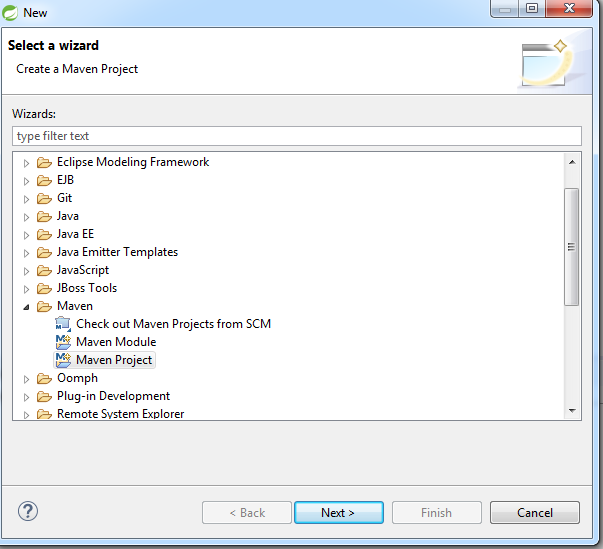
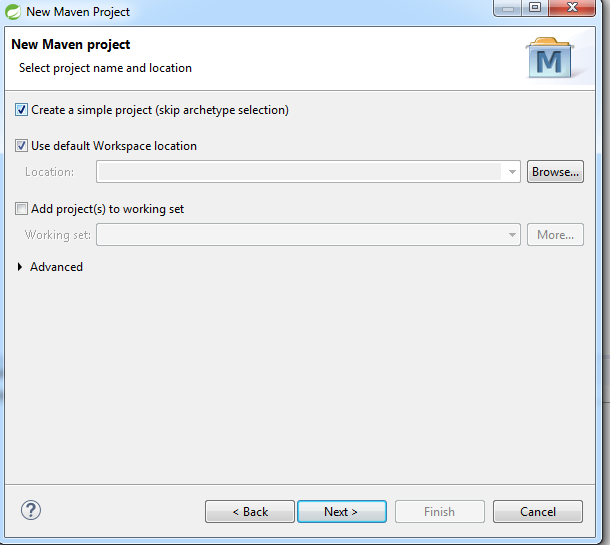
Spring Tool Suite Steps

1. Steps To Create Simple Spring Application In STS USING Maven Project.

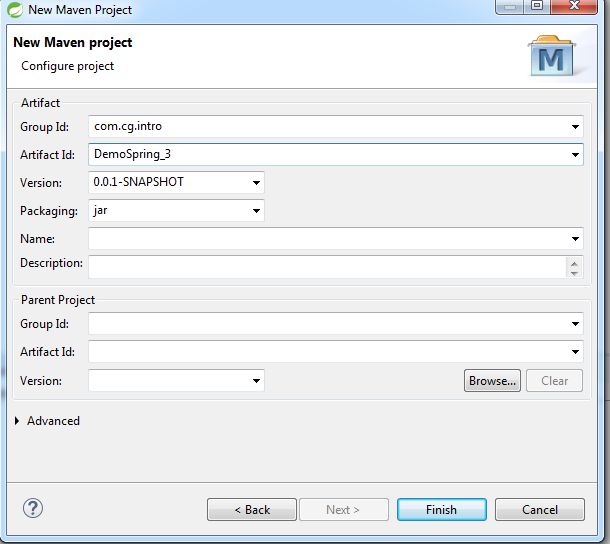
Step:-1) Go To New-Others- Create Simple Maven Project In STS.



Step:-2) Click Next – Next - button



Step:- 3) Give Group Id and Artifact Id and Click on Finish Button.



Step:- 4) In Maven Project initial pom.xml file created is as follows.

<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/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.cg.intro</groupId>

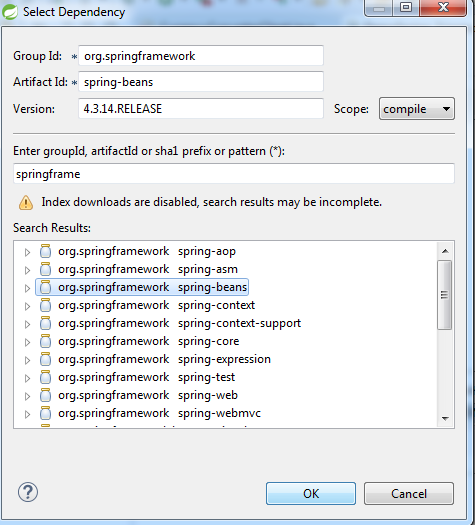
<artifactId>DemoSpring\_3</artifactId>

<version>0.0.1-SNAPSHOT</version>

</project>

Step: - 5) Add the spring related dependencies in Pom.xml as follows.

* + 1. Open pom.xml- > Click on Dependency tab -> Click on Add Button.
    2. Add the following dependencies and click on OK button



After Adding all dependencies pom.xml look like as follows.

