:7017/rs/api/v2/payment/transactions/external/T123</link>

Example of Getting a List of Payment Transactions for a Particular Billing Account

This example shows you how to use the payments Web service to get a list of payment transactions made for a particular billing account.

Operation

Use the following operation to get a list of payment transactions made for a particular billing account:

```
GET /payment/transactions?billingAccountid=BS1|AC-B2B-123361-01
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/payment/transactions?billingAccountid=BS1|AC-B2B-123361-01

Sample XML Response

The following is an example of the XML response:

```
<report>
  <links>
     <rel>self</rel>
     <href>http://localhost:7007/rs/api/v2/payment
     /transactions?billingAccountid=BS1|AC-B2B-123361-01</href>
  </links>
  <paymentTransactionList>
     <countInPage>3</countInPage>
     <header>
        <paymentAmount>USD</paymentAmount>
     </header>
     <items>
        <links>
           <rel>self</rel>
           <href>http://localhost:7007/rs/api/v2/payment/transactions
           /101521440037376420</href>
        </links>
        <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
        <type_disp>Bank Transfer</type_disp>
        <paymentScheduledDate pattern="MM/dd/yyyy">08/20/2015
        </paymentScheduledDate>
        <paymentTransactionDate pattern="MM/dd/yyyy HH:mm:ss">
        </paymentTransactionDate>
        <paymentAmount>216.58</paymentAmount>
        <status_disp>Scheduled</status_disp>
        <paymentType>check</paymentType>
     </items>
     <items>
        <links>
           <rel>self</rel>
           <href>http://localhost:7007/rs/api/v2/payment/transactions
           /101521440037360309</href>
        </links>
        <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
        <type_disp>Bank Transfer</type_disp>
        <paymentScheduledDate pattern="MM/dd/yyyy">08/20/2015
        </paymentScheduledDate>
        <paymentTransactionDate pattern="MM/dd/yyyy HH:mm:ss">
        </paymentTransactionDate>
        <paymentAmount>643.57</paymentAmount>
        <status_disp>Scheduled</status_disp>
        <paymentType>check</paymentType>
     </items>
     <items>
        <links>
           <rel>self</rel>
           <href>http://localhost:7007/rs/api/v2/payment/transactions
           /101521440037367471</href>
        </links>
        <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
        <type_disp>Bank Transfer</type_disp>
        <paymentScheduledDate pattern="MM/dd/yyyy">08/20/2015
        </paymentScheduledDate>
        <paymentTransactionDate pattern="MM/dd/yyyy HH:mm:ss">
```

```
        </paymentTransactionDate>
        <paymentAmount>350.29</paymentAmount>
        <status_disp>Scheduled</status_disp>
        <paymentType>check</paymentType>
      </items>
  </paymentTransactionList>
</report>
```

Example of Getting a List of Scheduled Payment Transactions for a Particular Billing Account

This example shows you how to use the payments Web service to get a list of scheduled payment transactions for a particular billing account.

Operation

Use the following operation to get a list of scheduled payment transactions for a particular billing account:

```
GET /payment/transactions?billingAccountid=BS1|AC-B2B-123361-01&status=2004
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/payment/transactions?billingAccountid=BS1|AC-B2B-123361-01&status=2004&startDate=02/01/2015&endDate=09/09/2015

Sample XML Response

The following is an example of the XML response:

```
<report>
  <links>
     <rel>self</rel>

<href>http://localhost:7007/rs/api/v2/payment/transactions?startDate=
02/01/2015&status=2004&endDate=09/09/2015&billingAccountid=BS1|AC-B2B-123361-
01</href>
  </links>
  <paymentTransactionList>
     <countInPage>2</countInPage>
     <header>
        <paymentAmount>USD</paymentAmount>
     </header>
     <items>
        <links>
           <rel>self</rel>
```

```
            <href>http://localhost:7007/rs/api/v2/payment/transactions
            /101521440037360309</href>
         </links>
         <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
         <type_disp>Bank Transfer</type_disp>
         <paymentScheduledDate pattern="MM/dd/yyyy">08/20/2015
         </paymentScheduledDate>
         <paymentTransactionDate pattern="MM/dd/yyyy HH:mm:ss">
         </paymentTransactionDate>
         <paymentAmount>643.57</paymentAmount>
         <status_disp>Scheduled</status_disp>
         <paymentType>check</paymentType>
      </items>
      <items>
         <links>
            <rel>self</rel>
            <href>http://localhost:7007/rs/api/v2/payment/transactions
            /101521440037376420</href>
         </links>
         <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
         <type_disp>Bank Transfer</type_disp>
         <paymentScheduledDate pattern="MM/dd/yyyy">08/20/2015
         </paymentScheduledDate>
         <paymentTransactionDate pattern="MM/dd/yyyy HH:mm:ss">
         </paymentTransactionDate>
         <paymentAmount>216.58</paymentAmount>
         <status_disp>Scheduled</status_disp>
         <paymentType>check</paymentType>
      </items>
   </paymentTransactionList>
</report>
```

Example of Getting a Specific Payment Transaction

This example shows you how to use the payments Web service to get a specific payment transaction.

