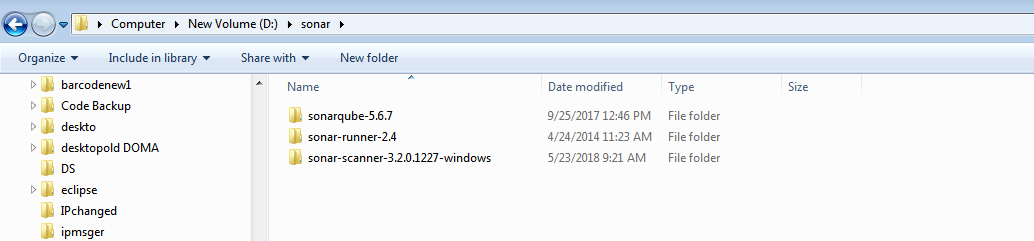
**Download:**

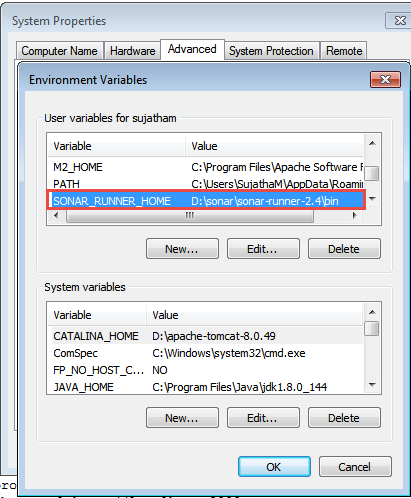
netstat -ano | findstr :9000

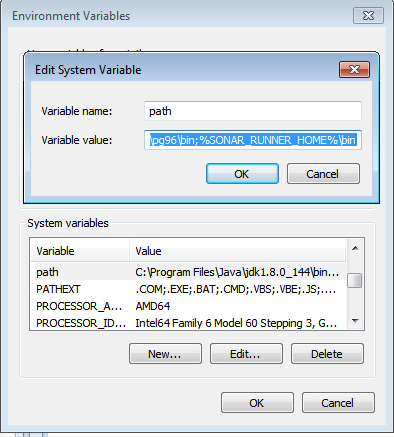
taskkill /PID 4828 /F

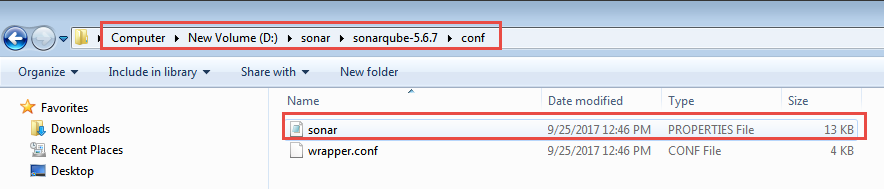
clean org.jacoco:jacoco-maven-plugin:prepare-agent package sonar:sonar