<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/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cg.intro</groupId>

<artifactId>DemoSpring\_3</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>4.3.14.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.3.14.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>4.3.14.RELEASE</version>

</dependency>

</dependencies>

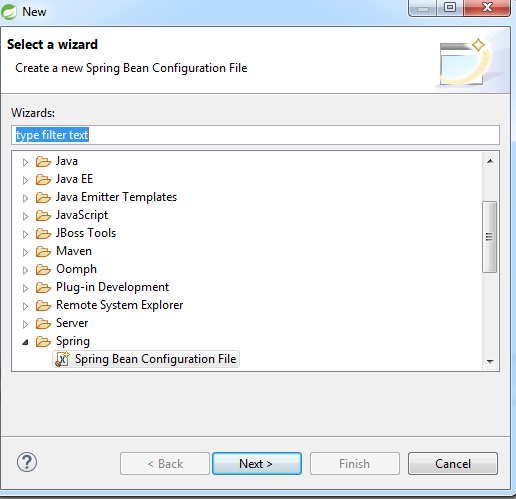
</project>

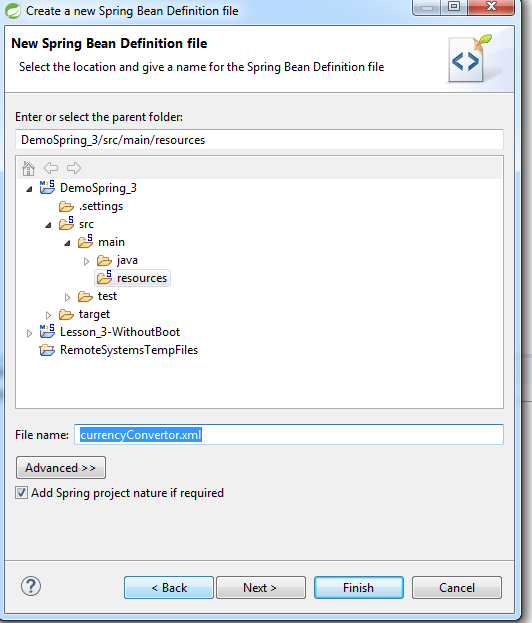
Note:-In case of any errors in pom.xml.

Select the Project-Right Click on Maven- Update Project and –Select Force Update option

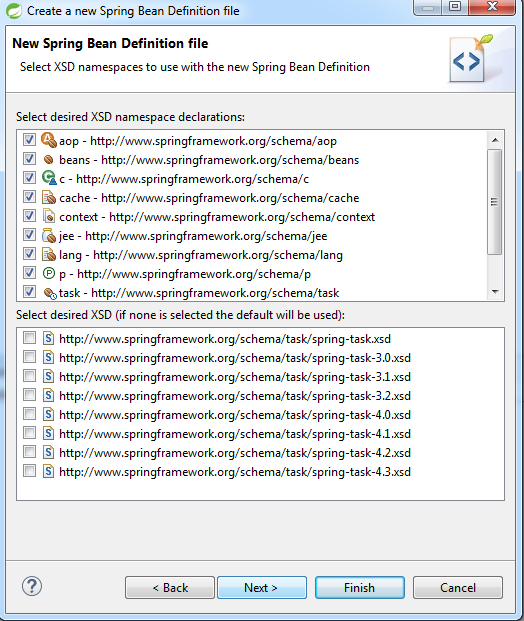
Use MVN Installed Option in case of any errors

Step:- 6) After Creating the Maven Project -Go to project’s src – main - resources folder and Right Click on it – Select New - Others - Spring - Spring bean Configuration File- Click on Next





Give a Xml file Name -Click On next



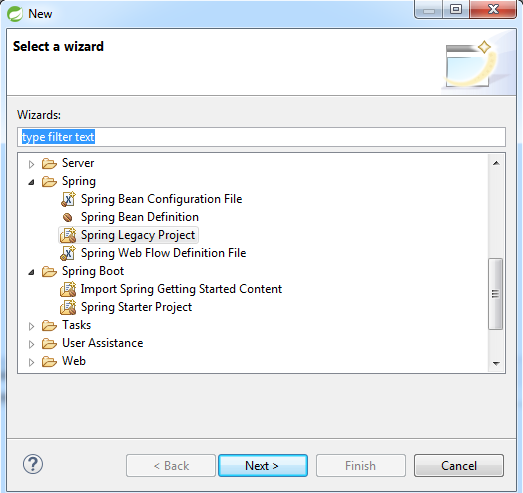
Select namespaces required and Click on finish.

Step: -7) Create java classes in Project\src\main\java folder .

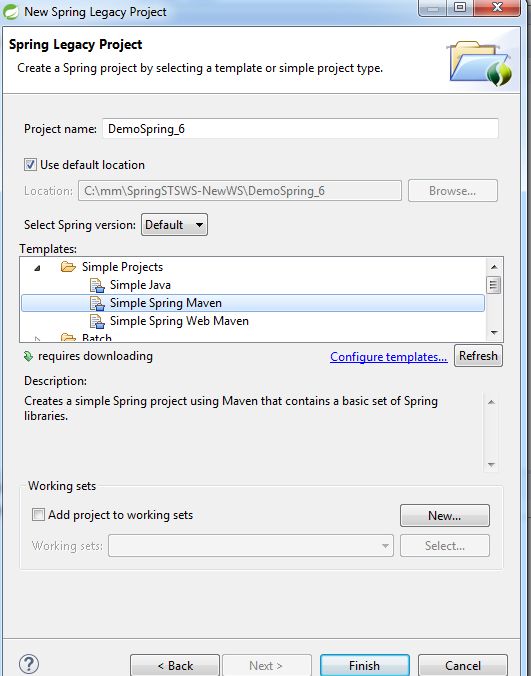
1. Create Simple Spring project in STS Using Spring legacy project

Note:-Initially May Require Internet Connection For Local Repository

Step:-1) Go To New -Others - Spring Legacy Project - Click on Next.

