Oracle Enterprise Repository

Policy API

REX now supports the following functions against Policies

- Query Policy:
  - Status of the Policy (pass/fail) on an Asset
  - Status of the collection of Policies on an Asset
  - Obtain XML from the Policy Assertion Technical Description Field
  - Assets that the Policy is applied too
- Update Policy
  - Maintain list of individual Policy Assertions on a Policy
  - Set status of individual Policy Assertions for an Asset.
  - Apply and remove Policy from assets

Additional Import(s) Required (Some may not be used in all examples.)

```
import com.flashline.registry.openapi.entity.Asset;
import com.flashline.registry.openapi.entity.PolicyAssertion;
import com.flashline.registry.openapi.entity.PolicyAssertionResult;
```

- **Note on Policies**
  - Policies in Oracle Enterprise Repository are a specific type of asset, based on the **Policy Type**. Refer to the **Asset API** use cases for information related to the creation, modification and removal of a Policy.

Definitions

- **Assertion**
  - An assertion is a policy statement added to a policy asset.

- **AssertionResult**
  - When a Policy has been applied to an asset, each assertion within the policy can be evaluated for the asset. The Assertion Result is pass, fail or unknown for any asset and assertion pair.

Methods

- There are four new methods available with the **`FlashlineRegistry`** service
  - `assetReadAppliedPolicies()`
  - `assetUpdateAppliedPolicies()`
  - `assetEvaluateAgainstPolicy()`
  - `assetEvaluateAgainstAllPolicies()`

Use Cases

Use Case: Create a Policy

Description

To create a new policy, create a new asset based on the Policy Type (102).

Sample Code:

```
package com.flashline.sample.policies;

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

import com.flashline.registry.openapi.entity.Asset;
import com.flashline.registry.openapi.entity.AssetType;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.entity.PolicyAssertion;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class CreatePolicySample {

  private static final String POLICY_TYPE_NAME_PREFIX = "Policies-Test Policy Type";
  private static final long ASSET_POLICY_ARCHETYPE = 102;
  private static final String POLICY_NAME_PREFIX = "Policies-Test Policy";
  private static final String POLICY_VERSION = "1.0";

  private static FlashlineRegistry mRepository = null;
  private static AssetType mPolicyAssetType = null;
  private AuthToken mAuthToken = null;

  public CreatePolicySample(String[] pArgs) {

    try {

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

      // //////////////////////////////
      // Authenticate with OER
      // //////////////////////////////
      mAuthToken  = mRepository.authTokenCreate(pArgs[1], pArgs[2]);

      mPolicyAssetType = createPolicyAssetType();
    } catch(Exception e) {

    }
  }


  public static void main(String[] pArgs) {
    try {

      CreatePolicySample lCreatePolicySample = new CreatePolicySample(pArgs);

      // ---------------------------
      // create a new policy object
      Asset lPolicy = lCreatePolicySample.createPolicy();

    } catch(Exception e) {
      e.printStackTrace();
    }

  }


  /**
   * Creates an asset policy with a unique name
   */
  private Asset createPolicy() throws RemoteException {
    String lPolicyName = POLICY_NAME_PREFIX + " " + System.currentTimeMillis();

    // ---------------------------
    // create a policy (an asset whose assettype's archtype is "102" (policy)
    Asset lPolicy = mRepository.assetCreate(mAuthToken, lPolicyName, POLICY_VERSION, mPolicyAssetType.getID());
    lPolicy.setCustomData("<custom-data></custom-data>");

    // ---------------------------
    // set some polcy assertions
    lPolicy.setPolicyAssertions(generateSampleAssertions());

    return mRepository.assetUpdate(mAuthToken, lPolicy);
  }
```

```
  /**
   * Returns several sample policy assertions for use in testing.
   * Located in a function to be shared between test calls.
   *
   * @return Array of policy assertions
   */
  private PolicyAssertion[] generateSampleAssertions() {
    PolicyAssertion[] lPolicyAssertions = new PolicyAssertion[3];
    String[] lPolicyAssertionNames = {"First", "Second", "Third"};
    for (int i=0; i<lPolicyAssertionNames.length; i++) {
      String lPolicyAssertionName = "My " + lPolicyAssertionNames[i] + " Assertion";

      lPolicyAssertions[i] = new PolicyAssertion();
      lPolicyAssertions[i].setName(lPolicyAssertionName);
      lPolicyAssertions[i].setDescription(lPolicyAssertionName + " Description");
      lPolicyAssertions[i].setTechnicalDefinition(lPolicyAssertionName + " Technical Definition");
    }

    return lPolicyAssertions;
  }

  /**
   * Creates an asset policy asset type with a unique name
   */
  private AssetType createPolicyAssetType() throws RemoteException {
    String lPolicyTypeName = POLICY_TYPE_NAME_PREFIX + " " + System.currentTimeMillis();

    // ---------------------------
    // create a new asset type
    AssetType lPolicyType = mRepository.assetTypeCreate(mAuthToken, lPolicyTypeName);

    // ---------------------------
    // update the asset type to be a policy asset type by settings the archtype = 102
    lPolicyType.setArcheTypeIDs(new long[] {ASSET_POLICY_ARCHETYPE});
    return mRepository.assetTypeUpdate(mAuthToken, lPolicyType);
  }

}
```

Use Case: Get All Policies

Description

To get all policies, find all assets whose assetype's archetype is a policy archetype (102).

