QAT Global
Code Sample

SysMgmt

Contents

1 Introduction

As part of QAT training, we are developing several code samples. These code samples illustrate various technical as well as business aspects. These samples will help the developer get a quick understanding of the technology and framework. This will considerably reduce the learning time. The samples can be used as a reference to develop the application.

1.1 Pre-requisites

In order to extract the maximum benefit out the samples, its expected that the developers have

- Completed Basic Java training
- Completed Basic Spring training
- Completed Development environment setup – JDK , Eclipse, Oracle/postgreSQL, Tomcat/JBoss SOAPUI etc.
- Acquire the sysmgmt-* projects from SVN or local file server or tech-lead/instructor. If downloading from SVN these projects are normally located under the trunk/Samples folder: sysmgmt-*
- **Ask your instructor whether you will be using postgreSQL or Oracle as your DB**

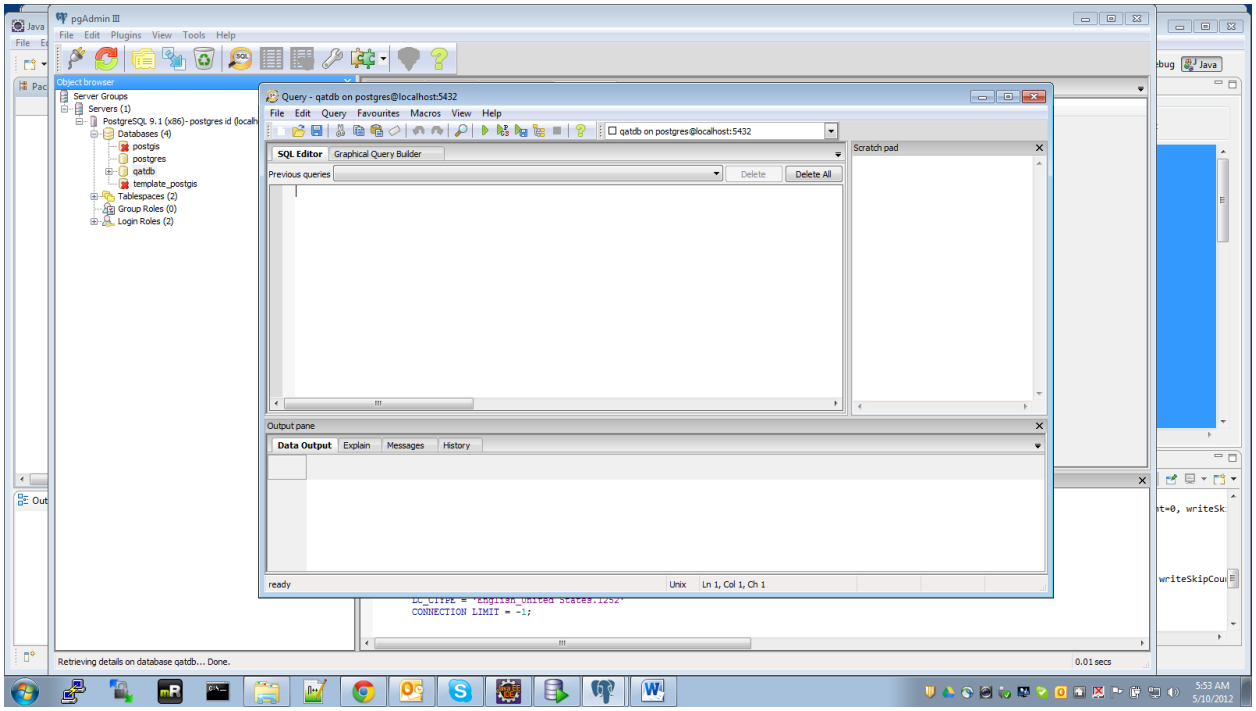
2 Setting up postgreSQL DB

2.1.1 Locate and open the file in Notepad++ sysmgmtsampledb-postgres.sql file in the following directory.

C:\QATEclipseWorkSpace \sysmgmt-test\setup

Do a select all and copy……

2.1.2 Open pgAdmin III and connect to localhost and select the “qatdb” under databases and click the SQL button from the menu



3. Copy the contents of the **sysmgmtsampledb-postgres.sql** file and into the window and run the statements. Verify the tables are there and now the database setup is complete.

