

Oracle Enterprise Repository

Subscriptions API

Overview

The subscriptions API provides a mechanism for users to manage the assets to which a user is subscribed. Subscription, in this context, refers specifically to *email subscriptions*. Subscriptions created through this API are the equivalent of users clicking the **Subscribe** button on the asset detail page. Once a user subscribes to an asset, they will be notified via email of events occurring on the asset. For a list of events for which subscribed users are notified, see the Email Templates section in [Oracle Enterprise Repository Administration Guide](#).

Using the subscriptions API of REX, developers can create, delete, and inspect subscriptions to lists of assets. The operations are always performed for the user identified in the authentication token passed as an argument to the various subscription methods.

Use Cases

Use Case: Create Subscription to Assets

Description

- Authenticate with REX.
- Read a list of asset summaries via a query.
- Subscribe to the matched assets.

Sample Code:

```
package com.flashline.sample.subscriptionapi;  
  
import java.net.MalformedURLException;
```

```

import java.net.URL;
import java.rmi.RemoteException;

import javax.xml.rpc.ServiceException;

import com.flashline.registry.openapi.base.OpenAPIException;
import com.flashline.registry.openapi.entity.AssetSummary;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.query.AssetCriteria;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class CreateSubscription {
    public static void main(String pArgs[]) throws OpenAPIException,
        RemoteException, ServiceException {

        try {

            // //////////////////////////////////////
            // Connect to Oracle Enterprise Repository
            // //////////////////////////////////////
            URL IURL = null;
            IURL = new URL(pArgs[0]);
            FlashlineRegistry repository = new FlashlineRegistryServiceLocator()
                .getFlashlineRegistry(IURL);

            // //////////////////////////////////////
            // Login to OER
            // //////////////////////////////////////
            AuthToken authToken = repository.authTokenCreate(pArgs[1], pArgs[2]);

            // //////////////////////////////////////
            // find the assets to which to subscribe
            // //////////////////////////////////////
            AssetCriteria criteria = new AssetCriteria();
            criteria.setNameCriteria("%");
            AssetSummary[] lAssetSummaries = repository.assetQuerySummary(authToken,
                criteria);

            // //////////////////////////////////////
            // Iterate through assets, pulling out the ids and adding
            // to the array of longs
            // //////////////////////////////////////
            long[] lAssetIDs = new long[lAssetSummaries.length];
            for (int i = 0; i < lAssetSummaries.length; i++) {
                lAssetIDs[i] = lAssetSummaries[i].getID();
            }

            // //////////////////////////////////////
            // Create the subscriptions. The value of "false" for the

```

```

// parameter pFailOnAnyError, will cause the operation to NOT
// fail for any asset to which the user does not have VIEW
// privileges, or for which the asset is not subscribable.
//
// If this value is not "false", the operation will throw
// an exception if any asset in the array of asset IDs is
// not subscribable or viewable by the user, and NONE of the
// subscriptions will be recorded in the repository.
// //////////////////////////////////////
repository.subscriptionCreate(authToken, lAssetIDs, false);

} catch (OpenAPIException lEx) {
    System.out.println("ServerCode = " + lEx.getServerErrorCode());
    System.out.println("Message   = " + lEx.getMessage());
    System.out.println("StackTrace:");
    lEx.printStackTrace();
} catch (RemoteException lEx) {
    lEx.printStackTrace();
} catch (ServiceException lEx) {
    lEx.printStackTrace();
} catch (MalformedURLException lEx) {
    lEx.printStackTrace();
}
}
}
}

```

Use Case: Delete Subscription to Assets

Description

- Authenticate with REX.
- Read a list of asset summaries via a query.
- Delete any subscriptions that may exist for the matched assets.

Sample Code:

```

package com.flashline.sample.subscriptionapi;

import java.net.MalformedURLException;
import java.net.URL;
import java.rmi.RemoteException;

import javax.xml.rpc.ServiceException;

import com.flashline.registry.openapi.base.OpenAPIException;

```

```

import com.flashline.registry.openapi.entity.AssetSummary;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.query.AssetCriteria;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class DeleteSubscription {
    public static void main(String pArgs[]) throws OpenAPIException, RemoteException,
        ServiceException {
        try {

            //////////////////////////////////////
            // Connect to Oracle Enterprise Repository
            //////////////////////////////////////
            URL IURL = null;
            IURL = new URL(pArgs[0]);
            FlashlineRegistry repository = new FlashlineRegistryServiceLocator()
                .getFlashlineRegistry(IURL);

            //////////////////////////////////////
            // Login to OER
            //////////////////////////////////////
            AuthToken authToken = repository.authTokenCreate(
                pArgs[1],pArgs[2]);

            //////////////////////////////////////
            // find the assets for which to delete subscriptions
            //////////////////////////////////////
            AssetCriteria criteria = new AssetCriteria();
            criteria.setNameCriteria("");
            AssetSummary[] lAssetSummaries = repository.assetQuerySummary(authToken, criteria);

            //////////////////////////////////////
            // Iterate through assets, pulling out the ids and adding
            // to the array of longs
            //////////////////////////////////////
            long[] lAssetIDs = new long[lAssetSummaries.length];
            for (int i = 0; i < lAssetSummaries.length; i++) {
                lAssetIDs[i] = lAssetSummaries[i].getID();
            }

            //////////////////////////////////////
            // Delete the subscriptions on the list of assets.
            //////////////////////////////////////
            repository.subscriptionDelete(authToken, lAssetIDs);

        } catch (OpenAPIException lEx) {
            System.out.println("ServerCode = " + lEx.getServerErrorCode());
            System.out.println("Message = " + lEx.getMessage());
            System.out.println("StackTrace:");
        }
    }
}

```

```

    lEx.printStackTrace();
} catch (RemoteException lEx) {
    lEx.printStackTrace();
} catch (ServiceException lEx) {
    lEx.printStackTrace();
} catch (MalformedURLException lEx) {
    lEx.printStackTrace();
}
}
}
}

```

Use Case: Read Subscriptions for Assets

Description

- Authenticate with REX.
- Read the list of subscribed assets for the authenticated user.