Operation

Use the following operation to get a specific payment transaction:

```
GET /payment/transactions/{paymentId}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/payment/transactions/101521440037376420

Sample XML Response

The following is an example of the XML response:

```
paymentTransactionProfile currency="USD">
  <links>
     <rel>self</rel>
     <href>http://localhost:7007/rs/api/v2/payment/transactions
     /101521440037376420</href>
  </links>
  <billerId>BS1</billerId>
  <billingAccountNumber>AC-B2B-123361-01</billingAccountNumber>
  <paymentType>Bank Transfer</paymentType>
  <paymentAmount>216.58</paymentAmount>
  <status>Scheduled</status>
     <paymentScheduledDate pattern="MM/dd/yyyy">08/20/2015</paymentScheduledDate>
     <paymentAccountName>testBank</paymentAccountName>
     <paymentAccountNumber>xxxx2211</paymentAccountNumber>
     <createdBy>Town, Frank</createdBy>
     <paymentCreatedDate pattern="MM/dd/yyyy">08/20/2015</paymentCreatedDate>
     <paymentModifiedDate pattern="MM/dd/yyyy">08/20/2015</paymentModifiedDate>
</paymentTransactionProfile>
```

Example of Getting a Particular External Payment Transaction

This example shows you how to use the payments Web service to get a specific external payment transaction.

Operation

Use the following operation to get a specific external bank payment transaction:

```
GET /payment/transactions/external/{trans_id}
```

Sample URL

The following is a sample URL for this Web service example:

http://myhost.example.com:7017/rs/api/v2/payment/transactions/external/T55555

Sample XML Response

The following is an example of the XML response:

```
<paymentExternalTransaction currency="USD">
  <links>
     <rel>self</rel>
     <href>http://localhost:7007/rs/api/v2/payment/transactions/external/T55555
     </href>
  </links>
  <billerId>BS1</billerId>
  <billingAccountNumber>AC-B2B-64521A</billingAccountNumber>
  <paymentAmount>333</paymentAmount>
  <paymentScheduledDate pattern="MM/dd/yyyy">04/21/2013</paymentScheduledDate>
```

```
    <paymentType>Wire Transfer</paymentType>
    <status>Paid</status>
    <transactionId>T55555</transactionId>
    <flexible1>aa</flexible1>
    <flexible2>bb</flexible2>
    <flexible3>cc</flexible3>
    <flexible4>dd</flexible4>
    <flexible5>ee</flexible5>
  </paymentExternalTransaction>
```

6 Customizing RESTful Resources

This chapter describes how to customize some of the RESTful resources that are available with Web services. It includes the following topics:

Customizing Analytics Resources

Many of Oracle Billing Insight RESTful read APIs are provided through the Oracle Billing Insight reporting engine framework. Each resource is mapped to an Oracle Billing Insight report, in which data sources, SQL, and transformers can be specified in an XML configuration. With this flexible architecture, you can extend the current analytic REST resources if needed.

See the following topics for information about customizing analytics resources:

About the Resource and Report ID Mapping File

All preconfigured resources implemented using the reporting engine are listed in the ws_reportIdMap.properties file, located in the following directory, in which *EDX_HOME* is the directory where you installed Oracle Billing Insight:

- **UNIX.** *EDX_HOME*/config/webservice
- **Windows.** *EDX_HOME*\config\webservice

You can edit the ws_reportIdMap.properties file to customize or extend existing analytics RESTful resources. Each resource entry uses the following format:

```
URI Resource=reportId, {reportId for B2C}
```

URI Resource represents the resource without the input parameter, and reportId is the report identifier specified in the report XML file. If the same URI can be used for both the B2B and B2C applications but with a different report ID, then specify the report ID for the B2B application first, followed by the B2C application. Any additional parameters that you provide must be passed as parameters in the URL. The following report examples are from the report ID mapping property file:

```
/analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/
overview=telco_std_r3,telco_std_b2c_r3