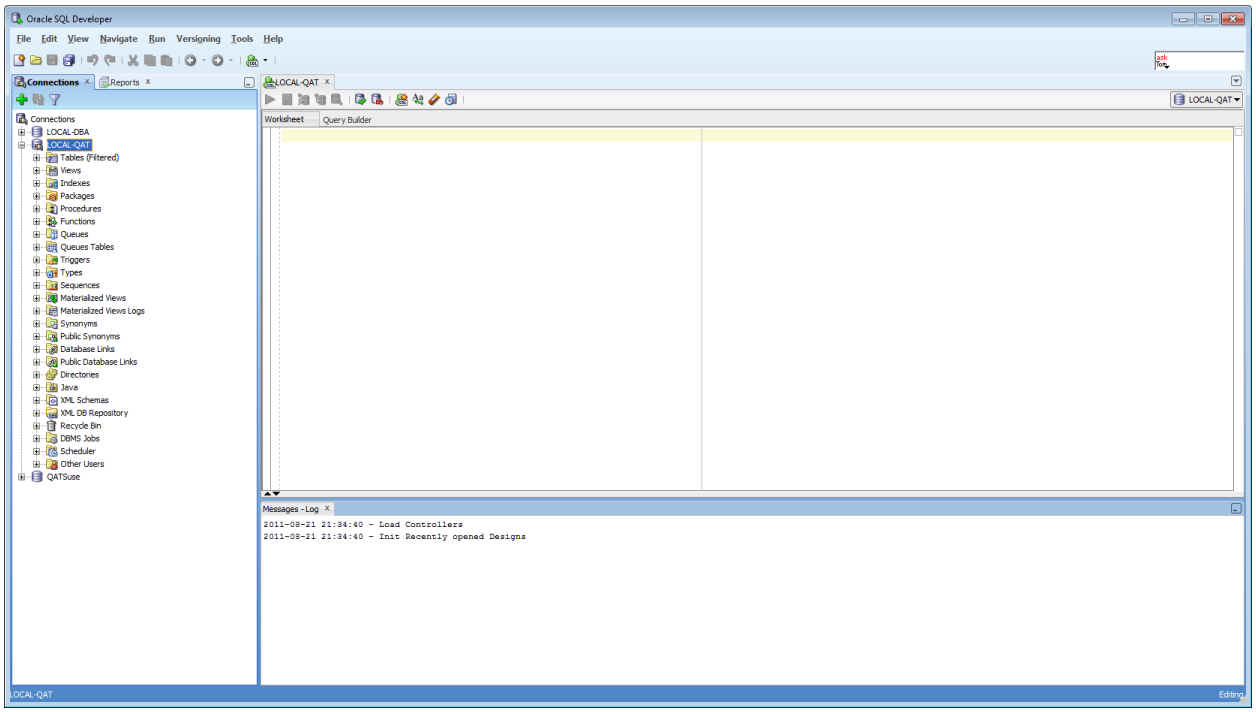
2.2 Setting up ORACLE DB

2.2.1 Locate and open the file in Notepad++ sysmgmtsampledb-oracle.sql file in the following directory.

C:\QATEclipseWorkSpace \sysmgmt-test

Do a select all and copy……

2.2.2 Open SQL Developer and connect to localhost



2.2.3 Copy the contents of the sysmgmtsampledb-oracle.sql file and into the window and run the statements. Verify the tables are there and now the database setup is complete.

3 Building the project

3.1 Building the project

Note when you first open the projects and if Eclipse tries to build them the build will fail since all the dependencies cannot be resolved at this time until they are first built and published.