Sample Code:

```
package com.flashline.sample.policies;
```

```
import java.net.URL;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;

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

public class FindPoliciesSample {

  private static FlashlineRegistry mRepository = null;
  private static AuthToken mAuthToken = null;

  public FindPoliciesSample(String[] pArgs) {

    try {

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

      // //////////////////////////////
      // Authenticate with OER
      // //////////////////////////////
      mAuthToken  = mRepository.authTokenCreate(pArgs[1], pArgs[2]);

    } catch(Exception e) {

    }
  }


  public static void main(String[] pArgs) {
    try {

      FindPoliciesSample lFindPoliciesSample = new FindPoliciesSample(pArgs);

      AssetType[] lPolicyAssetTypes = null;
      Asset[] lPolicies = null;

      AssetTypeCriteria lAssetTypeCriteria = null;
      AssetCriteria lAssetCritera = null;

      List lListPolicies = new LinkedList();
      // -----------------------
      // search for all asset types that have the policy (102) archetype
      lAssetTypeCriteria = new AssetTypeCriteria();
      lAssetTypeCriteria.setArcheTypeCriteria("102");

      lPolicyAssetTypes = mRepository.assetTypeQuery(mAuthToken, lAssetTypeCriteria);

      for(int i=0; i<lPolicyAssetTypes.length; i++) {

        // -----------------------
        // for each policy assettype, search for all assets that are of policy assettype

        lAssetCritera = new AssetCriteria();
        lAssetCritera.setAssetTypeCriteria(lPolicyAssetTypes[i].getID());

        lPolicies = mRepository.assetQuery(mAuthToken, lAssetCritera);

        // -----------------------
        // add polices to list
        lListPolicies.addAll(Arrays.asList(lPolicies));

      }



    } catch(Exception e) {
      e.printStackTrace();
    }

  }

}
```

Use Case: Get/Set Policy Assertions

Description

To get policy assertions, call `getPolicyAssertions`. To set policy assertions, call `setPolicyAssertions`, then update the policy.

Sample Code:

```
package com.flashline.sample.policies;

import java.net.URL;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;

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

public class GetSetPolicyAssertionsSample {

  private static FlashlineRegistry mRepository = null;
  private static AuthToken mAuthToken = null;

  public GetSetPolicyAssertionsSample(String[] pArgs) {

    try {

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

      // //////////////////////////////
      // Authenticate with OER
      // //////////////////////////////
      mAuthToken  = mRepository.authTokenCreate(pArgs[1], pArgs[2]);

    } catch(Exception e) {

    }
  }


  public static void main(String[] pArgs) {
    try {

      GetSetPolicyAssertionsSample lGetSetPolicyAssertionsSample = new GetSetPolicyAssertionsSample(pArgs);

      AssetType[] lPolicyAssetTypes = null;
      Asset[] lPolicies = null;

      AssetTypeCriteria lAssetTypeCriteria = null;
      AssetCriteria lAssetCritera = null;

      List lListPolicies = new LinkedList();


      // -----------------------
      // search for all asset types that have the policy (102) archetype
      lAssetTypeCriteria = new AssetTypeCriteria();
      lAssetTypeCriteria.setArcheTypeCriteria("102");
  lPolicyAssetTypes = mRepository.assetTypeQuery(mAuthToken, lAssetTypeCriteria);

  for(int i=0; i<lPolicyAssetTypes.length; i++) {

    // -----------------------
    // for each policy assettype, search for all assets that are of policy assettype

    lAssetCritera = new AssetCriteria();
    lAssetCritera.setAssetTypeCriteria(lPolicyAssetTypes[i].getID());

    lPolicies = mRepository.assetQuery(mAuthToken, lAssetCritera);

    // -----------------------
    // add polices to list
    lListPolicies.addAll(Arrays.asList(lPolicies));

  }

  if(lListPolicies.size() > 0) {

    // -----------------------
    // get the first policy
    Asset lPolicy = (Asset)lListPolicies.get(0);


    // -----------------------
    // get the policy assertions
    PolicyAssertion[] lPolicyAssetions = lPolicy.getPolicyAssertions();


    // -----------------------
    // print out the policy assertions
    for(int i=0; i<lPolicyAssetions.length; i++) {
      lPolicyAssetions[i].toString();
    }

    // -----------------------
    // set different policy assertions
    lPolicy.setPolicyAssertions(generateNewAssertions());


    // -----------------------
    // update the asset with new assertions
    mRepository.assetUpdate(mAuthToken, lPolicy);


  } else {
    System.out.println("No policies were found in OER.");
  }



} catch(Exception e) {
      e.printStackTrace();
    }

  }


  /**
   * Returns several sample policy assertions for use in testing.
   * Located in a function to be shared between test calls.
   *
   * @return Array of policy assertions
   */
  private static PolicyAssertion[] generateNewAssertions() {
    PolicyAssertion[] lPolicyAssertions = new PolicyAssertion[3];
    String[] lPolicyAssertionNames = {"NEW-First", "NEW-Second", "NEW-Third"};
    for (int i=0; i<lPolicyAssertionNames.length; i++) {
      String lPolicyAssertionName = "My " + lPolicyAssertionNames[i] + " Assertion";

      lPolicyAssertions[i] = new PolicyAssertion();
      lPolicyAssertions[i].setName(lPolicyAssertionName);
      lPolicyAssertions[i].setDescription(lPolicyAssertionName + " Description");
      lPolicyAssertions[i].setTechnicalDefinition(lPolicyAssertionName + " Technical Definition");
    }

    return lPolicyAssertions;
  }

}
```