/statements/{statementKey}/accountSummary=statementAccountSummary
```

Resource Mapping Customization

You can use the Oracle Billing Insight reporting engine to define additional reporting XML. For more information on creating additional analytic reports, see *Implementation Guide for Oracle Billing Insight*.

Oracle Billing Insight supports the following types of customization for Web services:

- **Replacing preconfigured REST resources in the property file with your own reports and report IDs.** Edit the ws_reportIdMap.properties file, located in the following directory:
  - **UNIX.** *EDX_HOME*/config/webservice
  - **Windows.** *EDX_HOME*\config\webservice

  For example, you can change the existing resource entries in the property file as shown in the following examples:

  ```
  /analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/
  overview=telco_std_r3,my_b2c_serviceoverview_rpt

  /statements/{statementKey}/accountSummary=my_statementAccountSummary
  ```

- **Adding a resource to an existing resource category.** A resource category is defined in the URI section of a resource entry in the ws_reportIdMap.properties file, between the first and last forward slashes (/). For example, each of the following are categories:

  ```
  /analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/

  /statements/{statementKey}/
  ```

  You can add additional resources to a preconfigured category in the ws_reportIdMap.properties file. The additional mapping entry allows the Oracle Billing Insight REST services framework to invoke the corresponding reports without recompiling the application. For example, you can add the following entry to the property file:

  ```
  /analytics/accounts/{accountId}/serviceAgreements/{serviceNumber}/peaktime-
  calls=my_peaktime_call_report
  ```

Creating Additional Analytic Resources With the Reporting Engine

The Oracle Billing Insight reporting engine supports many Web services for the GET operation. The reporting engine also includes a separate Velocity template, additional report elements, and attributes to create XML content for Web service responses. All Web service responses created using the reporting engine have a root node of <report> …</report>. Some of the reporting elements and attributes are as follows:

- The collectionId attribute of the Transformer element specifies the name of the list, which contains one or more rows of data.
- The name attribute of the rows element specifies the name for each row returned in the result list.
- The id attribute of the column element specifies the attribute name of each row and object returned.
- The rsLink element links from one REST service to another. The link must be appended to the domain and root name to construct a full URL for the accessing additional REST resource. Links are provided to allow drilling down, as well as to navigate from one page result to the next or previous when multiple pages of results are returned.
- The rsPaging element specifies the page size for the REST services call only. If it is not specified, then the whole query result is returned.

About Authentication and Authorization

This topic provides the following information about Oracle Billing Insight user authentication and authorization using Web services:

- "About Authentication With Web Services" on page 115
- "About Authorization With Web Services" on page 116
- "About Protection From Cross-Site Request Forgery" on page 116

About Authentication With Web Services

To invoke Oracle Billing Insight Web services to create, read, update or delete (CRUD) business objects, the caller must be authenticated as a registered user. The Oracle Billing Insight REST services server authenticates REST service users in the same way as users who log in using the Oracle Billing Insight Web application. Once a user is authenticated, the REST services server returns a token to the client. The client must add an HTTP header with the attribute name ebrstoken and add the value of the returned token to each REST services request. For example, in the Jersey client, you can use the WebResource.Builder.header(name,value) method to add the ebrstoken name and the token value to the HTTP header.

The token has an expiration period. The default period is 20 minutes. The default string token has a length of 48. You can optionally change both the string length and duration of the token in the webservice.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/modules/webservice
- **Windows.** *EDX_HOME*\xma\config\modules\webservice

Change the property values in the IWebserviceAuthTokenProvider bean.

By default, Oracle Billing Insight uses its preconfigured authentication provider to authenticate users. You can use a different authentication provider, such as an external system. For information on how to customize Oracle Billing Insight to use a different authentication server, see *Implementation Guide for Oracle Billing Insight*.

About Authorization With Web Services

Oracle Billing Insight REST services server uses the same authorization scheme as the server provided in the Oracle Billing Insight Web application. For example, if a request is made on behalf of a registered user to the /analytics/accounts service, then only the accounts that the user is permitted to access are returned in the response. In addition, the Web service server provides capabilities that are not supported in the Oracle Billing Insight Web application. As a result, the Oracle Billing Insight REST services server provides more authorization rules. For example, using a REST services request, the CSR administrator can create and update companies, accounts, or service agreements.

About Protection From Cross-Site Request Forgery

Oracle Billing Insight uses the server-side request filter, com.sun.jersey.api.container.filter.CsrfProtectionFilter, to protect from a cross-site request forgery (CSRF) attack. The request filter checks for an X-Requested-By header in incoming HTTP requests other than GET, OPTIONS, or HEAD, by default. If the header is not found, then Response.Status.BAD_REQUEST returns.

You must add an X-Requested-By header with an arbitrary value to all HTTP POST, PUT and DELETE requests sent to your REST endpoints.

About Customizing Localization Values

This topic provides the following information about customizing localization using Web services with Oracle Billing Insight:

About the Locale

The locale values of the request and response are specified in the HTML header. In service requests, the locale can be retrieved from the Accept-language attribute. Oracle Billing Insight supports one acceptable language for each request. The following XML shows an example of a request using the Accept-language attribute:

