**SoapUI-Maven-Jenkins Integration:**

**This configuration is very basic and provides the skeletal structure for integration between Jenkins and SoapUI tests configured within a Maven project. Here we are using the SoapUI free version.**

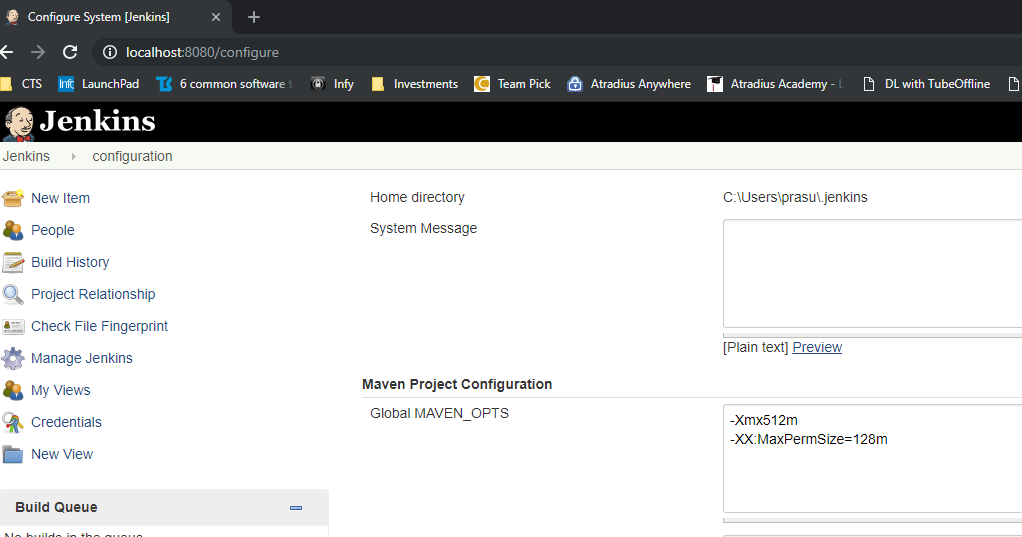
**For SoapUI Pro licensed version, Jenkins provides a separate Add-In. Pro version is not covered here.**

**This configuration can be customized to include for different environments, databases, reporting formats, scheduling, etc.**