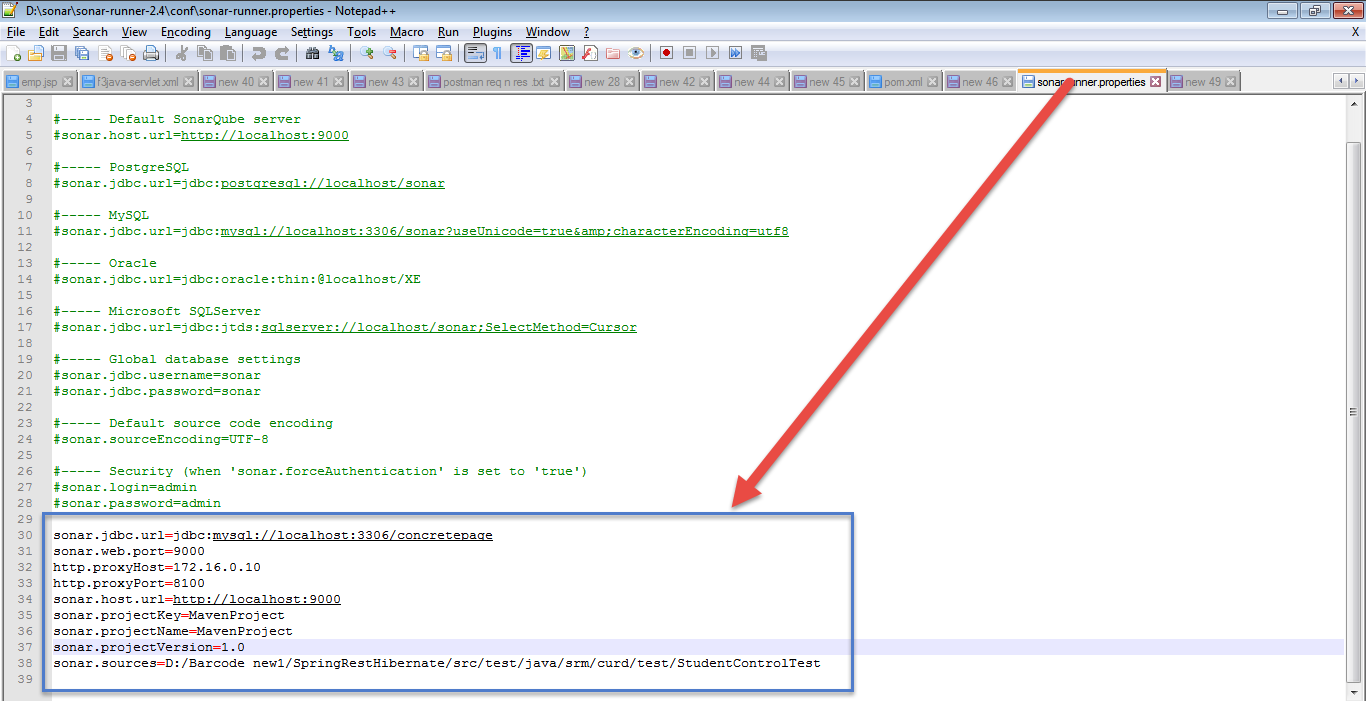
SonarQube n Runner

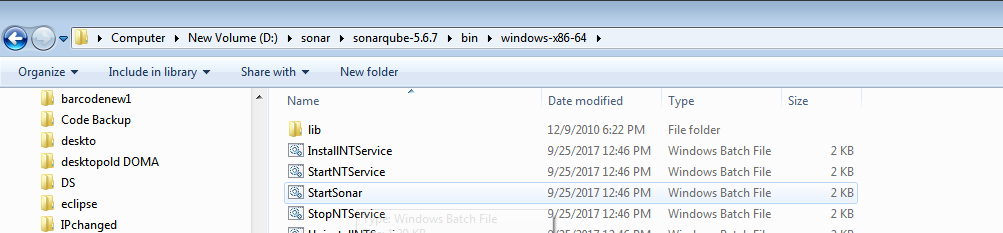


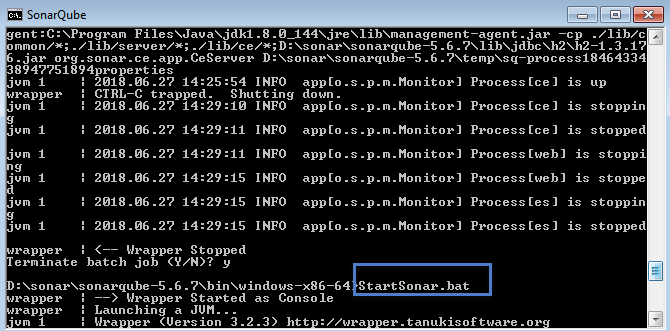










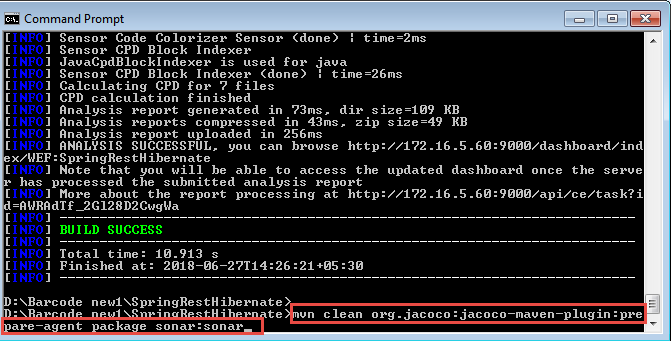


set global max\_allowed\_packet = 100\*1024\*1024;

