OPTUM™

Optum Global Solutions

https://oneconnect.uhg.com/docs/DOC-53617

Open Shift Enterprise V3

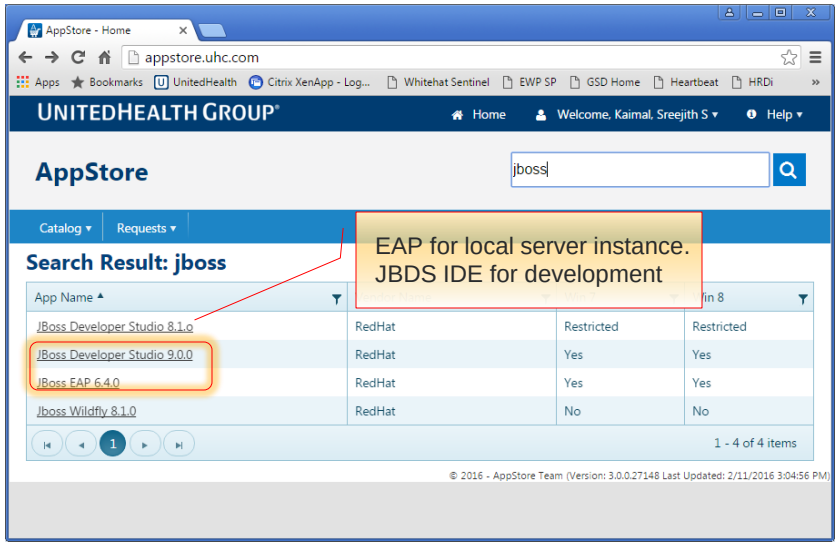
Day 03

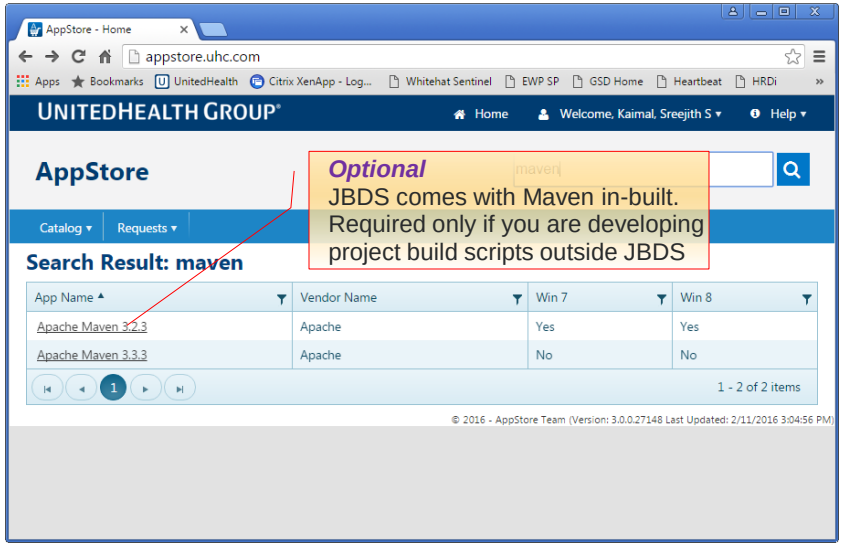
Training Presentation for OpenShift Enterprise

