Services Rest

Kirk S. Kalvar

# Software Requirements

Java 1.6

SoapUI 4.5.1 - The SoapUI test run from junit is created using the SoapUI 4.5.1 Application externally

# Project Creation

mvn archetype:generate -DarchetypeCatalog=http://download.java.net/maven/2

mvn eclipse:eclipse -Dwtpversion=2.0

# Test Service Online

Once deployed using eclipse you can use a browser and enter the following url to see if the application is online:

http://localhost:8080/ServicesRest/

# Eclipse jboss deployment environment setup

## Download jboss 5.1 GA

Menu Bar

Preferences/Server/Runtime Environments

http://sourceforge.net/projects/jboss/files/JBoss/JBoss-5.1.0.GA/jboss-5.1.0.GA.zip/download

## Publish Java Class Changes

In Eclipse, double click jboss server in the Servers view. Under Publishing select Automatically publish when resource changes. Under Application reload behavior, select Customize application reload behavior and enter the following regular expression pattern:

## \.jar$|\.class$|\.jsp$

## Build Automatically

Menu Bar

Project/Build Automatically

## Eclipse JBoss Admin Console

Using a browser enter the following url:

http://localhost:8080/admin-console

userid: admin

password: admin

# Maven deployment environment dependency

Since github will not allow you to check in compressed archives (zip, tar.gz, etc), you'll need to download jboss-5.1.0-GA manually and placed it in the following workspace directory:

<eclipse workspace>/ServicesRest/server/test/src/test/jboss

# Testing

This project has a couple of ways to be tested. There is a maven built-in junit test which deploys the war to a cargo-plug-in supported version of jboss 5.1.0.GA and then runs a soap test via junit against the jboss deployed war.

Since this is a Eclipse Dynamic Web project you can also test it within Eclipse environment using a jboss 5.1 Runtime deployment controlled by Eclipse's Server View.

Additionally you can manually deploy ServicesRestWAR-0.0.1-SNAPSHOT.war to jboss-5.1.0.GA/server/default/deploy