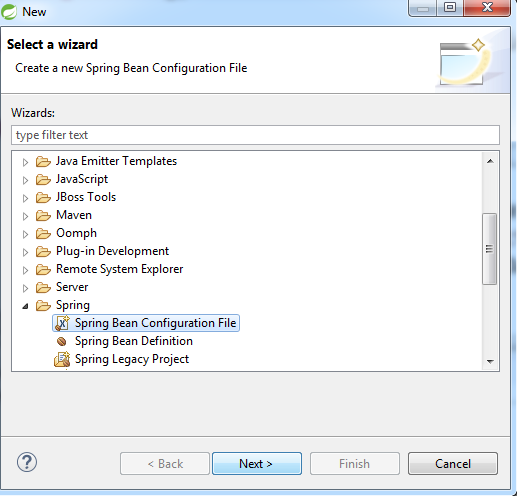


Step :-2 ) Give Project Name - Select Simple Spring Maven - Click On Finish Option.

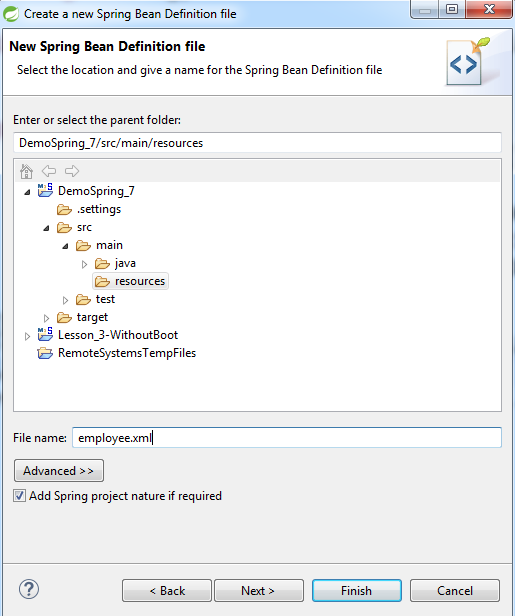


Step :-3) After Creation of Simple Spring Maven project create java classess in project\src\main\java folder and xml files in the projecs\src\main\resources folderes as follows.

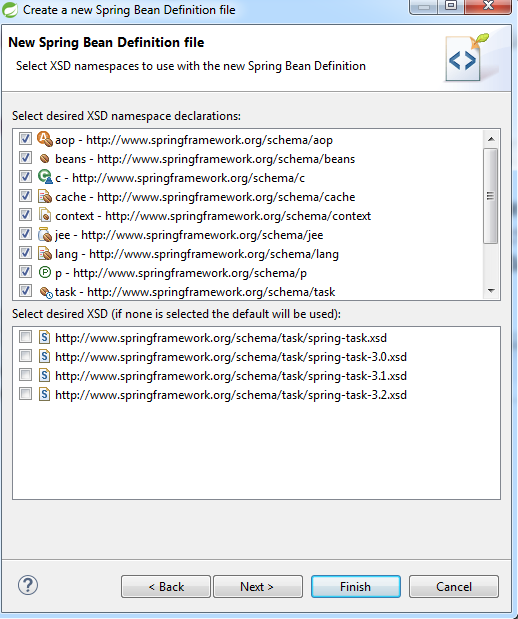
Go To New –Others - Spring - Spring Bean Configuration Files - Click On Next



Give xml File Name - Click On Next as follows



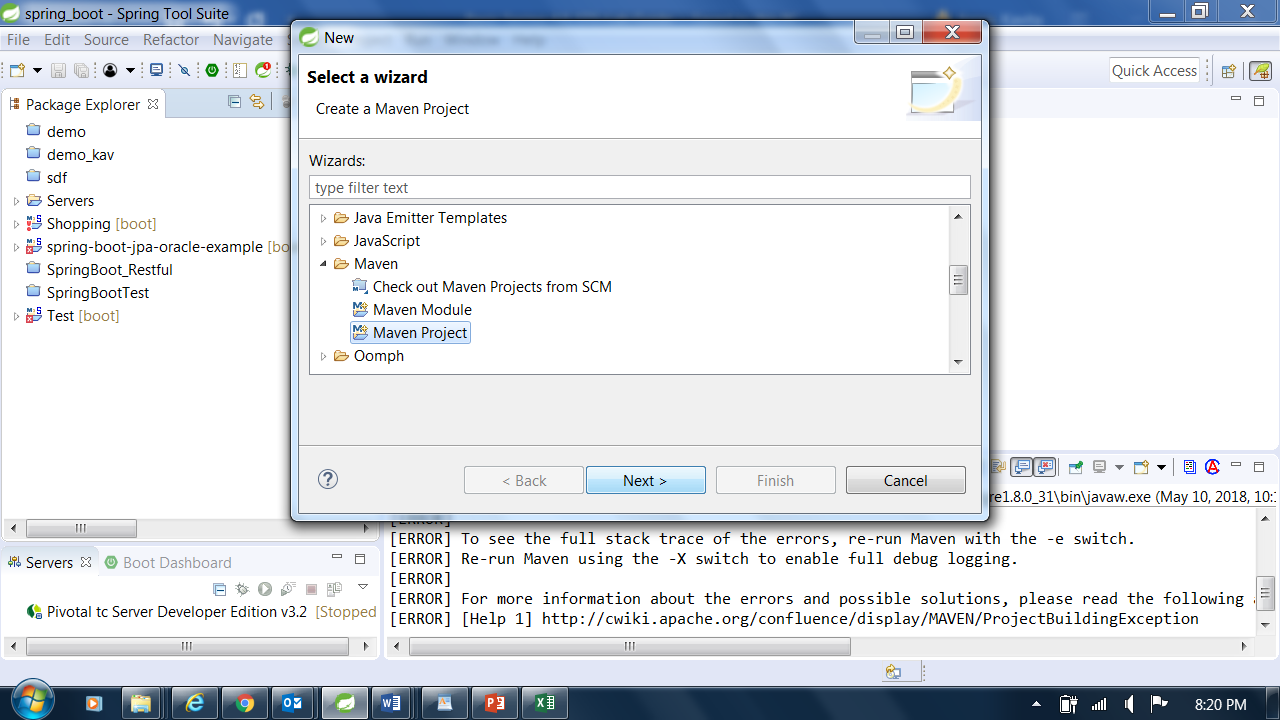
Select the namespaces require in the xml file And Click On Next - Finish.



