

# Oracle Enterprise Repository

## Notification API

### Overview

The Notification Subsystem provides a Web Services-based mechanism that can be used to create Oracle Enterprise Repository notifications.

### Use Cases

#### Use Case: Read Notification Substitution List

##### *Description*

To create a new read notification substitution list.

#### Use Case: Create A Notification

##### *Description*

To create a Oracle Enterprise Repository notifications.

Sample Code:

```
package com.flashline.sample.artifactstoreapi;  
  
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.ArtifactStoreBean;
import com.flashline.registry.openapi.entity.AuthToken;
import com.flashline.registry.openapi.query.ArtifactStoreCriteria;
import com.flashline.registry.openapi.service.v300.FlashlineRegistry;
import com.flashline.registry.openapi.service.v300.FlashlineRegistryServiceLocator;

public class ArtifactStores {
    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]);

            // -----
            // query for an artifact store
            ArtifactStoreCriteria lArtifactStoreCriteria = null;
            ArtifactStoreBean[] lArtifactStoreBeans = null;
            ArtifactStoreBean lArtifactStoreBean = null;

            lArtifactStoreCriteria = new ArtifactStoreCriteria();
            lArtifactStoreCriteria.setHostCriteria("existing-artifact-store.com");
            lArtifactStoreCriteria.setBasepathCriteria("/");

            lArtifactStoreBeans = repository.artifactStoreQuery(authToken, lArtifactStoreCriteria, false);

            // create a missing artifact store if missing and based on the criteria

            lArtifactStoreCriteria = new ArtifactStoreCriteria();

            lArtifactStoreCriteria.setHostCriteria("missing-artifact-store.com");
            lArtifactStoreCriteria.setBasepathCriteria("/");

            // a new artifact store will be created
            lArtifactStoreBeans = repository.artifactStoreQuery(authToken, lArtifactStoreCriteria, true);

```

```
lArtifactStoreBean = lArtifactStoreBeans[0];

} catch(Exception e) {
    throw new RuntimeException(e.getMessage());
}

}
}
```