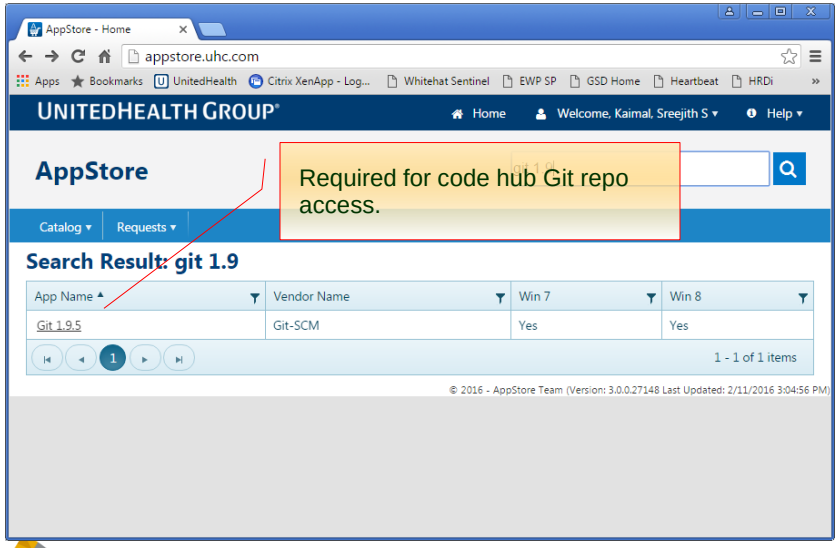
Sreejith Kaimal

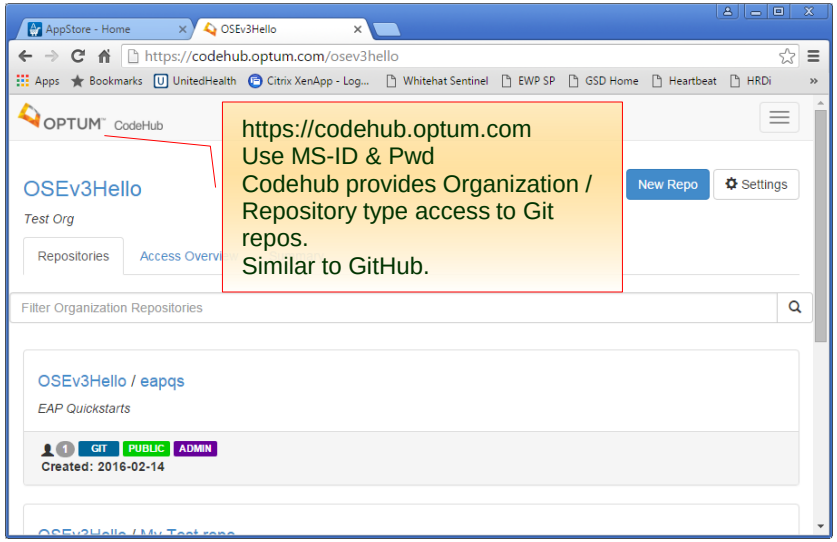
February 6, 2016

Required Apps

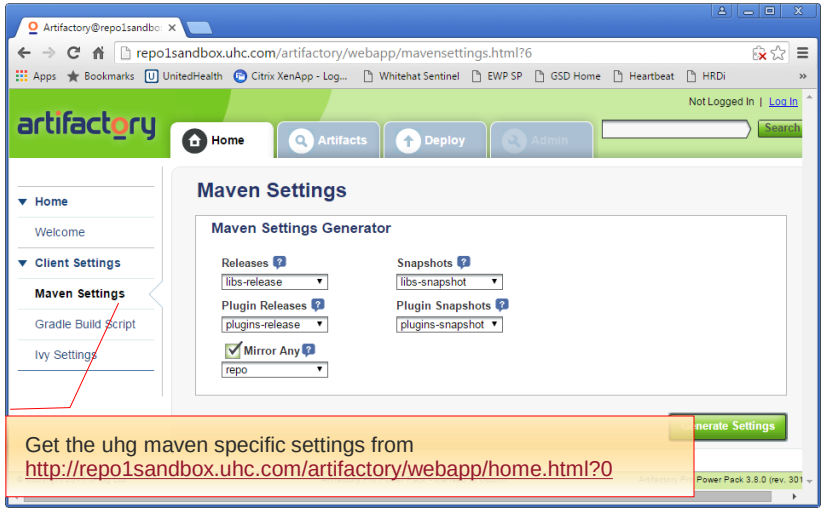






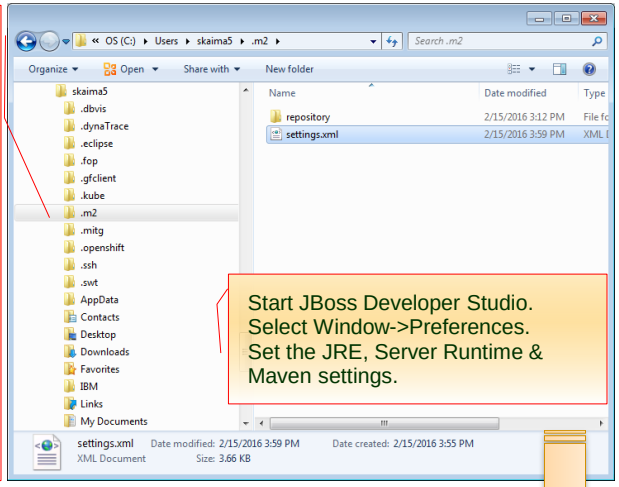


JBoss Developer Studio – Creating a Java EE Project for OSE v3

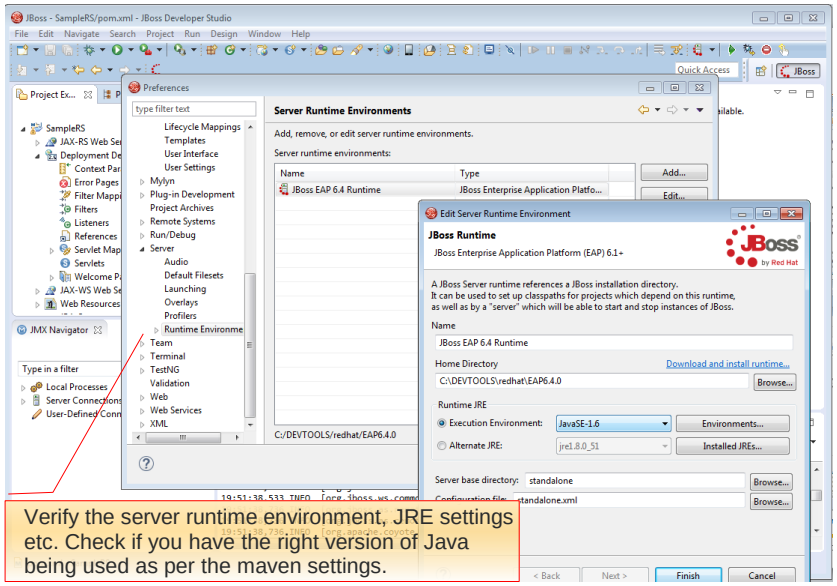


Get the uhg maven specific settings from
http://repo1sandbox.uhc.com/artifactory/webapp/home.html?0

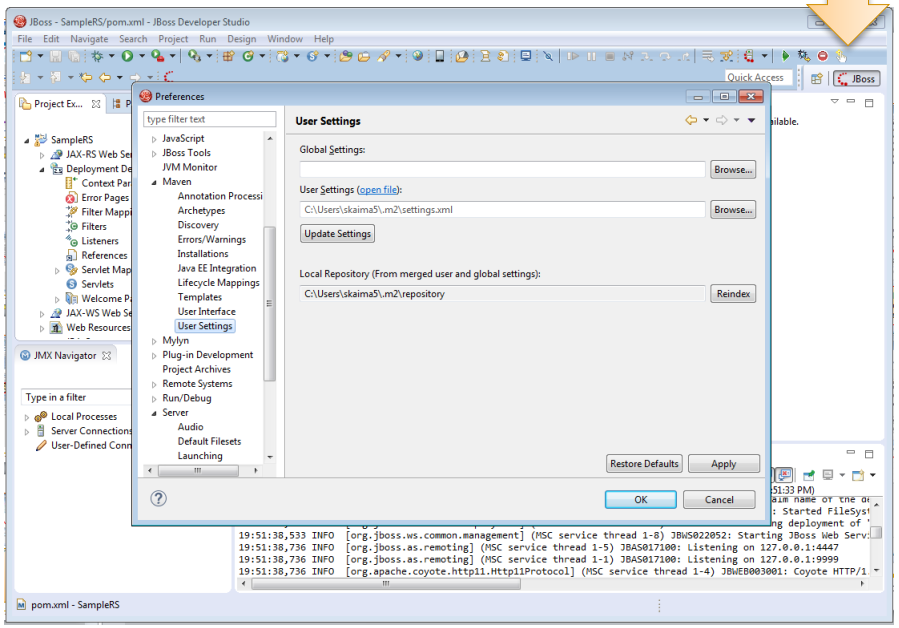
settings.xml would be defaulted to maven repos.
Some of it wont work within uhg network.
You need to replace this with the settings.xml generated from artifactory.
The settings.xml will be present in C:\users\<ntid>\.m2
Any new repos if added will be merged into this.
Verify this on the JBDS maven preferences.