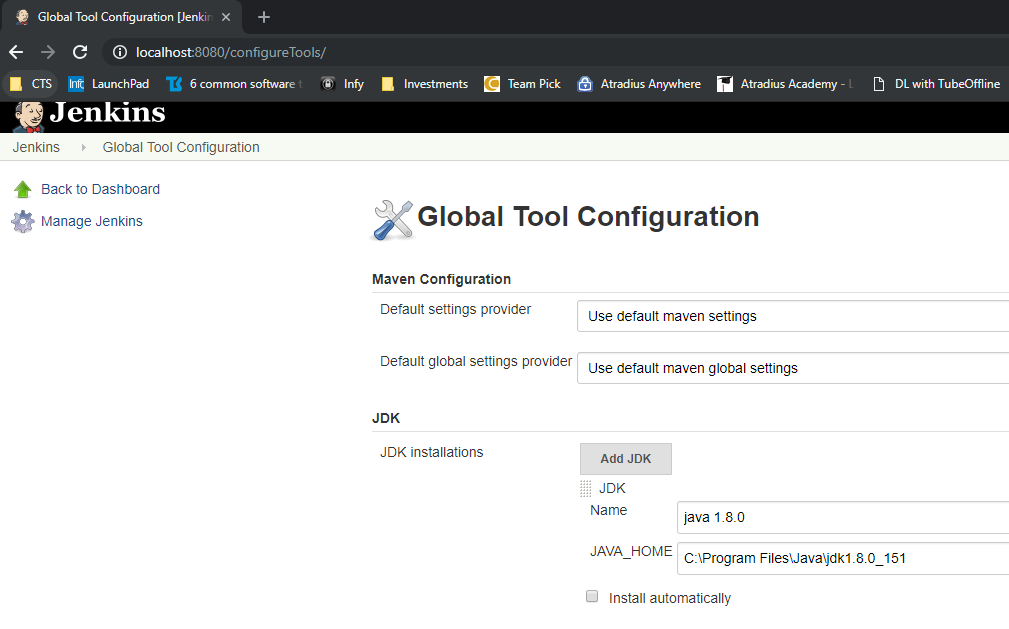
**Jenkins Configuration for Maven:**

-Xmx512m

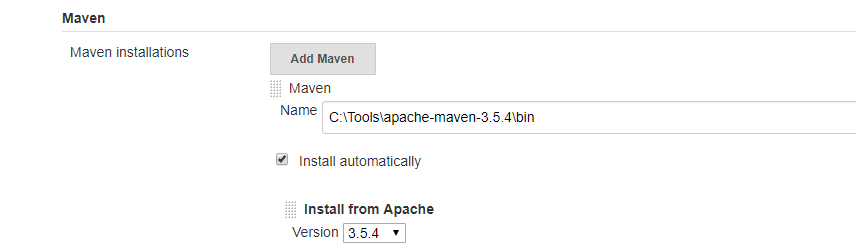
-XX:MaxPermSize=128m



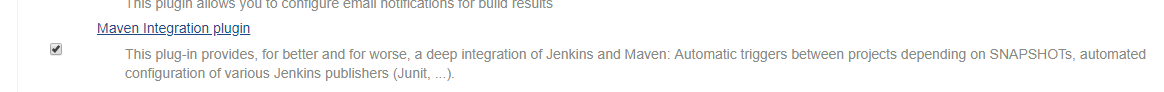
**JDK Path**



**Maven Path**

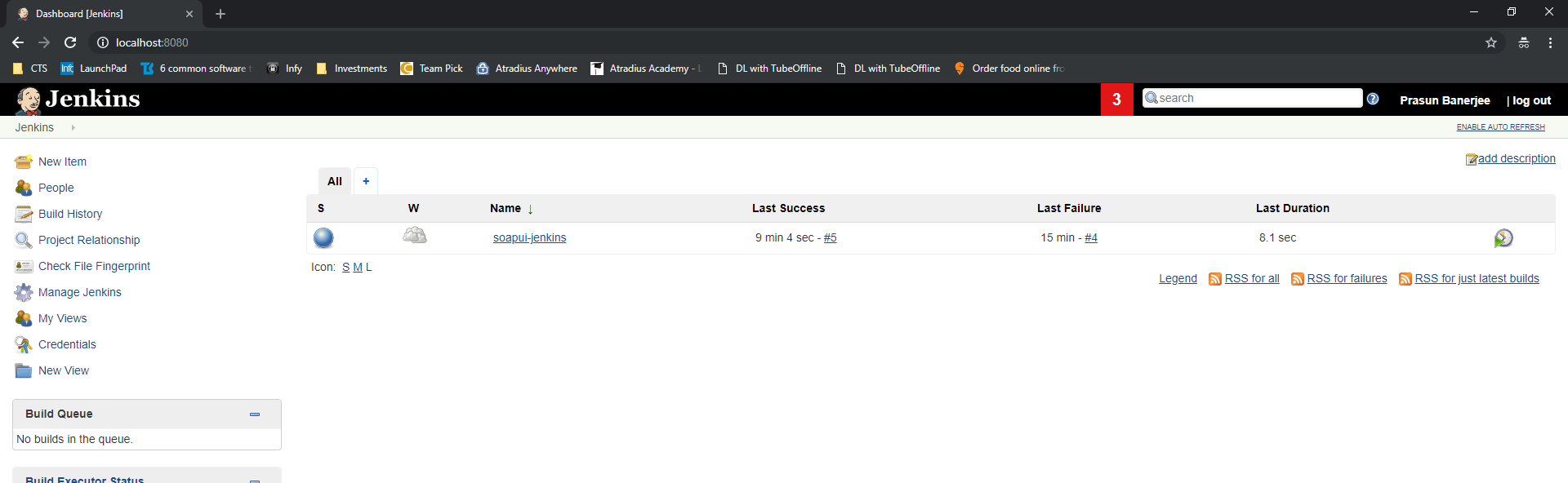


**Install Maven Integration Plugin in Jenkins**



**Jenkins Build Configuration**

Create a Maven Project type in Jenkins

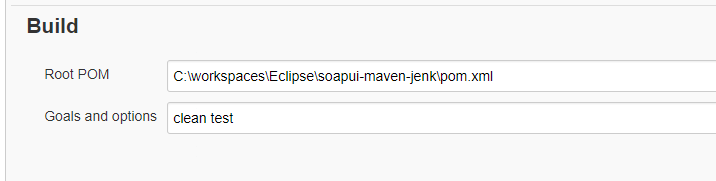


**Basic Build Configuration:**

**Provide the path of the POM File**

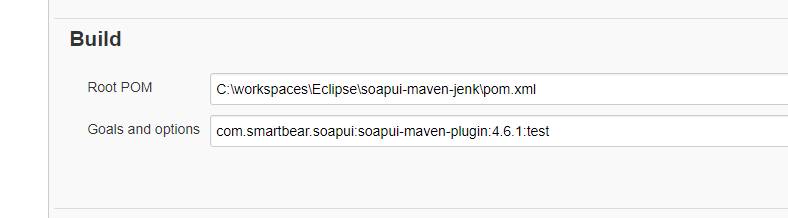
**Provide the Maven Command to run the tests**

