Oracle Enterprise Repository

Vendor API

Overview

Description

Vendors are the original source of assets, and are responsible for their support. Vendors are identified by a single name string.

Definitions

Validation - When saving a Vendor, Oracle Enterprise Repository currently validates that:

- The vendor name has to be less than 250 characters
- The Vendor name is unique

Related subsystems

There is a one to many relationship between assets and vendors (i.e. multiple assets can be linked to the same vendor, but an asset can only have one vendor). When creating or editing assets the Vendor ID metadata element linking the Vendor to the asset can also be modified.

Additional Import(s) Required

```
import com.flashline.registry.openapi.entity.Vendor;
import com.flashline.registry.openapi.query.VendorCriteria;
```

Use Cases

Use Case: Manipulating Vendors

Description

- Adding a new Vendor to Oracle Enterprise Repository.
- Assigning an existing Vendor to an asset.

Sample Code:

```
package com.flashline.sample.vendorapi;

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.entity.Vendor;
import com.flashline.registry.openapi.query.VendorCriteria;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class Vendors {
  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);

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

      // ----------------------------------------
      // Create a new vendor
      String newVendorName = "My Vendor";
      Vendor newVendor = repository.vendorCreate(authToken, newVendorName);
      System.out.println("The new vendor id =\"" + newVendor.getID() + "\"");

      // -----------------------------------------
      // Find a vendor and update an asset to use it
      VendorCriteria criteria = new VendorCriteria();
      criteria.setNameCriteria(newVendorName);
      Vendor[] vendors = repository.vendorQuery(authToken, criteria);
      long myVendorID = vendors[0].getID();

      long MY_ASSET_ID = 569;
      Asset myAsset = repository.assetRead(authToken, MY_ASSET_ID);
      // MY_ASSET_ID must be the asset id of an asset in the repository
      myAsset.setVendorID(myVendorID);
      repository.assetUpdate(authToken, myAsset);

      // ----------------------------------------
      // clean up
      myAsset.setVendorID(0);
      repository.vendorDelete(authToken, newVendor.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();
    }
  }

}
```