In addition you will see IVY errors related to not being able to resolve certain dependencies defined in the project classpath.

This is okay since all the various projects and dependencies have not yet been built so the first thing you will need to do is build the projects.

3.1.1 Building and publishing for the first time

In order for IVY dependency management to work we need to build the projects and then publish them to a local IVY repository which will be located on your development machine. This in turn enables the dependency management capabilities of IVY to be leveraged in order to build the rest of our projects.

Start with the sysmgmt-interface project.

1. Open the build.ant.xml file located in the *build* folder.
2. Run this ANT script which builds the project source and publishes a JAR file to your local IVY repository which in turn will allow subsequent projects to resolve and build correctly.
3. Review the output and notice how the various dependencies are resolved.

Next open up the sysmgmt-implementation project

1. Open the build.ant.xml file located in the *build* folder.
2. Run this ANT script which builds the project source and publishes a JAR file to your local IVY repository which in turn will allow subsequent projects to resolve and build correctly.
3. Review the output and notice how the various dependencies are resolved.

Next open up the sysmgmt-batch project (separate class for later)

1. Open the build.ant.xml file located in the *build* folder.
2. Run this ANT script which builds the project source and publishes a JAR file to your local IVY repository which in turn will allow subsequent projects to resolve and build correctly.
3. Review the output and notice how the various dependencies are resolved.

If after doing this any of the projects are still failing to build under Eclipse then right-click on the project and select IVY➔Resolve
This will tell Eclipse to re-resolve all of the IVY dependencies and should clear-up any build issues.

You will notice the test project does not have an ANT build script. With this project all you need to do is perform a build under Eclipse for everything to work prior to executing any tests. You may have to perform an IVY Resolve as mentioned above.

3.1.2 Subsequent changes and building

When you make a source change make sure you **re-run the build scripts** so the most recent version of your project JAR is published to your local repository and the projects that depend on these changes point to the “latest and greatest”. For example if you update the interface project make sure you re-run the build script so the implementation projects has or uses the “latest and greatest” version.

3.2 Deploying the default sysmgmt-web war

3.2.1 Building the projects using Eclipse

Make sure you build the projects prior to executing the ANT script. If Eclipse auto-build is not turned on then manual build each of the projects in Eclipse.

3.2.2 Run the buildWar.ant.xml script

1. Locate the buildWar.ant.xml script located in the sysmgmt-web project under the build folder.
2. Right-click on this file and “Run As” followed by the first “Ant Build” menu item. This will execute the ANT build script producing a war file under the “dist” folder under the sysmgmt-web project folder.

3.2.3 Deploy to your web server

1. Copy the qat-sysmgmt-sample.war from the “dist” folder to either the tomcat or jboss hot deploy directories
2. Wait 30 seconds and then proceed to the testing section.

4 Testing

4.1 Testing via SOAP-UI

4.1.1 Double click on SOAP-UI icon. Incase, the console shows the message JAVA_HOME is not set, please set the JAVA_HOME.

4.1.2 To create a new project, press Ctrl+N or File->New Soap UI project. Or import “QAT-SysMgmt-CXF-soapui-project.xml” from the unittest\config directory in your source tree.