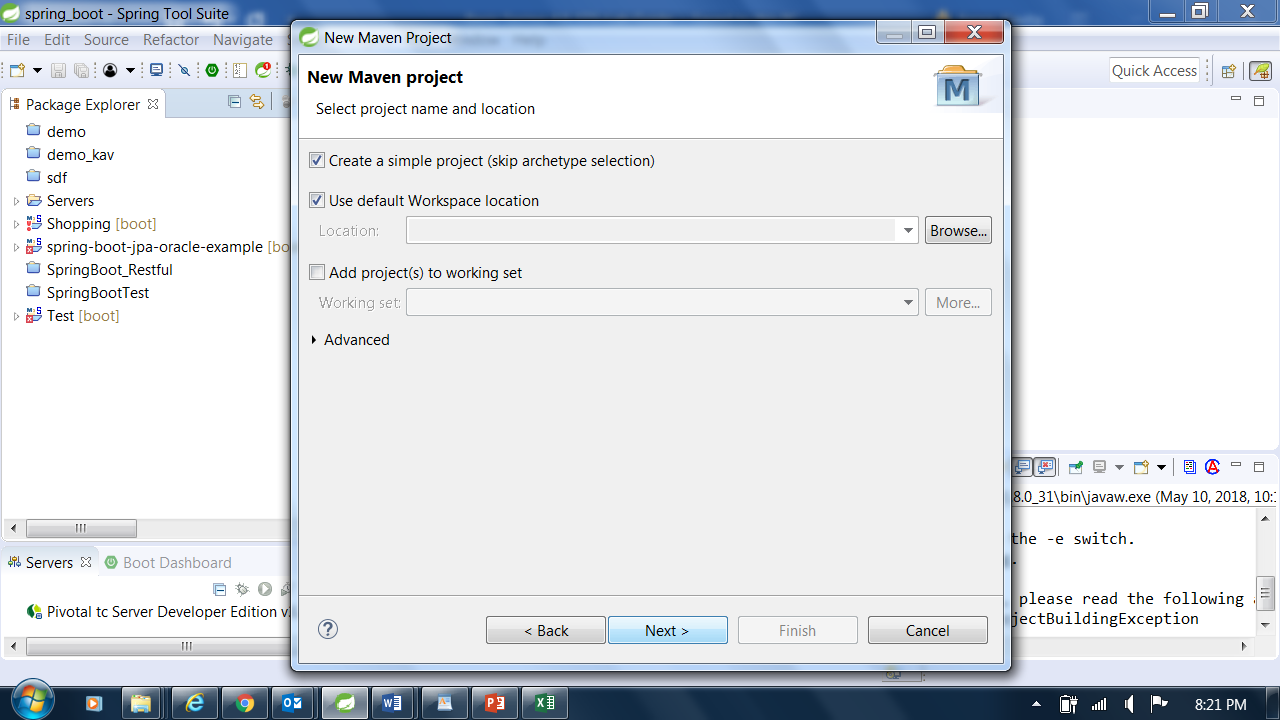
3) Steps To Create Spring Boot Application in STS using a Maven project