Start JBoss Developer Studio.
Select Window->Preferences.
Set the JRE, Server Runtime & Maven settings.

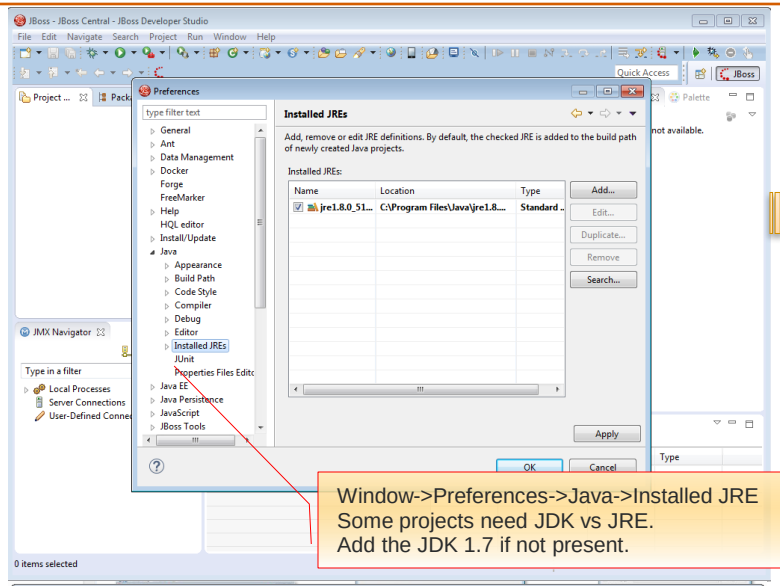


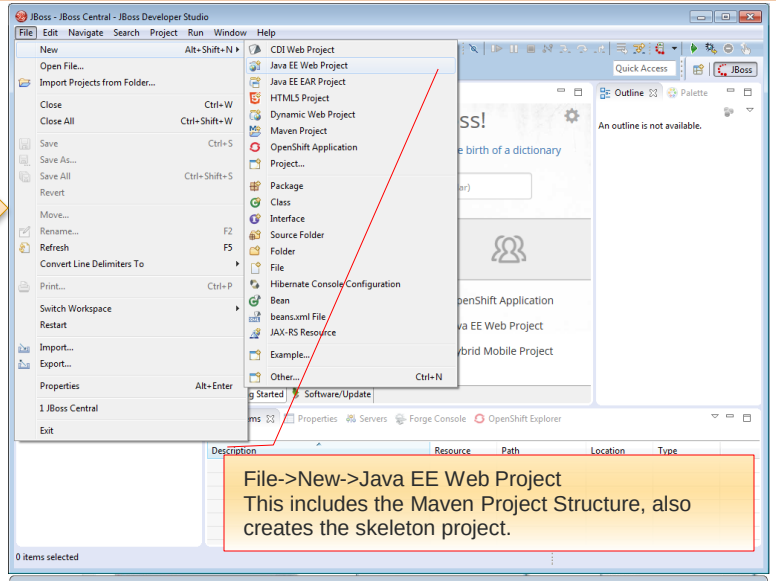
Verify the server runtime environment, JRE settings etc. Check if you have the right version of Java being used as per the maven settings.

