

AcceptableValueLists API

Overview

Description / Definition

Acceptable Value Lists are used in single- and multiple-selection drop-down box metadata elements.

When creating or editing an asset type, Acceptable Value Lists are used as metadata elements. These metadata elements are referenced by ID in the editor and viewer XML for the asset type/compliance template.

When creating or editing assets, values contained in Acceptable Value Lists are used as options for the metadata elements defined for the particular asset type/compliance template. To use the acceptable values for an Acceptable Value List, the custom data for the asset (`Asset.GetCustomData()`) is modified to reference the ID of the acceptable value.

Use Cases

Use Case: Create and Edit an Acceptable Value List

Description

Create a new acceptable value list and enter it into Oracle Enterprise Repository.

Sample code is as follows:

```
package com.flashline.sample.acceptablevaluelists;  
  
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.AcceptableValue;
import com.flashline.registry.openapi.entity.AcceptableValueList;
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 CreateAndEditAcceptableValueList {
    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);

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

            // //////////////////////////////////////
            // Build an array of acceptable values for the list.
            // //////////////////////////////////////
            String newAcceptableValueListName = "My AcceptableValueList "+Calendar.getInstance().getTimeInMillis();
            AcceptableValue[] acceptableValues = new AcceptableValue[3];
            acceptableValues[0] = new AcceptableValue();
            acceptableValues[0].setValue("My Value");

            acceptableValues[1] = new AcceptableValue();
            acceptableValues[1].setValue("My Next Value");

            acceptableValues[2] = new AcceptableValue();
            acceptableValues[2].setValue("My Last Value");

            // //////////////////////////////////////
            // Create the AcceptableValueList in Repository
            // //////////////////////////////////////
            AcceptableValueList newAcceptableValueList = repository

```

```

        .acceptableValueListCreate(authToken, newAcceptableValueListName,
            acceptableValues);

    System.out.println("The new acceptableValueList id =\" +
        + newAcceptableValueList.getID() + "\"");

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

```

Use Case: Find an Acceptable Value List and use it in an asset

Description

Populate an asset's single or multiple selection lists with acceptable values.

Sample code is as follows:

```

package com.flashline.sample.acceptablevaluelists;

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.AcceptableValue;
import com.flashline.registry.openapi.entity.AcceptableValueList;
import com.flashline.registry.openapi.entity.Asset;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.query.AcceptableValueListCriteria;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

```

```

public class FindAcceptableValueListAndUseInAsset {
    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);

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

            //////////////////////////////////////
            // Find the AcceptableValueList
            //////////////////////////////////////
            AcceptableValueListCriteria criteria = new AcceptableValueListCriteria();
            criteria.setNameCriteria("My AcceptableValueList");
            AcceptableValueList[] acceptableValueLists = repository
                .acceptableValueListQuery(authToken, criteria);

            AcceptableValueList myAcceptableValueList = acceptableValueLists[0];
            AcceptableValue[] acceptableValues = myAcceptableValueList
                .getAcceptableValues();

            //////////////////////////////////////
            // Find one value within the AcceptableValueList
            //////////////////////////////////////
            AcceptableValue myAcceptableValue = null;
            for (int i = 0; i < acceptableValues.length; i++) {
                if (acceptableValues[i].getValue().equals("My Value")) {
                    myAcceptableValue = acceptableValues[i];
                    break;
                }
            }
            long myAcceptableValueID = myAcceptableValue.getID();
            Asset myAsset = repository.assetRead(authToken, 561);
            String customData = myAsset.getCustomData();

            //////////////////////////////////////
            // Modify customData to use myAcceptableValueID.

```

```

// ///////////////////////////////////
String modifiedCustomData = customData;

// ...

// ///////////////////////////////////
// save modified custom data
// ///////////////////////////////////
myAsset.setCustomData(modifiedCustomData);

repository.assetUpdate(authToken, myAsset);
} catch (OpenAPIException IEx) {
    System.out.println("ServerCode = " + IEx.getServerErrorCode());
    System.out.println("Message   = " + IEx.getMessage());
    System.out.println("StackTrace:");
    IEx.printStackTrace();
} catch (RemoteException IEx) {
    IEx.printStackTrace();
} catch (ServiceException IEx) {
    IEx.printStackTrace();
} catch (MalformedURLException IEx) {
    IEx.printStackTrace();
}
}
}
}

```