NOTE:-Need Internet connection to Create this project

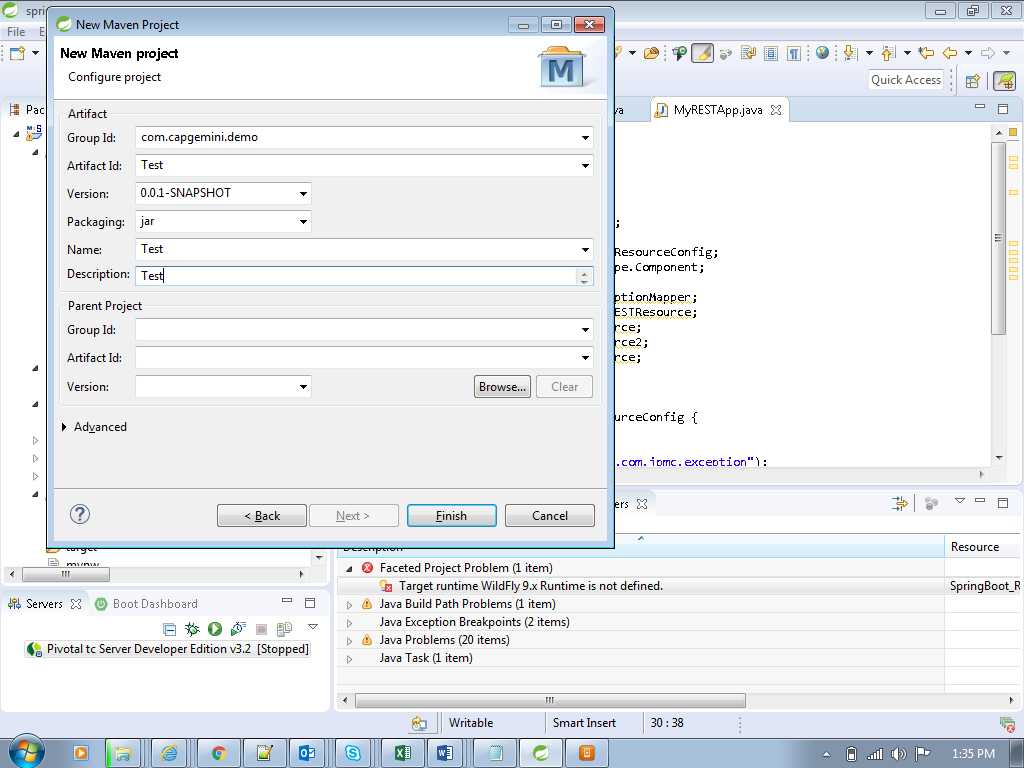
Step:-1) - Go To File menu 🡪New –Other - Maven -Maven Project- Click on Next



.



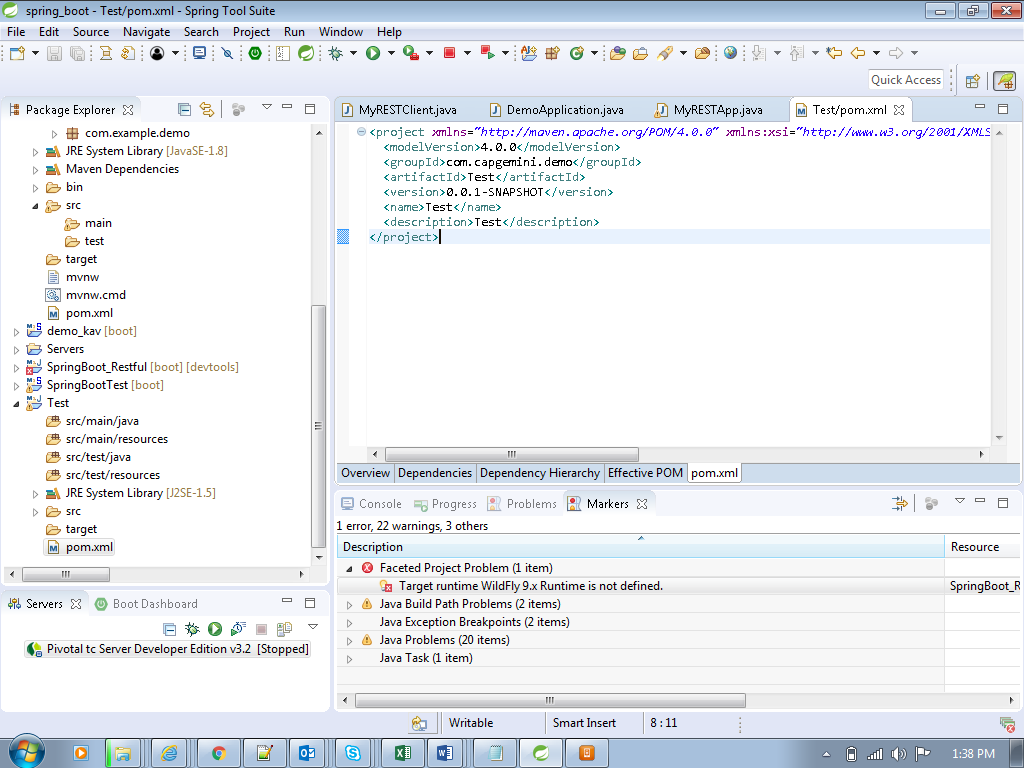
Step 2)Select the checkbox, Create a simple project and Click On Next-