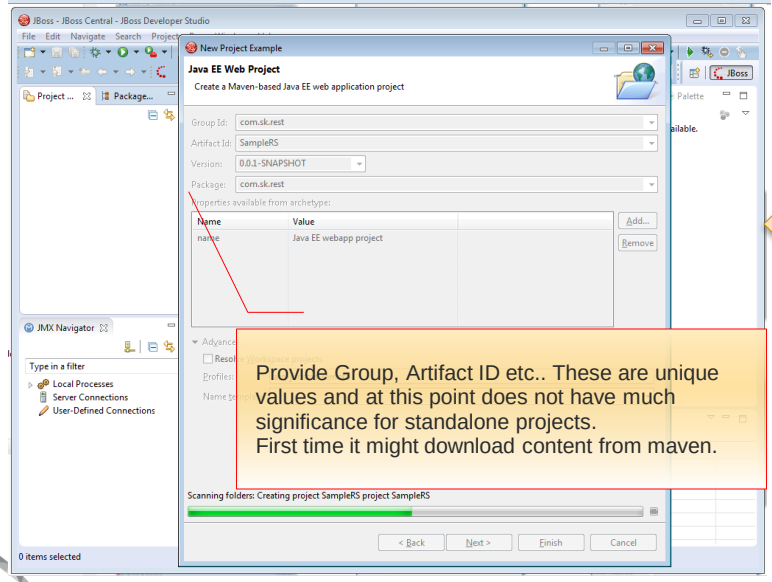


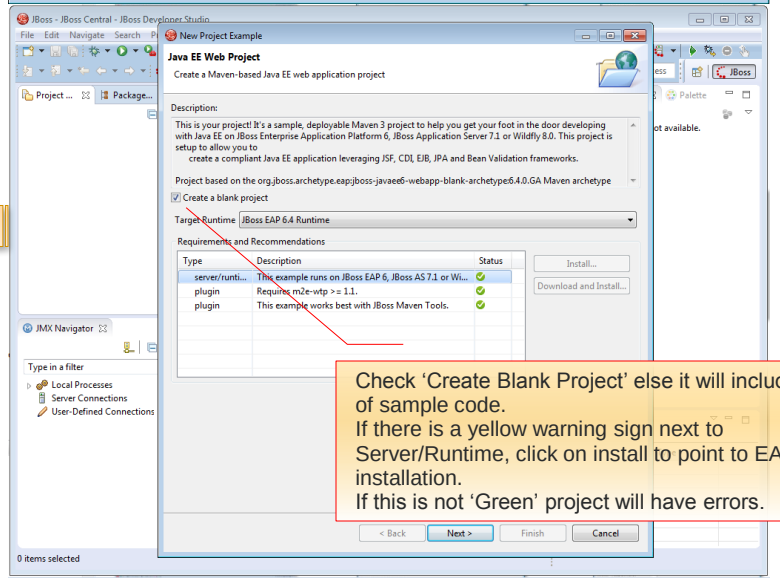
http://repo1sandbox.uhc.com/artifactory/webapp/mavensettings.html?1

JBDS – Creating a Java EE Project for OSE v3

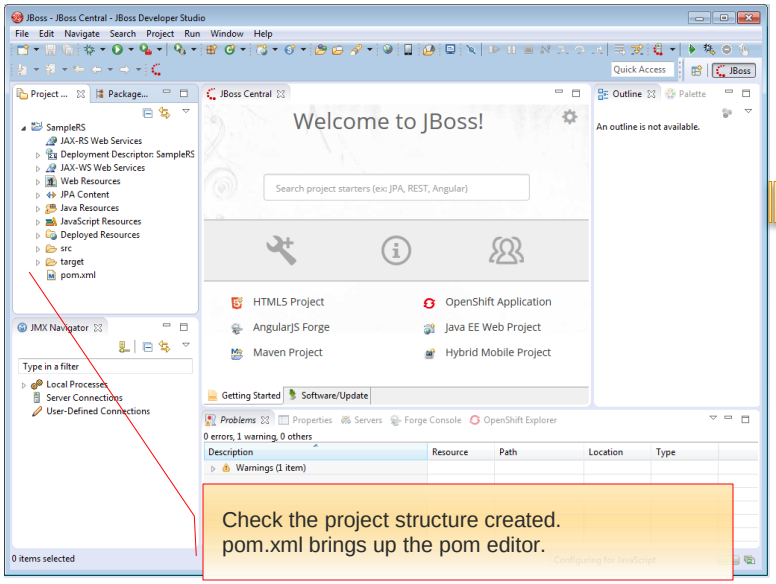




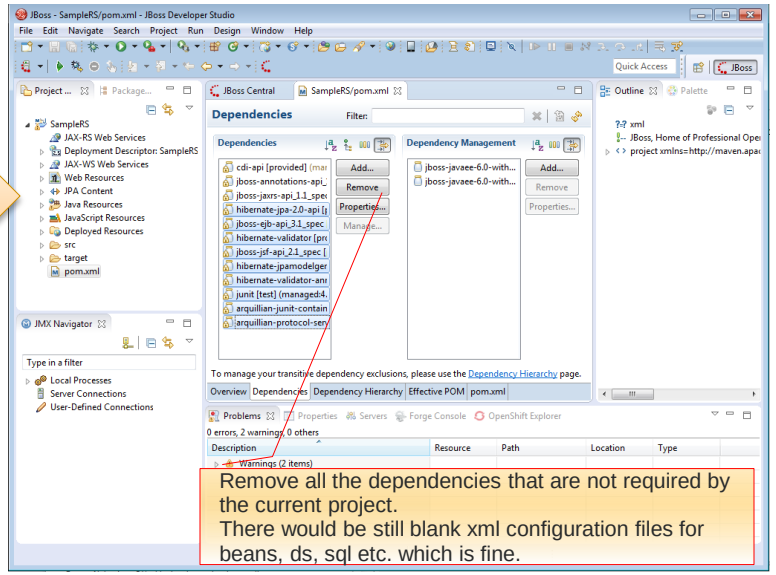




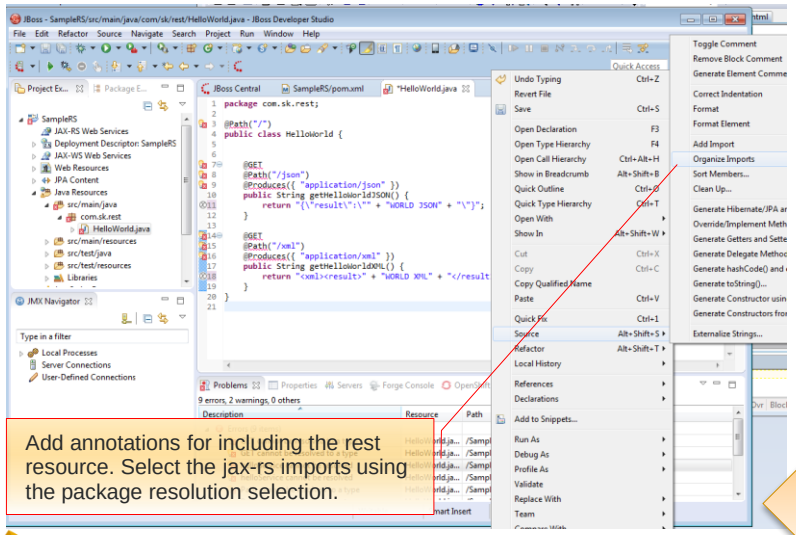
JBDS – Creating a Java EE Project for OSE v3