**mvn clean org.jacoco:jacoco-maven-plugin:prepare-agent package sonar:sonar**

**mvn clean org.jacoco:jacoco-maven-plugin:prepare-agent package sonar:sonar**

**-Dmaven.clean.failOnError=false**



Pom.xml

<!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-surefire-plugin -->

<dependency>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.12.4</version>

</dependency>

<dependency>

<groupId>org.codehaus.mojo</groupId>

<artifactId>sonar-maven-plugin</artifactId>

<version>3.0.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.sonarsource.scanner.maven/sonar-maven-plugin -->

<dependency>

<groupId>org.sonarsource.scanner.maven</groupId>

<artifactId>sonar-maven-plugin</artifactId>

<version>3.0.1</version>

</dependency>

**In Testcase @Ignore to be added**

package curd.test;

@Ignore

@WebAppConfiguration

@RunWith(SpringJUnit4ClassRunner.class)

@ContextConfiguration({"classpath:Spring-Datasource.xml"})

public class StudentControlTest {

--

}

**Hint:**

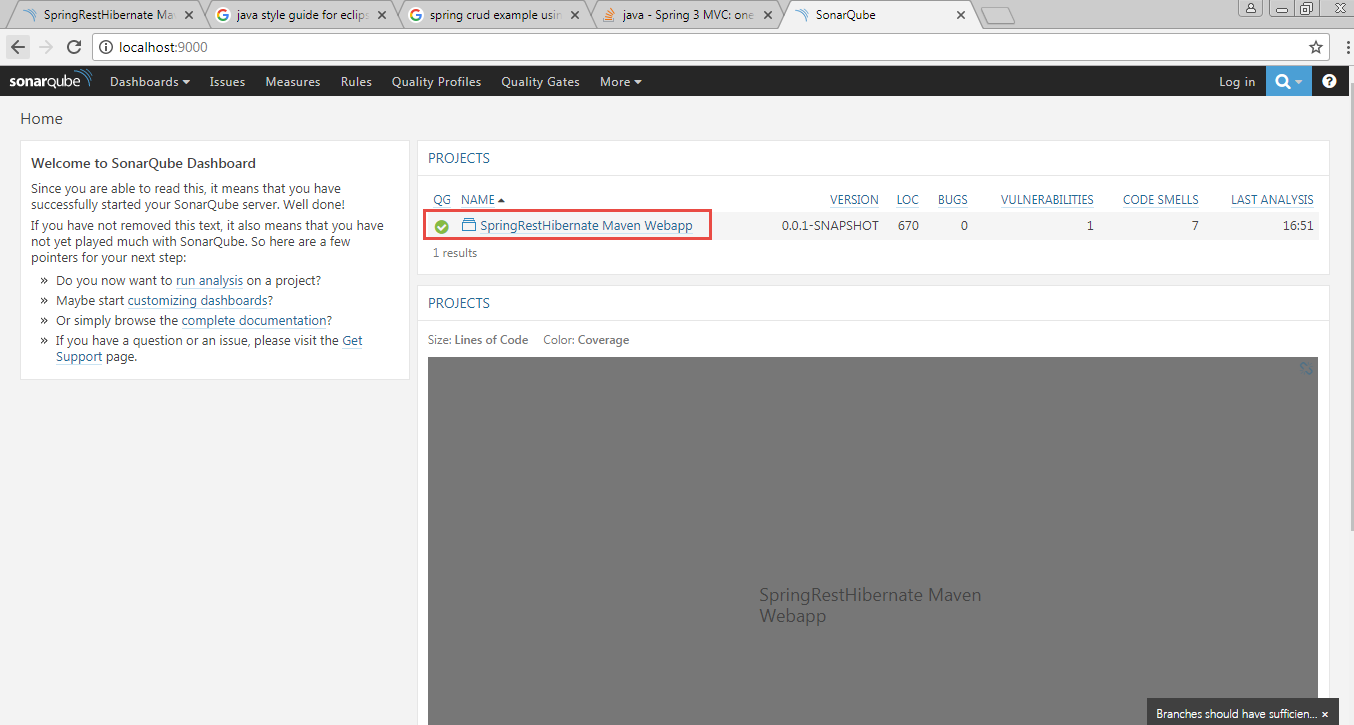
Eclipse marketplace ->Install Sonar Lint –right click Project –sonar lint-> **analyze option**

mvn dependency::tree

mvn clean install –DskipTests

set global max\_allowed\_packet = 100\*1024\*1024;

mvn clean org.jacoco:jacoco-maven-plugin:prepare-agent package sonar:sonar



**Reffered:**

<https://stackoverflow.com/questions/25897568/how-to-create-environment-variable-sonar-runner-home-for-sonar-eclipse-in-win>