```
POST /rsclient/webservice.action HTTP/1.1
Host: myhost.example.com:7001
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:10.0.7) Gecko/20100101 Firefox/
10.0.7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

In the service responses, the locale value is in the Content-Language attribute, as shown:

```
HTTP/1.1 200ok
Date: Wed, 05 Sep 2012 22:14:15 GMT
Content_Length:533
Content-Type:application/xhtml+xml
Content-Language: en-US
X-Powered-By:Servlet/2.5 JSP/2.1
```

About the Date Format

For a particular locale, when parsing date values in requests or output date values in responses, the pattern used to convert a string to or from the date value is specified in an attribute of the date tag element. For example:

```
<Call_date pattern="MM/dd/yyyy">09/15/2012</Call_date>
```

This date attribute is included in all responses of GET requests.

For all POST and PUT resources, the Oracle Billing Insight server requires the XML payload to specify date pattern in the tag element attribute. The server code can then parse the information accordingly. In the following input XML example, the Accept-Language element is es_ES in the request header:

```
<paymentAccountActivity>
  <billerId>BS1</billerId>
  <accountNumber>7836380B2B1</accountNumber>
  <currBalance>786,00</currBalance>
  <lastPaymentReceivedAmount>120,00</lastPaymentReceivedAmount>
  <lastPaymentReceivedDate pattern="dd/MM/yyyy">13/08/2012<
  /lastPaymentReceivedDate>
</paymentAccountActivity>
```

If the date pattern is not explicitly specified in the element attribute, then the default date pattern is used. The default date pattern for each supported locale is specified in the webservice.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/modules/webservice
- **Windows.** *EDX_HOME*\xma\config\moduleswebservice

In the webservice.xma.xml file, the bean `DatePatternFactory` includes a list of `DatePattern` beans. Each `DatePattern` bean represents a pattern for a supported locale. The locale value must match the value stored in the EDX_SYS_LANG table locale CODE column. The `datePatternMap` contains a map of the pattern keys and values, which lists the possible patterns that can be used for pure Date and Datetime. A *pure* Date or Datetime value does not include a presentation format. The key and value must be consistent with the resource bundle files that are used by the Oracle Billing Insight Web application.

About Numbers and Currencies

Numbers in both requests and responses use the default pattern of the authenticated user's preferred locale. For example, the number 120.00 is displayed in locale en_US and 120,00 in the es_ES locale.

The currency information does not change when the user's locale preference changes. The currency information comes from the statement fact, and this currency information is used in the billing system for the account. The currency code that is displayed in the Web application is included as part of the XML response.

For GET business object resources, the currency code is specified in an attribute of a tag element, for example:

```
<disputeInfo currency="USD">
  <disputeId>90007</disputeId>
  <disputeRefNumber>900067</disputeRefNumber>
  <accountNumber>7836380B2B1</accountNumber>
  <disputeAmount>12.3</disputeAmount>
  <disputeDesc> chris webservice</disputeDesc>
  <disputeReason>1</disputeReason>
  <disputeStatus>open</disputeStatus>
</disputeInfo>
```

For GET report resources, the currency code is specified in the <Header> tag for each column that requires the currency, for example:

```
<header>
<Total>USD</Total>
</header>
```

Outbound Web Services

Oracle Billing Insight outbound Web services can be invoked from the Oracle Billing Insight application. There are two trigger points in the Oracle Billing Insight Self-Service application:

- When a user profile is created or updated
- When a dispute is created or updated

For each of these triggers, there is a Web service connector interface. The methods on the connector interface allow system-integration developers to invoke any operations on the external system, such as updating a user profile to a back-end user management system, or creating a dispute in a CRM system. Configure the implementation classes of these interfaces in the webservice.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/modules/webservice
- **Windows.** *EDX_HOME*\xma\config\moduleswebservice

For example:

```
<bean id="IDisputeWSConnector"
class="com.mycompany.ebilling.myDisputeWSConnector"></bean>
<bean id="IUserProfileWSConnector" class="com.mycompany.myUserProfileWSConnector">
</bean>
```

For more information, see the reference API Javadoc for the following:

- com.edocs.common.api.webservice.connector.IUserProfileWSConnector
- com.edocs.common.api.webservice.connector.IDisputeWSConnector

Index