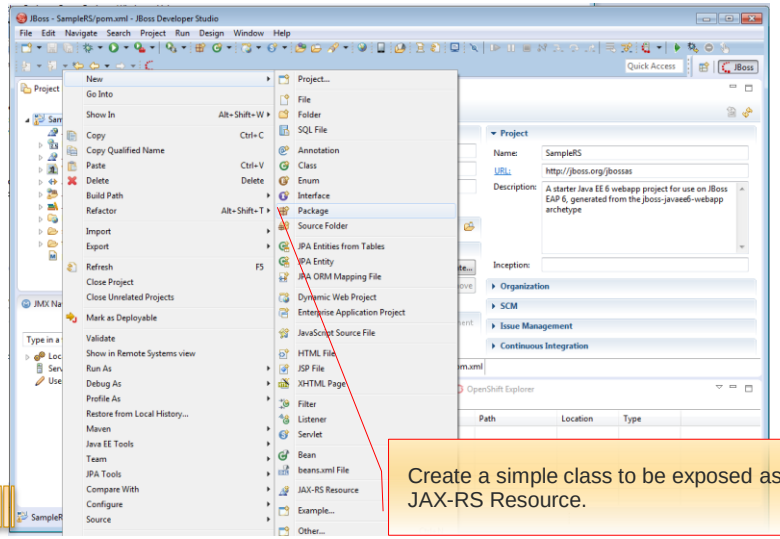
Check the project structure created.
pom.xml brings up the pom editor.



Remove all the dependencies that are not required by the current project.
There would be still blank xml configuration files for beans, ds, sql etc. which is fine.

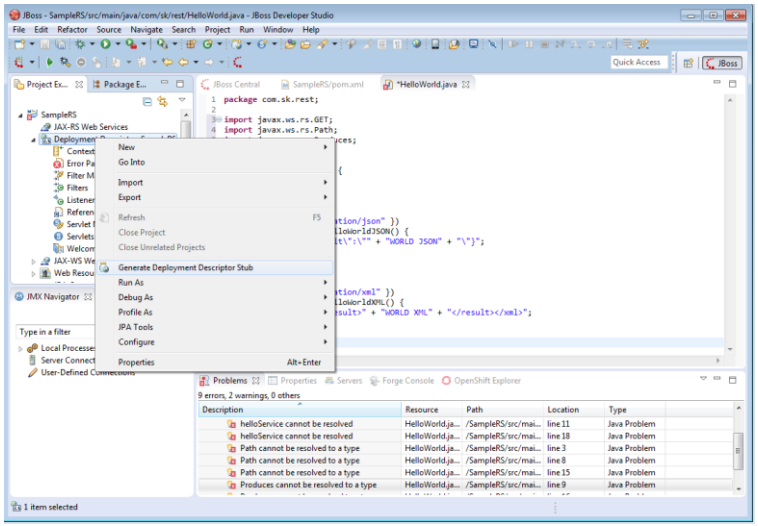


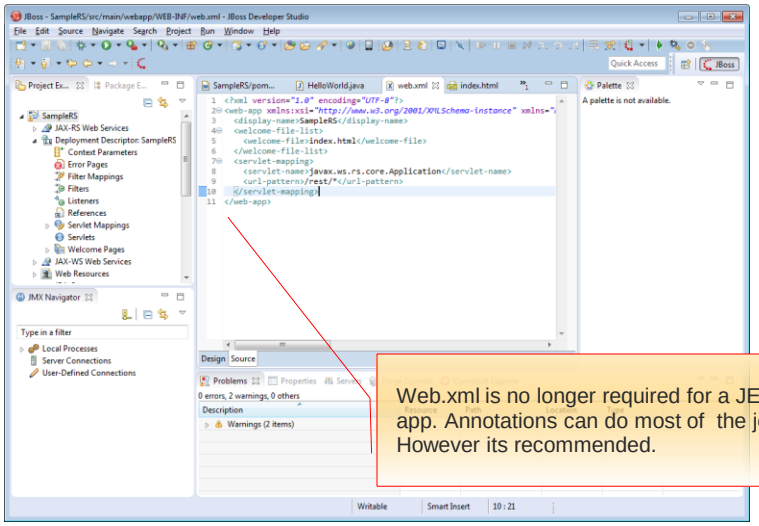
Add annotations for including the rest resource. Select the jax-rs imports using the package resolution selection.



Create a simple class to be exposed as a JAX-RS Resource.

JBDS – Creating a Java EE Project for OSE v3





These three methods are specified in the JAX-RS 1.1 specification to deploy JAX-RS applications

1. Subclassing javax.ws.rs.core.Application and using @ApplicationPath

```
@ApplicationPath("/mypath")
public class MyApplication extends Application {
}
```

2. Subclassing javax.ws.rs.core.Application and using web.xml

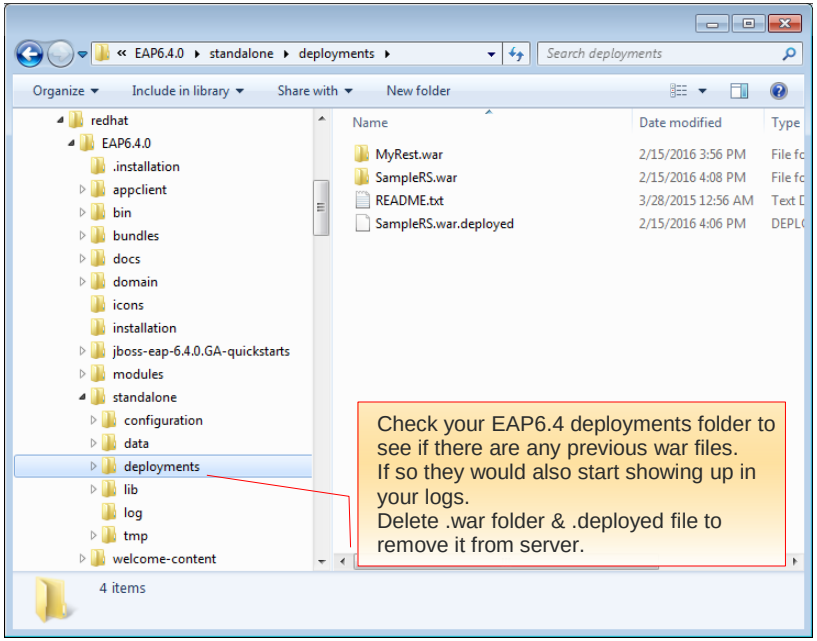
```
<servlet-mapping>
   <servlet-name>com.acme.MyApplication</servlet-name>
   <url-pattern>/hello/*</url-pattern>
</servlet-mapping>
```