<http://ashismo.github.io/java-code%20quality%20analyzer/2015/05/12/SONAR-the-Java-Code-Analyzer>

<https://www.youtube.com/watch?v=Gu9skLzRSao>

<https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner+for+Maven>

<http://ashismo.github.io/java-code%20quality%20analyzer/2015/05/14/Java-Code-Coverage-Report-in-SONAR-Dashboard-powered-by-JaCoCo>

<https://www.petrikainulainen.net/programming/maven/creating-code-coverage-reports-for-unit-and-integration-tests-with-the-jacoco-maven-plugin/>

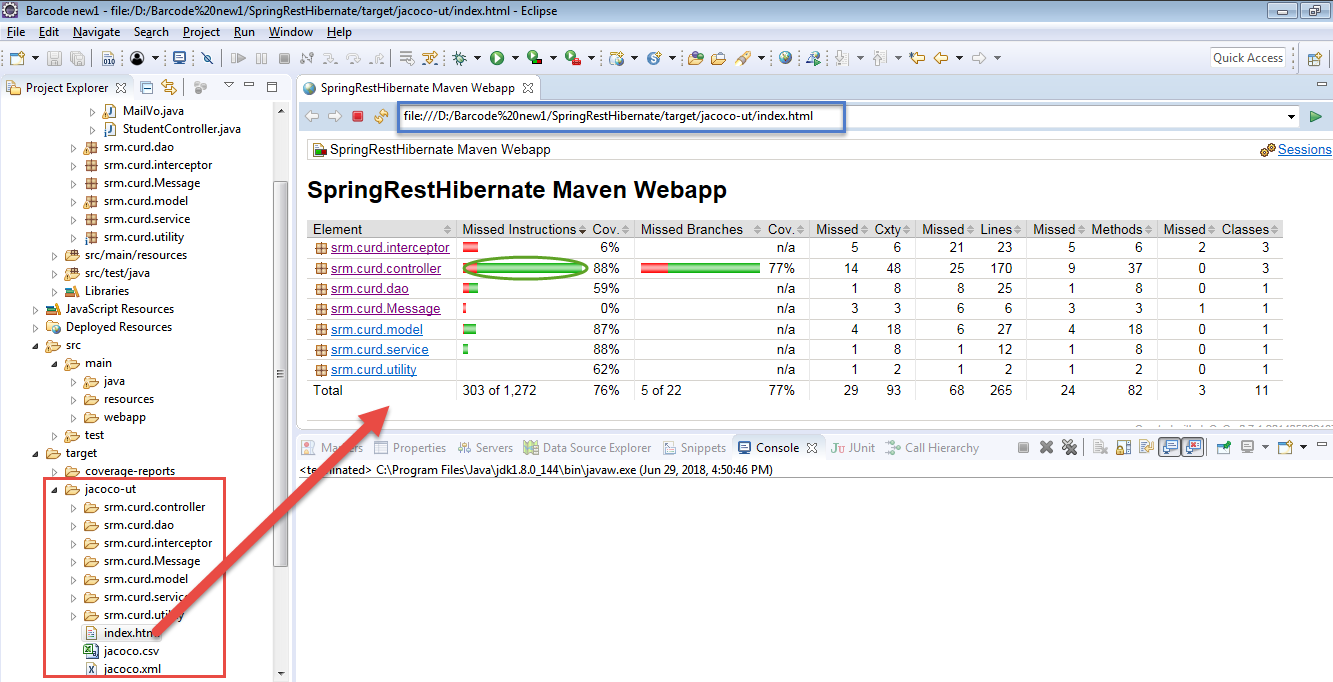
**To know java process in cmd window**

jcmd –l

taskkill/pid 6427 /f

**check style**

<https://www.seas.upenn.edu/~cis121/current/java_style_guide.html>



**To improve code coverage all the methods to be tested in test class**

**POM.XML for code coverage report and test case integrartion with Project all the reports will be stored in target folder jacoo-ut \index.htl –open with –web it ll display the Report of code coverage in all the package structure**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>WEF</groupId>