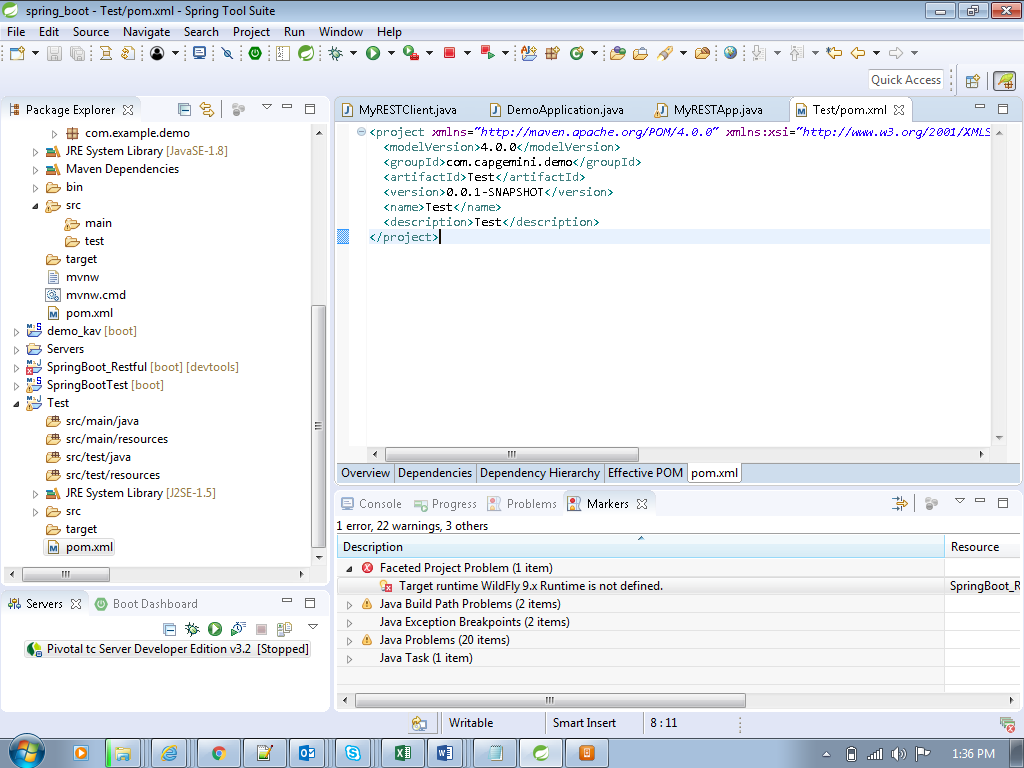
Step 3)Specify the group id, artifact Id, name and description

Click On Finish

Observe the package explorer for the folder structure of the newly created project



Double click on pom.xml and see its default content as below



Add the following code in the pom.xml .so that your pom.xml matches the one below

<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/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.capgemini.demo</groupId>

<artifactId>Test</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Test</name>

<description>Test</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.0.1.RELEASE</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<dependencies>

<dependency>

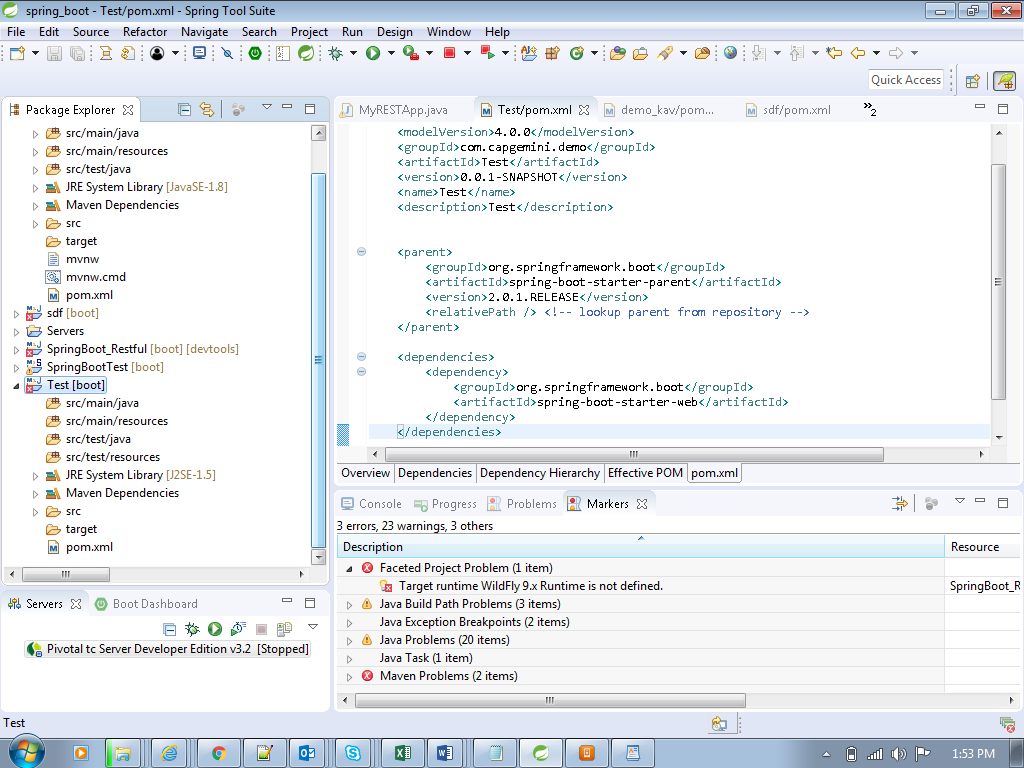
<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