Use Case: Get Policies That Have Been Applied To An Asset

Description

Call `assetReadAppliedPolicies` to obtain policies applied to an asset.

Sample Code:

```
package com.flashline.sample.policies;

import java.net.URL;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;

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

public class GetAppliedPoliciesSample {

  private static FlashlineRegistry mRepository = null;
  private static AuthToken mAuthToken = null;

  public GetAppliedPoliciesSample(String[] pArgs) {

    try {

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

      // //////////////////////////////
      // Authenticate with OER
      // //////////////////////////////
      mAuthToken  = mRepository.authTokenCreate(pArgs[1], pArgs[2]);

    } catch(Exception e) {

    }
  }


  public static void main(String[] pArgs) {
    try {

      GetAppliedPoliciesSample lGetAppliedPoliciesSample = new GetAppliedPoliciesSample(pArgs);

      long lAssetId = 50000;

      // ---------------
      // read the policed appled to asset 50000
      Asset[] lAppliedPolicies = mRepository.assetReadAppliedPolicies(mAuthToken, lAssetId);


    } catch(Exception e) {
      e.printStackTrace();
    }

  }


}
```

Use Case: Set Which Policies Are Applied To An Asset

Description

Call `assetUpdateAppliedPolicies` to update policies that have been applied to an asset.

Sample Code:

```
package com.flashline.sample.policies;

import java.net.URL;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;

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

public class ApplyPoliciesSample {

  private static FlashlineRegistry mRepository = null;
  private static AuthToken mAuthToken = null;

  public ApplyPoliciesSample(String pArgs[]) {

    try {

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

      // //////////////////////////////
      // Authenticate with OER
      // //////////////////////////////
      mAuthToken  = mRepository.authTokenCreate(pArgs[1], pArgs[2]);

    } catch(Exception e) {

    }
  }


  public static void main(String[] pArgs) {
    try {
```

```
   ApplyPoliciesSample lApplyPoliciesSample = new ApplyPoliciesSample(pArgs);

   long lAssetId = 50000;
   long[] lPolicyIds = {50000, 50001, 50002};

   mRepository.assetUpdateAppliedPolicies(mAuthToken, lAssetId, lPolicyIds);


  } catch(Exception e) {
   e.printStackTrace();
  }

 }


}
```

Use Case: Evaluate Asset Compliance

Description

Use `assetEvaluateAgainstPolicy` to determine an asset's compliance with a specified policy. Use `assetEvalauteAgainstAllPolicies` to determine an asset's compliance against all applied policies.

Sample Code:

```
package com.flashline.sample.policies;

import java.net.URL;

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 PolicyEvaluationSample {

 private static FlashlineRegistry mRepository = null;
 private static AuthToken mAuthToken = null;

 public PolicyEvaluationSample(String[] pArgs) {

  try {

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

```
      mRepository = new FlashlineRegistryServiceLocator().getFlashlineRegistry(lURL);

      // //////////////////////////////
      // Authenticate with OER
      // //////////////////////////////
      mAuthToken  = mRepository.authTokenCreate(pArgs[1], pArgs[2]);

    } catch(Exception e) {

    }
  }


  public static void main(String[] pArgs) {
    try {

      PolicyEvaluationSample lPolicyEvalSamp = new PolicyEvaluationSample(pArgs);
      long lAssetId = 50000;
      long lPolicyId = 50001;
      String lEvaluationResult = null;

      // --------------------
      // evaluate asset id 50000 against policy id 50001
      // the return will be one of the following values "pass", "fail", "unknown"
      lEvaluationResult = mRepository.assetEvaluateAgainstPolicy(mAuthToken, lAssetId, lPolicyId);


      // --------------------
      // evaluate asset id 50000 against all polices applied to the asset
      // the return will be one of the following values "pass", "fail", "unknown"
      lEvaluationResult = mRepository.assetEvaluateAgainstAllPolicies(mAuthToken, lAssetId);

    } catch(Exception e) {
      e.printStackTrace();
    }

  }





}
```