<artifactId>SpringRestHibernate</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>SpringRestHibernate Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-compiler-plugin -->

<dependency>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/cglib/cglib -->

<dependency>

<groupId>cglib</groupId>

<artifactId>cglib</artifactId>

<version>3.2.6</version>

</dependency>

<!-- https://mvnrepository.com/artifact/junit/junit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>4.3.15.RELEASE</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest-all</artifactId>

<version>1.3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.3.1.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.9.5</version>

</dependency>

<dependency>

<groupId>org.apache.velocity</groupId>

<artifactId>velocity</artifactId>

<version>1.7</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api -->

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>javax.servlet.jsp-api</artifactId>

<version>2.3.2-b02</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-tx -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/postgresql/postgresql -->

<!-- <dependency> <groupId>postgresql</groupId> <artifactId>postgresql</artifactId>

<version>9.1-901.jdbc4</version> </dependency> -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.10</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.9.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/log4j/log4j -->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.mail/mail -->

<dependency>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

<version>1.4.4</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.sonarsource.scanner.maven/sonar-maven-plugin -->

<dependency>

<groupId>org.sonarsource.scanner.maven</groupId>

<artifactId>sonar-maven-plugin</artifactId>

<version>3.0.1</version>

</dependency>

</dependencies>

<build>

<finalName>SpringRestHibernate</finalName>

<!-- Configuring The JaCoCo Maven Plugin -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

<configuration>

<failOnError>false</failOnError>

</configuration>

</plugin>

<!-- Configuring The JaCoCo Maven Plugin -->

<plugin>

<groupId>org.jacoco</groupId>

<artifactId>jacoco-maven-plugin</artifactId>

<version>0.7.1.201405082137</version>

<executions>

<!-- Prepares the property pointing to the JaCoCo runtime agent which

is passed as VM argument when Maven the Surefire plugin is executed. -->

<execution>

<id>pre-unit-test</id>

<goals>

<goal>prepare-agent</goal>

</goals>

<configuration>

<!-- Sets the path to the file which contains the execution data. -->

<destFile>${project.build.directory}/coverage-reports/jacoco-ut.exec</destFile>

<!-- Sets the name of the property containing the settings for JaCoCo

runtime agent. -->

<propertyName>surefireArgLine</propertyName>

</configuration>

</execution>

<!-- Ensures that the code coverage report for unit tests is created

after unit tests have been run. -->

<execution>

<id>post-unit-test</id>

<phase>test</phase>

<goals>

<goal>report</goal>

</goals>

<configuration>

<!-- Sets the path to the file which contains the execution data. -->

<dataFile>${project.build.directory}/coverage-reports/jacoco-ut.exec</dataFile>

<!-- Sets the output directory for the code coverage report. -->

<outputDirectory>${project.build.directory}/jacoco-ut</outputDirectory>

</configuration>

</execution>

</executions>

</plugin>

<!-- Configuring The Maven Surefire Plugin -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.12.4</version>

<configuration>

<forkCount>3</forkCount>

<reuseForks>true</reuseForks>

<argLine>-Xmx1024m -XX:MaxPermSize=256m</argLine>

<testFailureIgnore>true</testFailureIgnore>

<!-- Sets the VM argument line used when unit tests are run. -->

<argLine>${surefireArgLine}</argLine>

<!-- Skips unit tests if the value of skip.unit.tests property is true -->

<skipTests>${skip.unit.tests}</skipTests>

</configuration>

</plugin>

<!-- Configuring The Maven Failsafe Plugin -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-failsafe-plugin</artifactId>