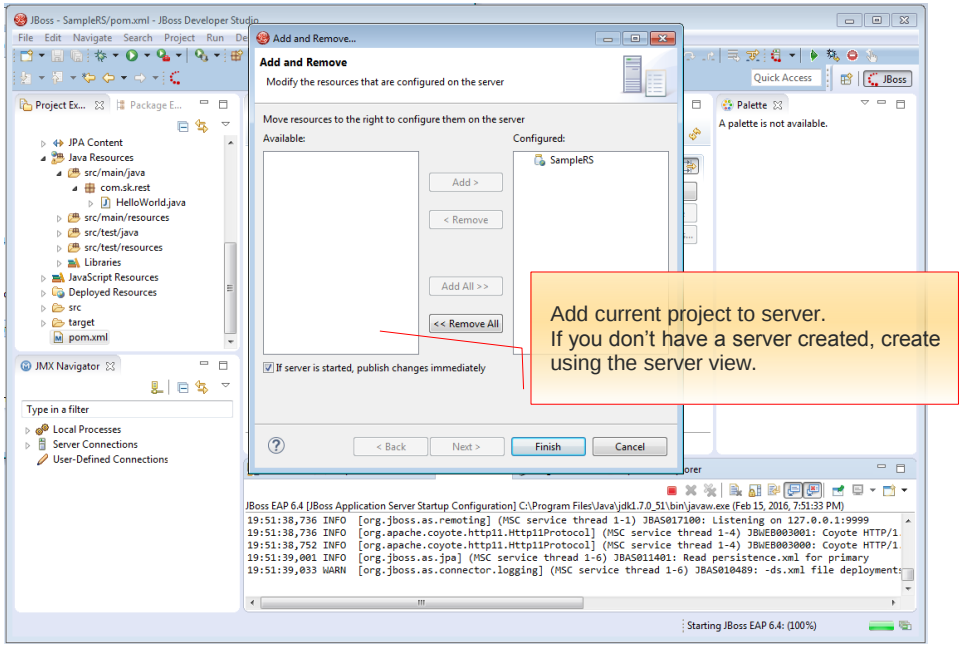
3. Using web.xml

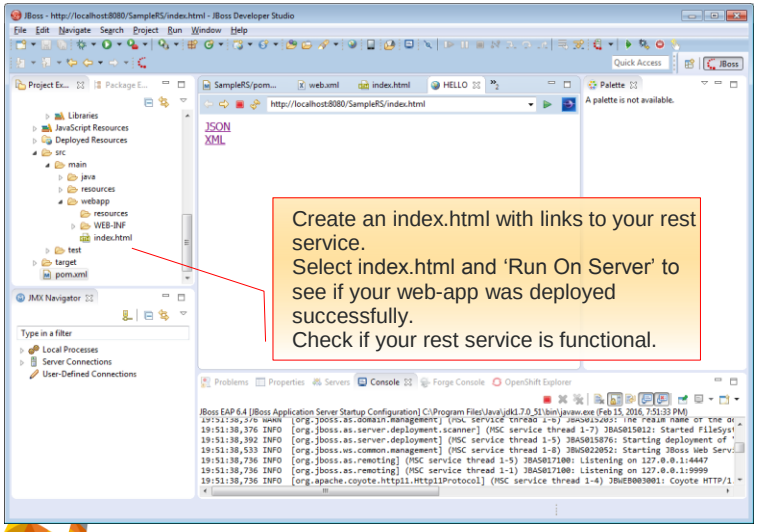
```
<servlet-mapping>
   <servlet-name>javax.ws.rs.core.Application</servlet-name>
   <url-pattern>/hello/*</url-pattern>
</servlet-mapping>
```

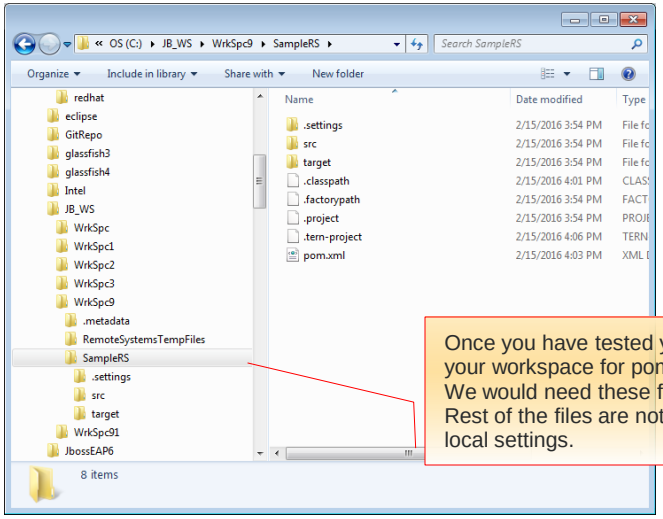
These options are available via the New->JAX-RS Resource wizard on JBDS.
pom.xml should have the dependency configuration for CDI, Annotations, JAX-RS.