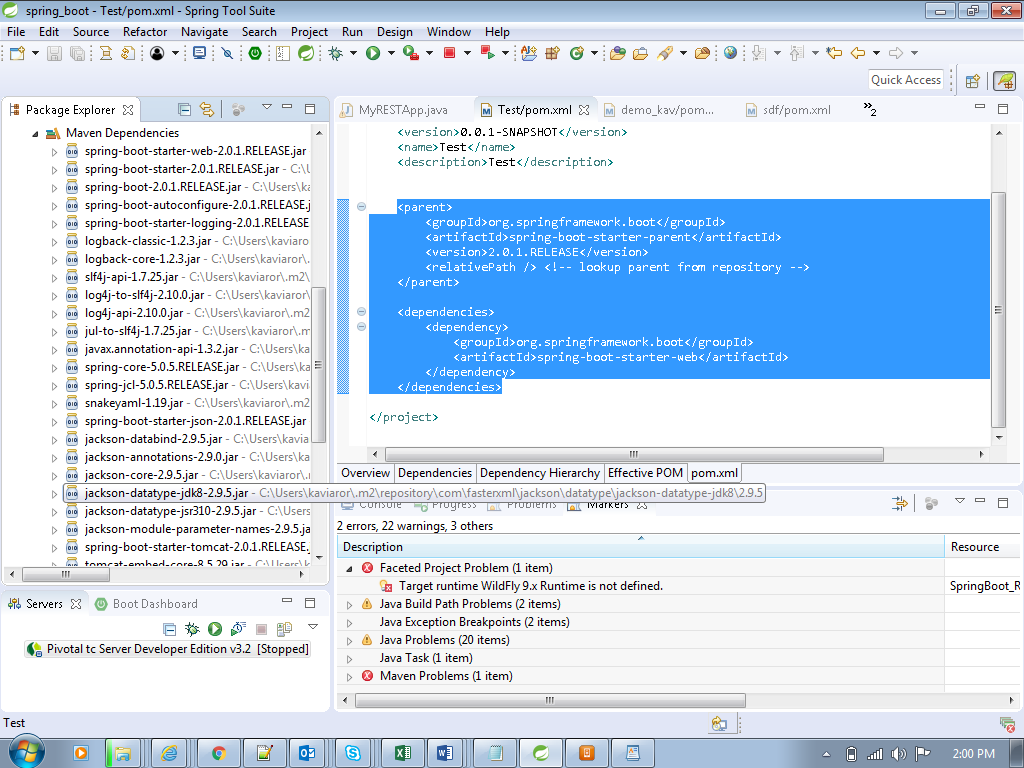
</dependencies>

</project>



After making changes to pom.xml, observe project folder structure in package explorer

Specifically observe that “Maven Dependencies” has been included into the project



These jar files are among those that you have copied physically into the project when you worked without Maven.

This is the beauty of what Spring boot and Maven does for you.

To remove the errors shown on the project in package explorer, do the following.

Right click on the Test project in package explorer 🡪 Maven 🡪 Update Project

Right click the project, create new java class having the following code in main method

@SpringBootApplication

**public** **class** Client {

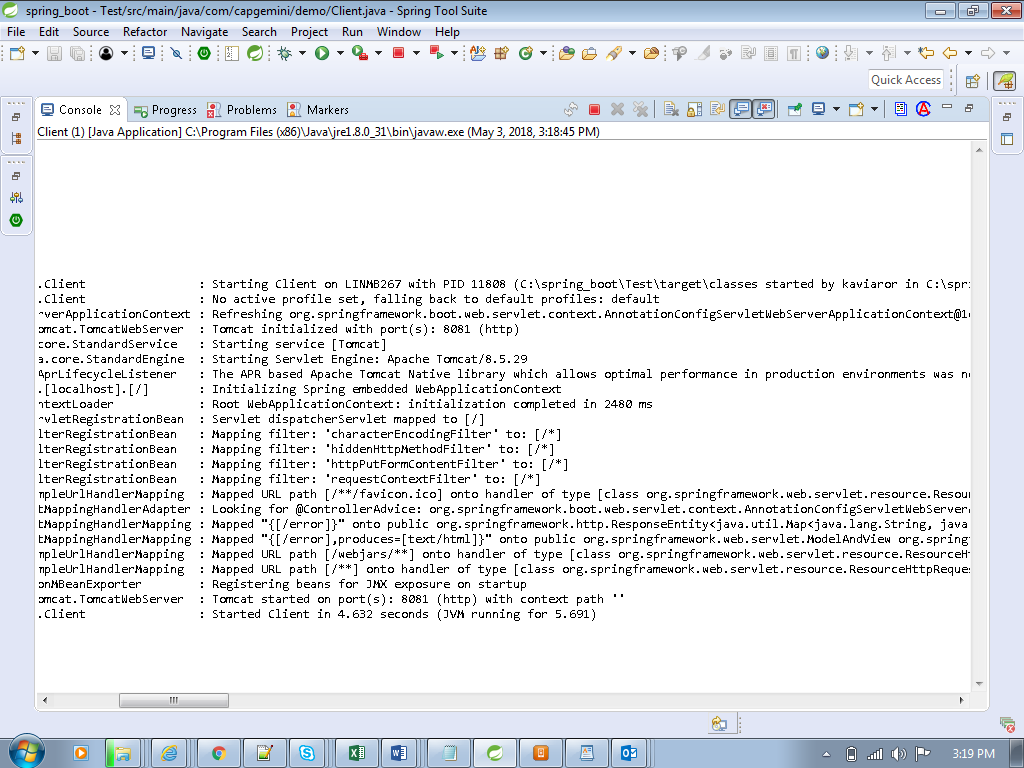
**public** **static** **void** main(String[] args) {

SpringApplication.*run*(Client.**class**,args);

}

}

Run the application as a java application and observe the console as shown below



You can keep a file called application.properties at the project level and write the following to change the default port number of 8080 which Tomcat server uses to start up

Server.port=8081

Navigate to browser and type <http://localhost:8081>

You will observe that nothing happens

Hence code the following in a new controller class

Create a class which acts as a controller

@RestController

**public** **class** HelloController {

@RequestMapping("/hello")

**public** String sayHi() {

**return** "Hi";

}

}

With the above controller class added, run the application again as a java application .