4.1.3 Enter the project specific information.

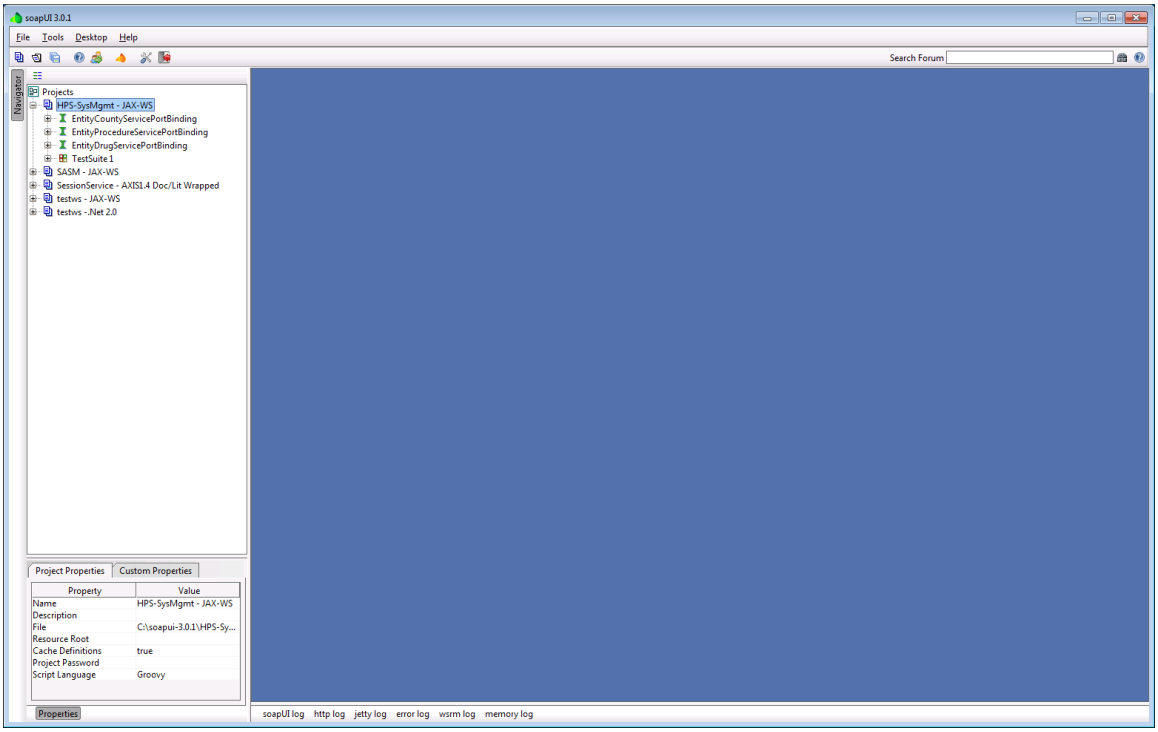
The WebServices WSDL URLs are the following::

http://localhost:8080/qat-sysmgmt-sample/services/ws/CountyService?WSDL

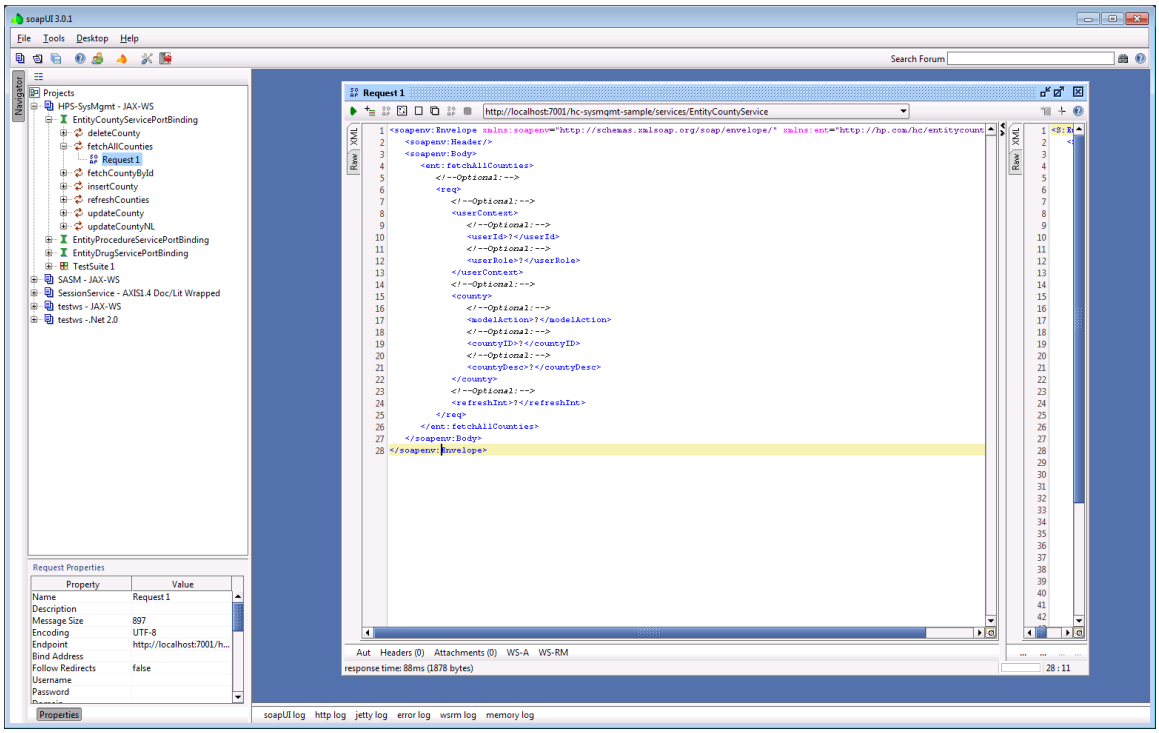
http://localhost:8080/qat-sysmgmt-sample/services/ws/ProcedureService?WSDL