**mvn clean test**

****

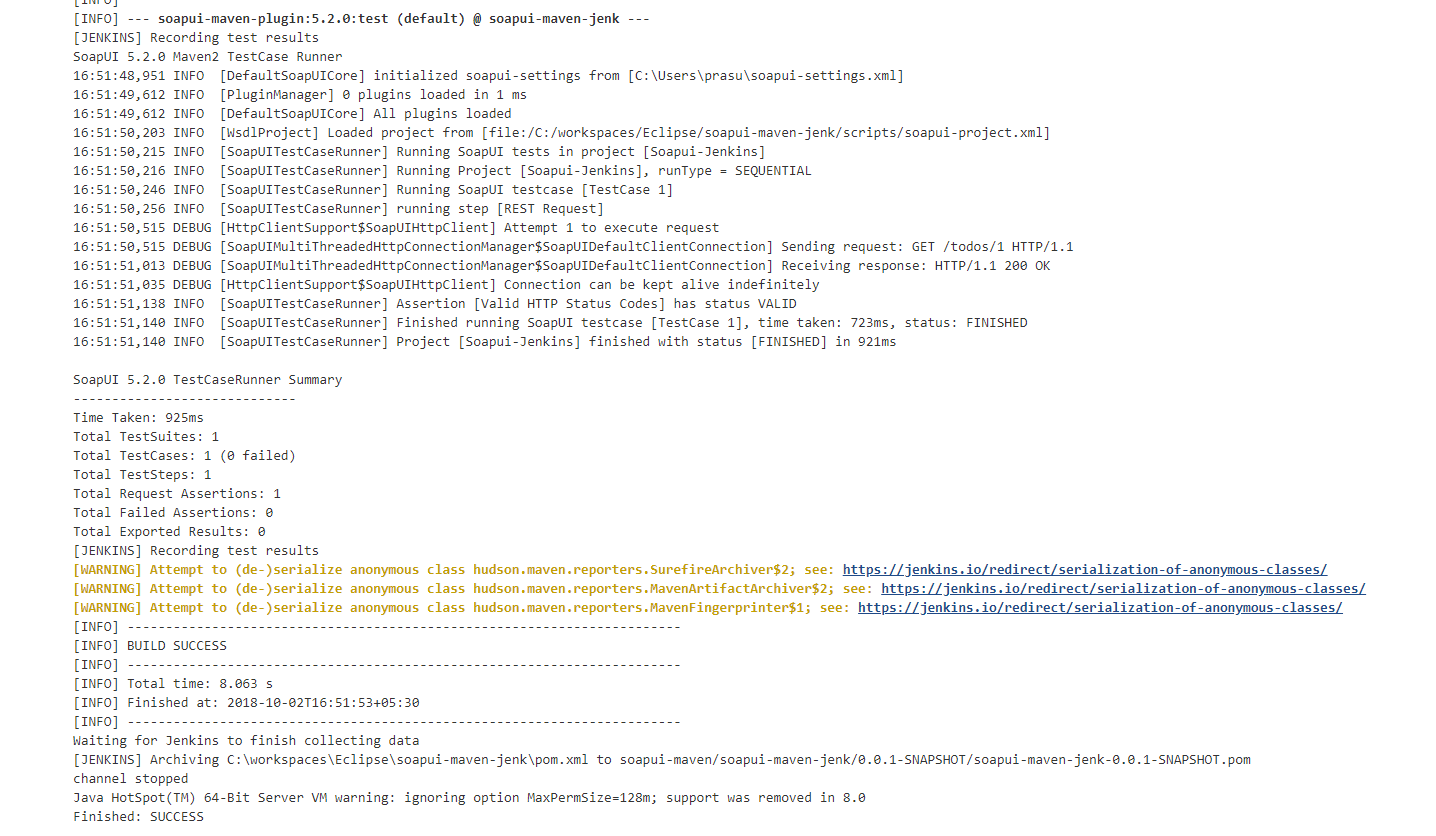
**You can also provide the below command to run the test. This allows you to control and override the soapui-maven-plugin directly without the need to update the POM file**

com.smartbear.soapui:soapui-maven-plugin:4.6.1:test



**Execution and Console Output:**

**Console Output contains the details of the Soap UI Service Tests execution.**

****

**Additional Details:**

**Adding the Oracle JARs path to the POM file**

<dependency>

<groupId>com.oracle</groupId>

<artifactId>ojdbc14</artifactId>

<version>10.2.0.3.0</version>

<scope>system</scope>

<systemPath>${basedir}/ext/ojdbc14-1.0.jar</systemPath>

</dependency>

<dependency>

<groupId>com.oracle</groupId>

<artifactId>orai1</artifactId>

<version>11.2.0</version>

<scope>system</scope>

<systemPath>${basedir}/ext/orai18n.jar</systemPath>

</dependency>

<dependency>

<groupId>com.oracle</groupId>

<artifactId>ojdbc</artifactId>

<version>11.2.0</version>

<scope>system</scope>

<systemPath>${basedir}/ext/ojdbc6.jar</systemPath>

</dependency>

For each environment, we need to configure a profile in POM file

Each profile i.e. environment will have its own configurations.

Environment parameter (id) will need to be passed in the mvn command in Jenkins to execute the tests for a specific environment.

<profiles>

<profile> <!-- test -->

<id>test1</id>

<build>

<plugins>

<plugin>

<executions>

<execution>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

</profiles>

</project>

Once you add a profile then the maven command to run the tests will change to:

mvn clean test –P test1