JAX-RS Using JBDS

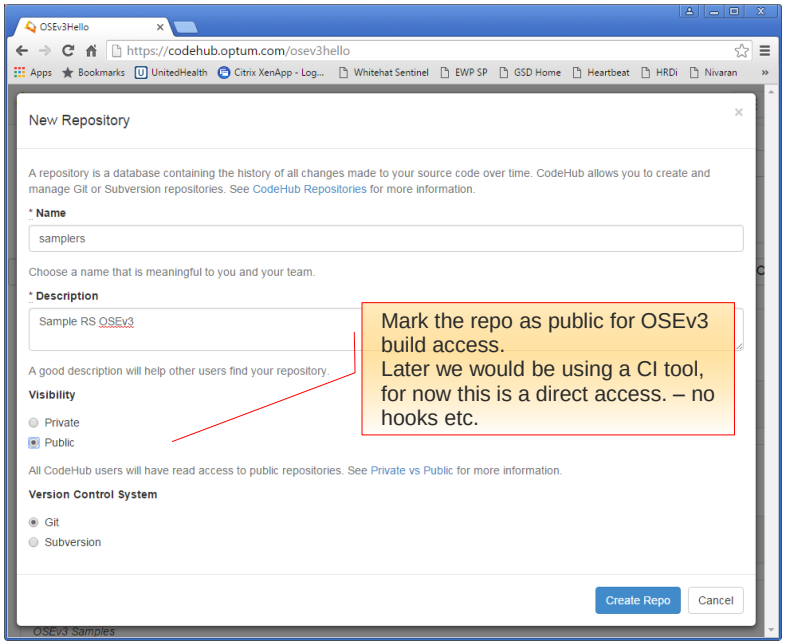




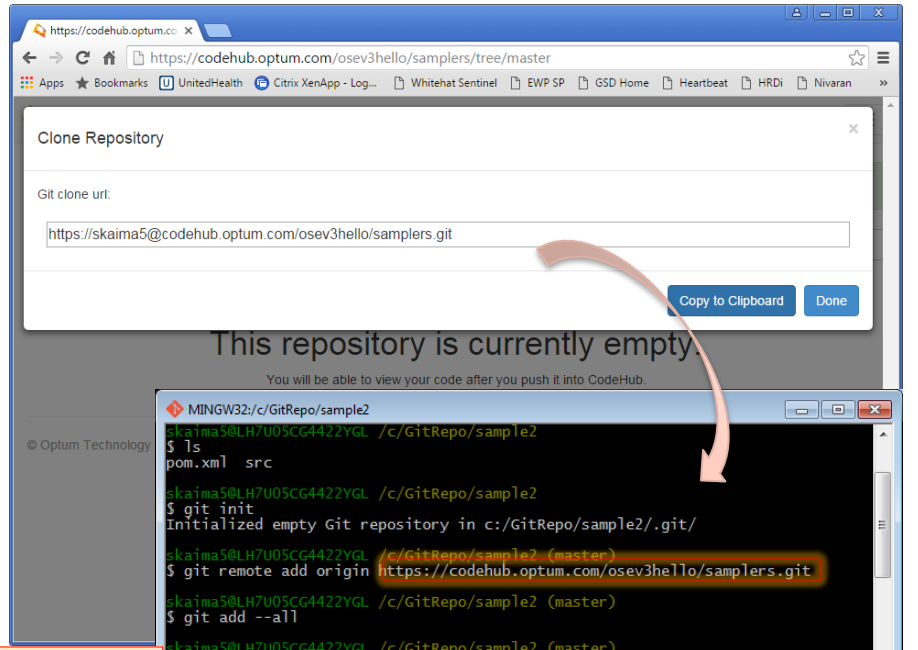


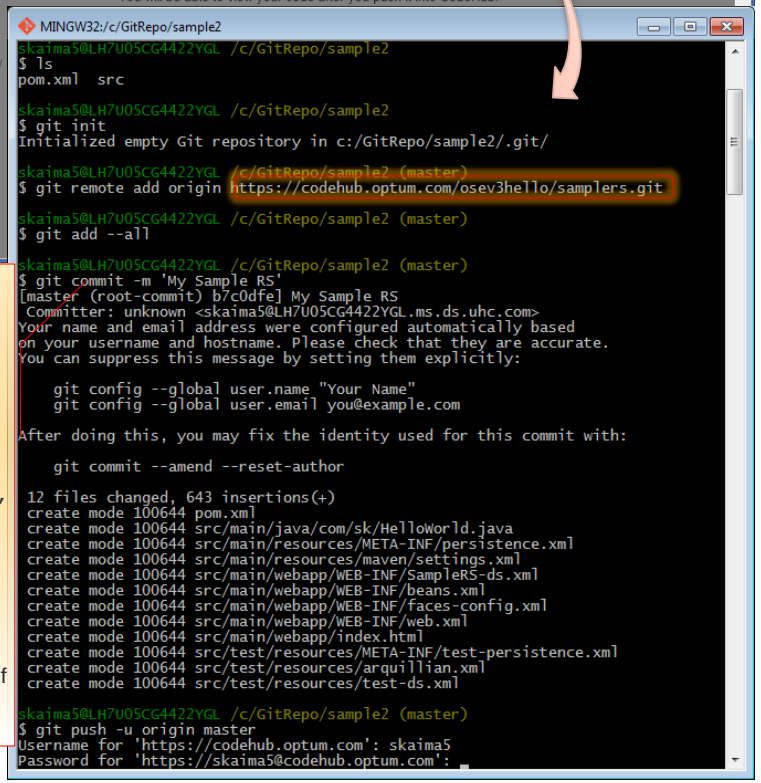


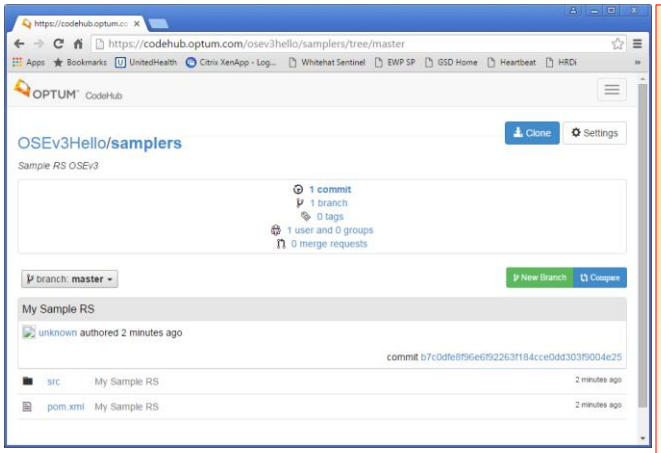
Git Repo & CodeHub



Mark the repo as public for OSEv3 build access.
Later we would be using a CI tool, for now this is a direct access. – no hooks etc.

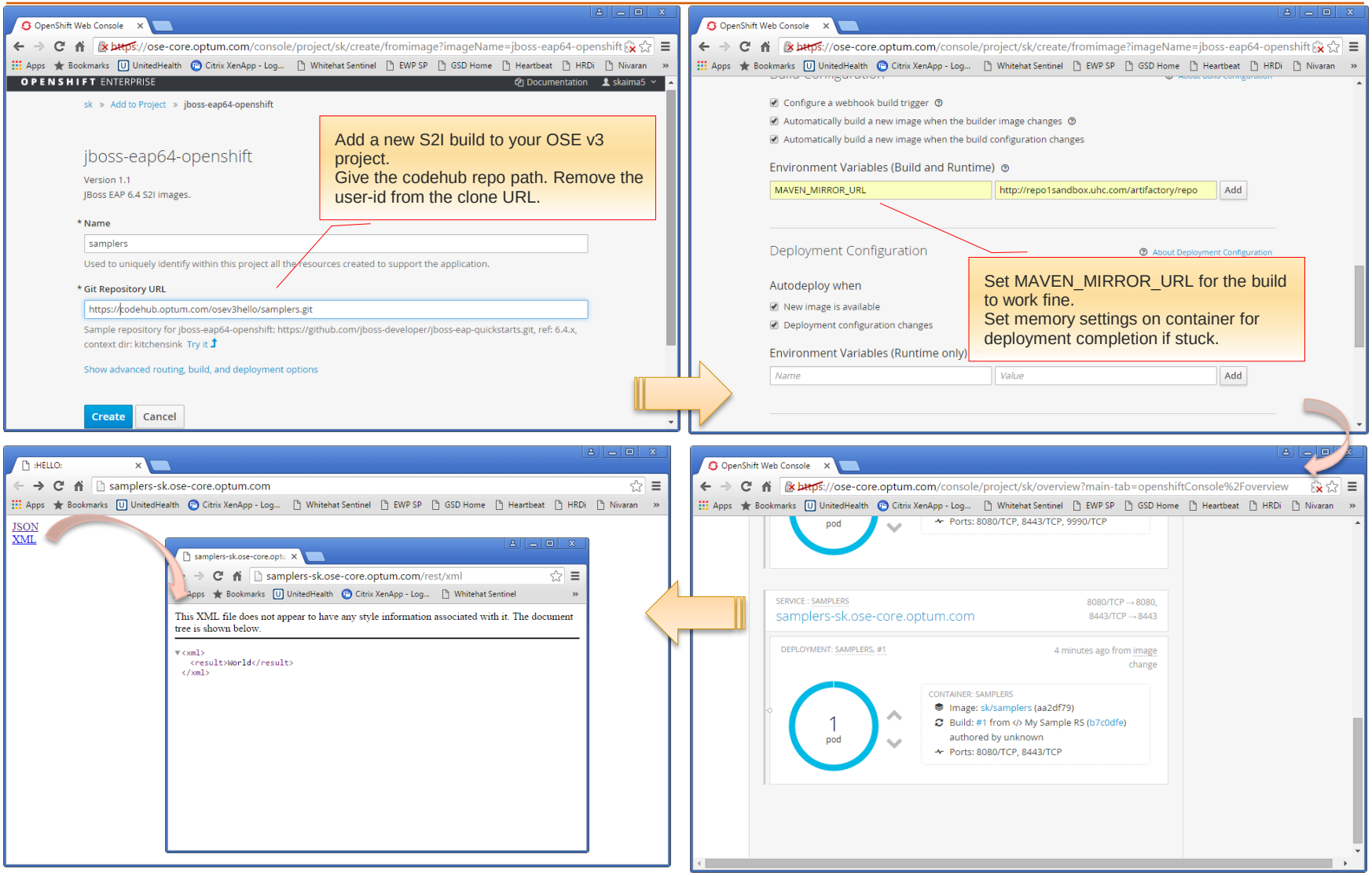






1. `git init`
   Initializes the git repo in the working directory.
2. `git remote add origin <URL>`
   Adds the remote server url for current git.
3. `Git add –all`
   Adds all the files to the repo
4. `Git commit –m 'comment'`
   Commits all changes to local repo
5. `Git push –u origin master`
   Pushes the changes to the remote server.
6. Verify on the codehub to see if your files are uploaded.

Build & Deploy



Repeat

1. The same steps work for any spring / hibernate / jsf web-app.
2. Try adding a Data Source in your web-app.
3. Try consuming a web-service & build a rest service around it.
4. Here are some useful links
   - JBoss Tools - Evaluate OpenShift 3 with the IDE
   - JBoss Tools - Create and Build OpenShift 3 Applications in the IDE
   - JBoss Tools - Set up and Remotely Monitor an OpenShift Database Server
   - JBoss Tools - Deploy a Docker Image to OpenShift 3

OPTUM™

Optum Global Solutions

Thank you.

Sreejith Kaimal
Sr.Architect
Provider Portal Applications