http://localhost:8080/qat-sysmgmt-sample/services/ws/DrugService?WSDL

4.1.4 A new project is created and all the WS methods that are to be tested are displayed. Save the project. Make sure you add WS-Security Outgoing Settings otherwise not all your requests will work.

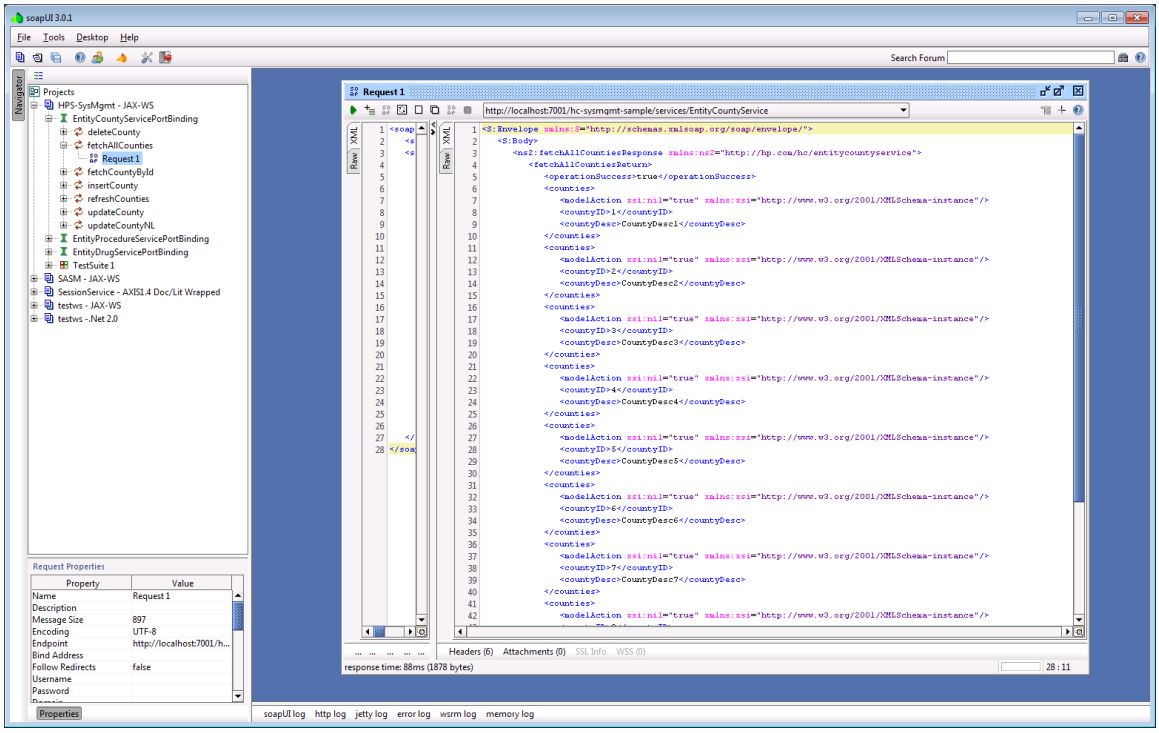


1. Click on the one of the requests.



2. You should see the request XML. Fill in appropriate parameters (where there are '?' marks) and click on the green button to execute the WS.

