Oracle Enterprise Repository

Asset Type API

Overview

Description

Types (Asset Types and Compliance Templates) define the structure of assets. Types consist of two main parts:

- **Editor**
  - Defines the metadata that is stored for the assets and determines how metadata elements are organized in the **Asset Editor**.
- **Viewer**
  - Defines how the metadata elements will be displayed in the asset detail in Oracle Enterprise Repository.

Definitions

When creating or editing a Type, acceptable value lists and categorization types are used as metadata elements. These metadata elements are referenced by ID in the Editor and Viewer XML for the Type.

When creating or editing assets, Types define the metadata elements that are used in the custom data for the asset (`Asset.GetCustomData()`).

- **Notes**
  - While the code examples in this section refer to **Asset Types**, please be aware that the processes described in the use cases can be used to create both **Asset Types** and **Compliance Templates**. For more information on **Compliance Templates** see the Oracle Enterprise Repository Compliance Templates Guide.

- Editor and viewer metadata is represented as CDATA escaped XML. Consequently, if a large number of AssetTypes are returned via a call to assetTypeQuery, it is possible that some XML parsers may exceed their *entity expansion limit*. On some popular parsers, the default entity expansion limit is set to 64,000. If this limit is exceeded, it can be increased on JAXP compliant processors by passing a command-line parameter called **entityExpanionLimit**. For example, passing the following parameter to the JVM will increase the entity expansion limit to 5 MB:

```
java -DentityExpansionLimit=5242880 com.example.MyApp
```

Use Cases

Use Case: Create and edit a new Type

Description

Adding a new Type to the repository.

Sample code:

```
package com.flashline.sample.assettypeapi;

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

import javax.xml.rpc.ServiceException;

import com.flashline.registry.openapi.base.OpenAPIException;
import com.flashline.registry.openapi.entity.AssetType;
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 CreateNewAssetType {
  public static void main(String pArgs[]) throws RemoteException, OpenAPIException,
      ServiceException {
    try {

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

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

      // //////////////////////////////
      // Select an existing Type on which to base the new one
      // //////////////////////////////
      String newAssetTypeName = "My AssetType";
      long baseAssetTypeID = 151;
      AssetType newAssetType = repository.assetTypeCreateClone(
          authToken, baseAssetTypeID, newAssetTypeName);
      System.out.println("The new Asset Type id =\"" + newAssetType.getID()
          + "\"");

      // //////////////////////////////
      // Manipulate xml strings
      // //////////////////////////////
      String lEditorXML = newAssetType.getEditorXML();
      String lViewerXML = newAssetType.getViewerXML();

      // Perform XML manipulation on the editor and viewer definitions...

      // //////////////////////////////
      // Set the new editor/viewer definitions on the asset type, and save the
      // type back to OER
      // //////////////////////////////
      newAssetType.setEditorXML(lEditorXML);
      newAssetType.setViewerXML(lViewerXML);

      repository.assetTypeUpdate(authToken, newAssetType);

      // ----------------------------------------
      // clean up sample
      repository.assetTypeDelete(authToken, newAssetType.getID());
    } 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: Create a Compliance Template Type

Description

Adding a new Compliance Template Type to the repository.

Sample code:

```
package com.flashline.sample.assettypeapi;

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

import javax.xml.rpc.ServiceException;

import com.flashline.registry.openapi.base.OpenAPIException;
import com.flashline.registry.openapi.entity.AssetType;
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 CreateNewComplianceTemplateType {

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

      ///////////////////////////////////////////////////////////
      // Connect to Oracle Enterprise Repository
      ///////////////////////////////////////////////////////////
      URL lURL = null;
      lURL = new URL(
          pArgs[0]);
      FlashlineRegistry repository = new FlashlineRegistryServiceLocator()
          .getFlashlineRegistry(lURL);
      // /////////////////////////////
      // Authenticate with OER
      // /////////////////////////////
      AuthToken authToken = repository.authTokenCreate(pArgs[1],
          pArgs[2]);

      // /////////////////////////////
      // Create a new compliance template.
      // /////////////////////////////
      String newAssetTypeName = "My Compliance Template"+Calendar.getInstance().getTimeInMillis();
      AssetType newAssetType = repository
          .assetTypeCreateComplianceTemplate(authToken, newAssetTypeName);

    } 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: Find Types

Description

Locating a Type in the repository.

Sample code:

```
package com.flashline.sample.assettypeapi;

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.AssetType;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.query.AssetTypeCriteria;
import com.flashline.registry.openapi.query.SearchTerm;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

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

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

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

      // //////////////////////////////
      // Create SearchTerms and set them on the AssetSearchCriteria
      // //////////////////////////////
      AssetTypeCriteria assetTypeCriteria = new AssetTypeCriteria();
      SearchTerm[] searchTerms = new SearchTerm[1];
      searchTerms[0] = new SearchTerm();
      searchTerms[0].setKey("name");
      searchTerms[0].setValue("Component");
      assetTypeCriteria.setSearchTerms(searchTerms);

      // //////////////////////////////
      // Perform the search using the specified criteria
      // //////////////////////////////
      AssetType[] assetTypes = repository.assetTypeQuery(authToken,
          assetTypeCriteria);

    } 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();
    }
  }
}
```

Pitfalls

Methods to avoid:

- `setIcon`
- `setID`

Use Case: Read tab types

Description

Retrieve the list of tabs available for an asset type.

Sample code:

```
package com.flashline.sample.assettypeapi;

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.AuthToken;
import com.flashline.registry.openapi.entity.TabTypeBean;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class ReadTabTypes {
  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(lURL);
      TabTypeBean[] lTabTypeBeans = null;

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


      ///////////////////////////////////
      // read the tab types of an assettype
      ///////////////////////////////////
      lTabTypeBeans = repository.assetTypeTabsRead(authToken, 100);


    } 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: Retrieve all Asset Type tabs

Description

Retrieves all asset type tabs within the Oracle Enterprise Repository.

Sample code:

```
package com.flashline.sample.assettypeapi;

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.AuthToken;
import com.flashline.registry.openapi.entity.TabTypeBean;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class ReadTabTypes {
  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(lURL);
      TabTypeBean[] lTabTypeBeans = null;

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


      ////////////////////////////////////
      // read the tab types of an assettype
      ////////////////////////////////////
      lTabTypeBeans = repository.assetTypeTabsRead(authToken, 100);


    } 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();
   }
  }
}
```

Example of the RelationshipTypeQuery

```
try
{
  RelationshipTypeCriteria rCriteria = new RelationshipTypeCriteria();
  RelationshipType[] allRelationshipTypes = FlashlineRegistry.relationshipQuery(lAuthToken, rCriteria);
}
catch (OpenAPIException e)
{
    e.printStackTrace();
}
catch (RemoteException re)
{
   re.printStackTrace();
}
```