Sample Code:

```

package com.flashline.sample.subscriptionapi;

import java.net.MalformedURLException;
import java.net.URL;
import java.rmi.RemoteException;

import javax.xml.rpc.ServiceException;

import com.flashline.registry.openapi.base.OpenAPIException;
import com.flashline.registry.openapi.entity.Asset;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class ReadSubscriptions {
    public static void main(String pArgs[]) throws OpenAPIException, RemoteException,
        ServiceException {
        try {

            //////////////////////////////////////
            // Connect to Oracle Enterprise Repository
            //////////////////////////////////////
            URL lURL = null;
            lURL = new URL(pArgs[0]);
            FlashlineRegistry repository = new FlashlineRegistryServiceLocator()

```

```

.getFlashlineRegistry(IURL);

////////////////////////////////////////
// Login to OER
////////////////////////////////////////
AuthToken authToken = repository.authTokenCreate(
    pArgs[1],pArgs[2]);

////////////////////////////////////////
// Read all of the assets to which the user is subscribed.
////////////////////////////////////////
long[] lSubscribedAssets = repository.subscriptionReadSubscribedAssets(authToken);

////////////////////////////////////////
// Print out the assets to which the user is subscribed
////////////////////////////////////////
Asset[] lAssets = repository.assetReadArray(authToken, lSubscribedAssets);
System.out.println("Subscribed Assets for user "+pArgs[1]);
for(int i=0; i<lAssets.length; i++){
    System.out.println(" -> "+lAssets[i].getLongName());
}

} catch (OpenAPIException lEx) {
    System.out.println("ServerCode = " + lEx.getServerErrorCode());
    System.out.println("Message   = " + lEx.getMessage());
    System.out.println("StackTrace:");
    lEx.printStackTrace();
} catch (RemoteException lEx) {
    lEx.printStackTrace();
} catch (ServiceException lEx) {
    lEx.printStackTrace();
} catch (MalformedURLException lEx) {
    lEx.printStackTrace();
}
}
}

```

Use Case: Read Users Subscribed to an Asset

Description

- Authenticate with REX.
- Read the list of users subscribed to a particular asset.

Sample Code:

```

package com.flashline.sample.subscriptionapi;

import java.net.MalformedURLException;
import java.net.URL;
import java.rmi.RemoteException;

import javax.xml.rpc.ServiceException;

import com.flashline.registry.openapi.base.OpenAPIException;
import com.flashline.registry.openapi.entity.AssetSummary;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.entity.RegistryUser;
import com.flashline.registry.openapi.query.AssetCriteria;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class ReadSubscribersToAsset {
    public static void main(String pArgs[]) throws OpenAPIException, RemoteException,
        ServiceException {
        try {

            ////////////////////////////////////////////////////
            // Connect to Oracle Enterprise Repository
            ////////////////////////////////////////////////////
            URL IURL = null;
            IURL = new URL(pArgs[0]);
            FlashlineRegistry repository = new FlashlineRegistryServiceLocator()
                .getFlashlineRegistry(IURL);

            ////////////////////////////////////////////////////
            // Login to OER
            ////////////////////////////////////////////////////
            AuthToken authToken = repository.authTokenCreate(
                pArgs[1],pArgs[2]);

            ////////////////////////////////////////////////////
            // Assume that this query will return some number of assets...
            ////////////////////////////////////////////////////
            AssetCriteria ICriteria = new AssetCriteria();
            ICriteria.setNameCriteria("%");
            AssetSummary[] IAssetSummaries = repository.assetQuerySummary(authToken, ICriteria);

            ////////////////////////////////////////////////////
            // Read the users that are subscribed to the first asset
            ////////////////////////////////////////////////////
            RegistryUser[] ISubscribedUsers =
                repository.subscriptionReadUsersSubscribedToAsset(authToken, IAssetSummaries[0].getID());

            for (int i=0; i<ISubscribedUsers.length; i++){
                System.out.println("Subscribed Users: "+ISubscribedUsers[i].getUserName());
            }
        }
    }
}

```

```
    }  
  } catch (OpenAPIException lEx) {  
    System.out.println("ServerCode = " + lEx.getServerErrorCode());  
    System.out.println("Message  = " + lEx.getMessage());  
    System.out.println("StackTrace:");  
    lEx.printStackTrace();  
  } catch (RemoteException lEx) {  
    lEx.printStackTrace();  
  } catch (ServiceException lEx) {  
    lEx.printStackTrace();  
  } catch (MalformedURLException lEx) {  
    lEx.printStackTrace();  
  }  
}  
}  
}
```