3. The XML response is shown in the next frame.

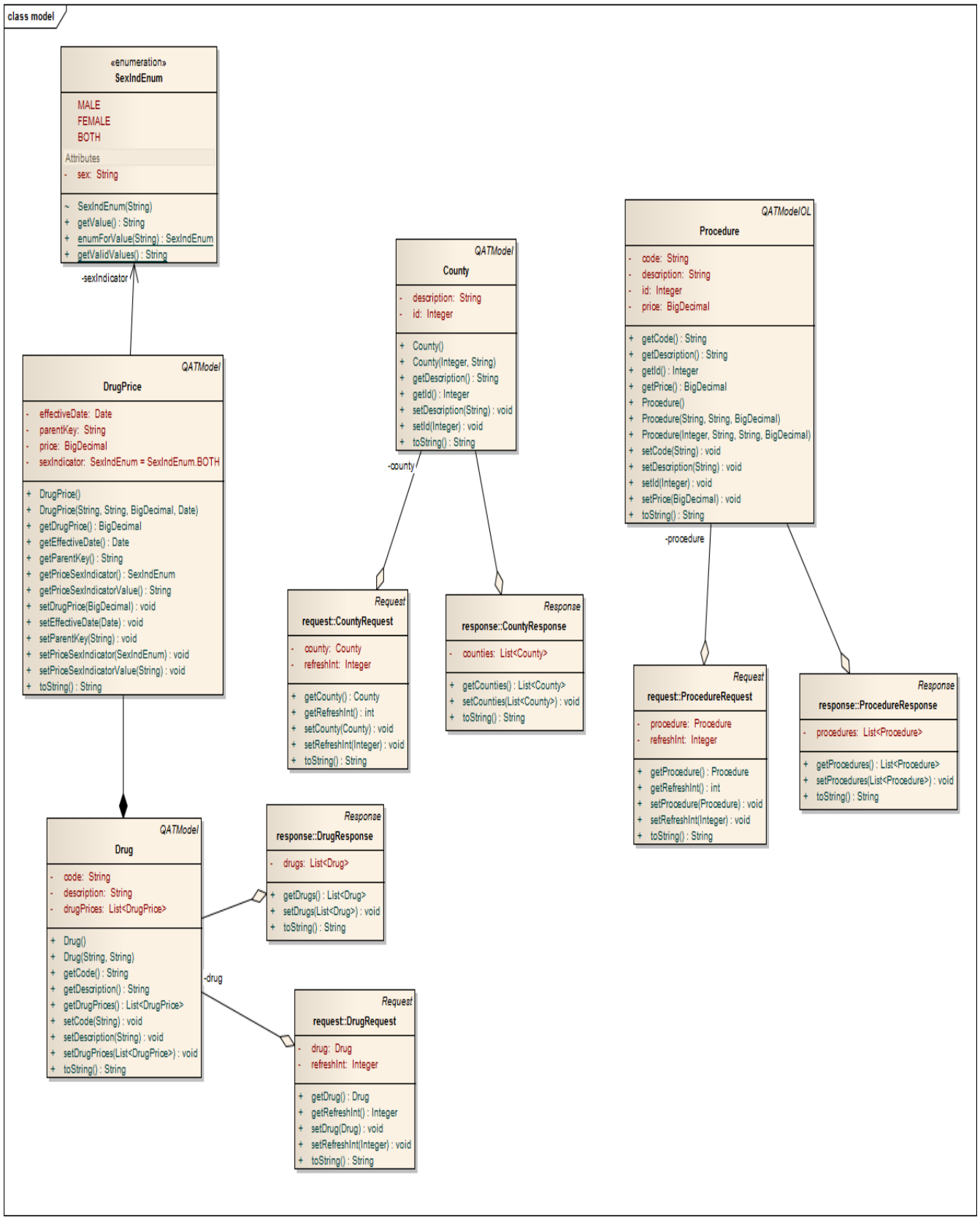


Modify the XML to feed in various parameters. You can export the request / results and make sure that the WS is functioning as required.

4. First test the following Requests,
    a. refreshCounties(XX) - this will load XX (it is an integer) rows of data
    b. refreshProcedures(XX) - this will load XX (it is an integer) rows of data
    c. refreshDrugs – No integer
    d. fetchAllCounties
    e. fetchAllProcedures
    f. fetchAllDrugs

    then use the ids returned in the above responses, to make the following requests

    g. fetchCountybyId
    h. fetchProcedurebyId
    i. fetchDrugByNDC
5. After run build.xml, do clean project and build inside of eclipse
    a. Then go to your bin directory inside of the eclipse project at a command prompt and run countyBASClient.bat and the cxf web service client will talk to cxf web service



5 Building and creating the WAR file using Ant/Ivy

This section describes how to build and generate the war file using Ant/Ivy. This should be done after you have completed all the previous sections including testing.

5.1 Building the projects using Ant/Ivy

Building the projects using Ant/Ivy requires the projects to be published to the repository. Since IVY uses a repository to resolve all it's dependencies the changes you make to the other projects must first be published to the repository so they will be included in the build.

Start with the sysmgmt-Interface project.

1. Open the publish.ant.xml file located in the build folder.
2. Run this ANT script publishing this module to the repository which will allow the other projects to resolve and build correctly.
3. Note the publish action copies the generated jar file to the repository to a unique location based on you user-id which you set earlier.

Next open up the sysmgmt-implementation project