<version>2.15</version>

<executions>

<!-- Ensures that both integration-test and verify goals of the Failsafe

Maven plugin are executed. -->

<execution>

<id>integration-tests</id>

<goals>

<goal>integration-test</goal>

<goal>verify</goal>

</goals>

<configuration>

<test>\*\*/\*.java</test>

<!-- Sets the VM argument line used when integration tests are run. -->

<argLine>${failsafeArgLine}</argLine>

<!-- Skips integration tests if the value of skip.integration.tests

property is true -->

<skipTests>${skip.integration.tests}</skipTests>

</configuration>

</execution>

</executions>

</plugin> </plugins></build></project>

Issue Faced:

Maven clean –Not Worked soln :Added plugin n then jcmd l used to know the running java (Process)

taskkill/pid 6427 /f

used then restarted eclipse

Additional note :

**Validation :**

**if** (Long.*valueOf*(id) == **null**) {

**return** **null**;

}

**if** (student == **null**) {

**return** **null**;

} **else** {

}

**if** (Long.*valueOf*(id) == **null**) {

**return** ResponseEntity.*ok*().body("User could not be deleted .");

} **else** {

**return parenMailids != null ? parenMailids : null;**

github revathi

https://github.com/revathiks/my-angular/blob/master/src/app/app-routing/app-routing.module.ts

**Failed to goal Maven failsafe plugin exception**

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.0</version>

<configuration>

<argLine>--add-modules java.base ${argLine} -Xmx1024m -XX:MaxPermSize=256m</argLine>

<forkCount>3</forkCount>

<reuseForks>true</reuseForks>

<testFailureIgnore>true</testFailureIgnore>

JWT TOKEN REFERENCE

[**https://www.boraji.com/spring-mvc-5-spring-security-5-hibernate-5-example**](https://www.boraji.com/spring-mvc-5-spring-security-5-hibernate-5-example)

**create table USERS\_SECURITY(**

**username varchar(50) not null primary key,**

**password varchar(100) not null,**

**enabled boolean not null**

**);**

**create table AUTHORITIES\_SECURITY (**

**username varchar(50) not null,**

**authority varchar(50) not null,**

**constraint fk\_authorities\_users foreign key(username) references USERS\_SECURITY(username)**

**);**

**insert into USERS\_SECURITY values('sam','sam',true);**

**insert into AUTHORITIES\_SECURITY values('sam','admin');**

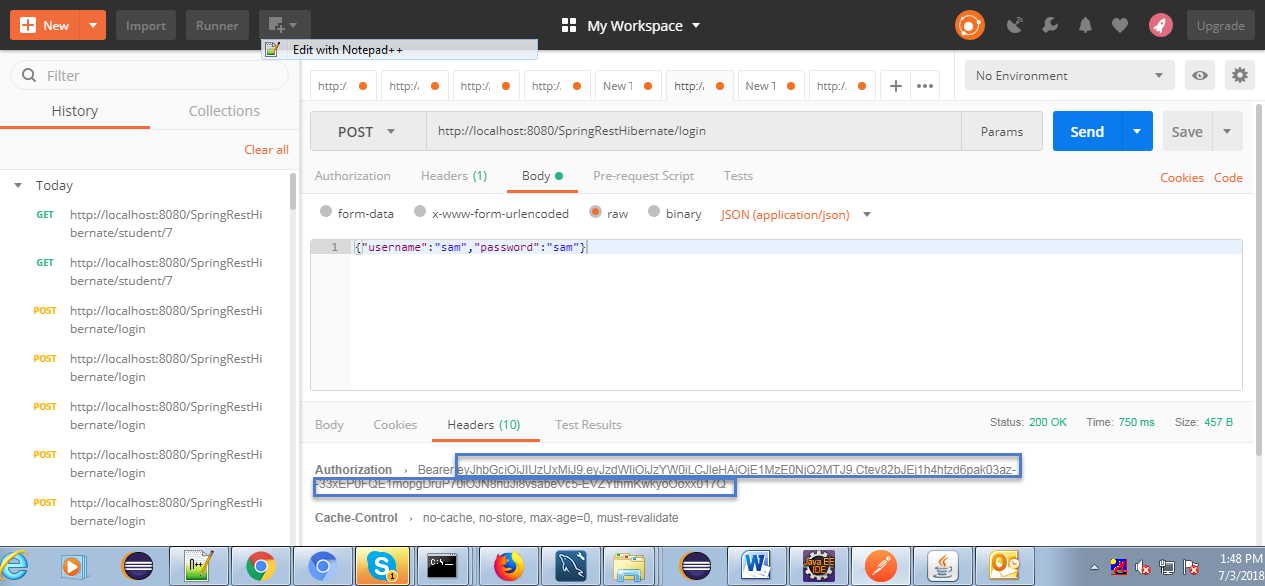
1.Authentication :