1. Open the ivy.xml file and locate the dependency to the sysmgmt-interface project. Uncomment this dependency line for the sysmgmt-interface project.
2. Save the file. The Eclipse/IvyDE plug-in will start resolution and complete in the background. If there are any problems a dialog will pop-up in Eclipse.
3. Open the publish.ant.xml file located in the build folder.
4. Run this ANT script publishing this module to the repository which will allow the other projects to resolve and build correctly.
5. You can then run the build.ant.xml script. This script does not require any changes and will build the project using Ant and Ivy. Review the output and notice how the various dependencies are resolved.

With the help of your instructor open a browser and navigate to the Artifactory repository and browse the repository looking at all the artifacts including the ones you just published. You instructor will provide the URL for the repository.

3.2 Generating the war file

Once you have completed the building and publishing of the dependent projects using Ant/Ivy as described above you can generate the war file using Ant/Ivy.

Using the sysmgmt-web project:

1. Open the ivy.xml file located under the build folder.
2. Uncomment the dependencies for the sysmgmt-interface and implementation projects.
3. Make sure you also comment-out the dependencies with the hard-code org=sysmgmt. These where used when you first ran the buildWar.ant.xml script and used a previously published set of dependencies.

4. Save the file.
5. Run the buildWar.ant.xml script file.
6. This will generate the war file and place it into the dist folder under the sysmgmt-web project.
7. Use the qat-sysmgmt-sample.war file in your deployment as described previously.

Repeat the process of making changes to the source code base, building and publishing the dependent projects and then building and deploying the war file as needed until your changes are all complete.