Post

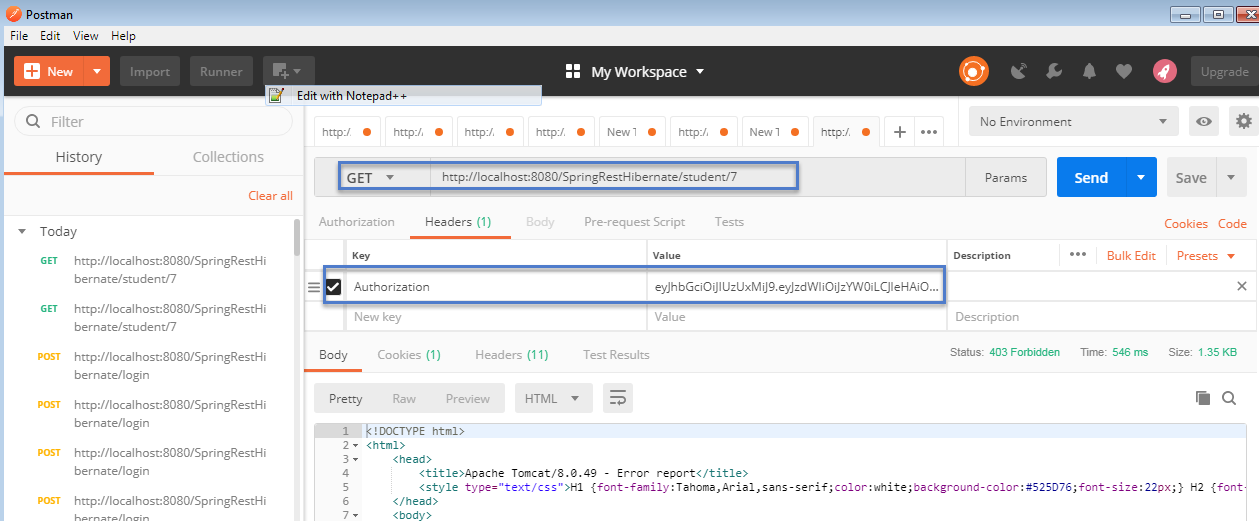
<http://localhost:8080/SpringRestHibernate/login>

**{"username":"sam","password":"sam"}**

**2.JWT Token Generation :Service calling by /login**



**Authenticated user only can access the other functionalities like create,update..etc**



[**http://www.in28minutes.com/spring-boot-maven-eclipse-troubleshooting-guide-and-faq**](http://www.in28minutes.com/spring-boot-maven-eclipse-troubleshooting-guide-and-faq)

**Integration Testing Refer**

[**https://leongfeng.github.io/2016/11/05/springboot-mockmvc&mockito/**](https://leongfeng.github.io/2016/11/05/springboot-mockmvc&mockito/)

[**https://jeroenbellen.com/testing-spring-rest-controllers/**](https://jeroenbellen.com/testing-spring-rest-controllers/)

[**https://www.logicbig.com/tutorials/spring-framework/spring-web-mvc/json-message-object-conversion.html**](https://www.logicbig.com/tutorials/spring-framework/spring-web-mvc/json-message-object-conversion.html)

[**https://memorynotfound.com/unit-test-spring-mvc-rest-service-junit-mockito/**](https://memorynotfound.com/unit-test-spring-mvc-rest-service-junit-mockito/)

**Resource bundle Properties**

/\* InputStream input = getClass().getClassLoader()

.getResourceAsStream("erp.properties");

Properties prop = new Properties();

if(input != null) {

prop.load(input);\*/

**Resource bundle Properties File Reading Without extension of .properties ie erp**

ResourceBundle rb = ResourceBundle.*getBundle*("erp");

System.***out***.println(rb.getString("jwtloginerrormsg")+" \*\*\*\* "+ rb.getString("jwtloginerrorcode"));