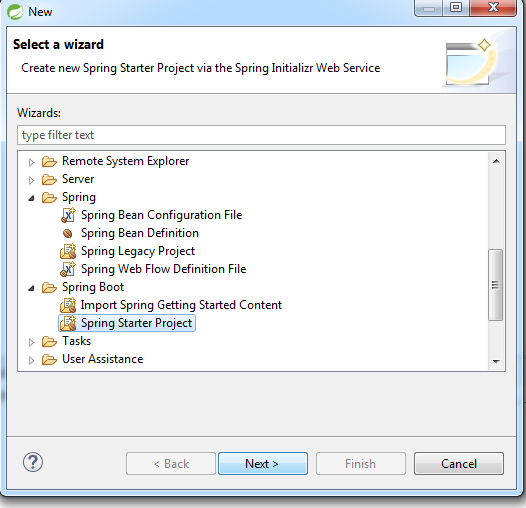
Navigate to browser and type <http://localhost:8081>

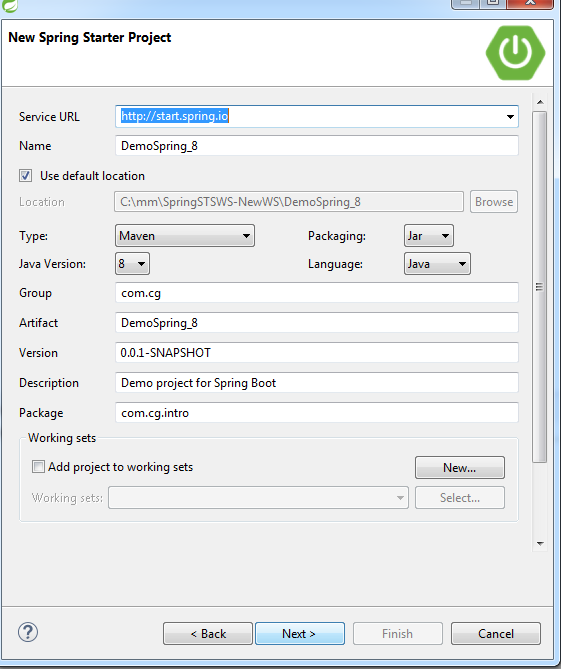
And observe the “Hi” message displayed on the browser page

4) Steps To Create Spring Boot project

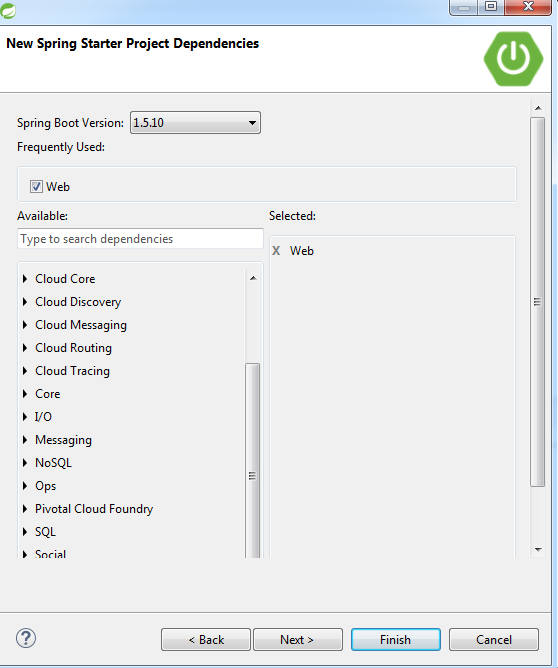
NOTE:-Need Internet connection to Create this project

Step:-1) - Go To New –Others - Spring Boot - Spring Starter Project- Click on Next

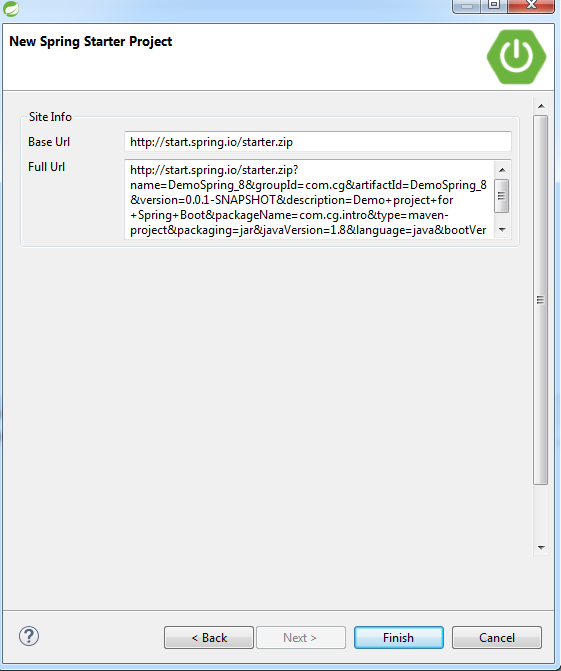




Again Click On Next- Select Web option - Click On Next



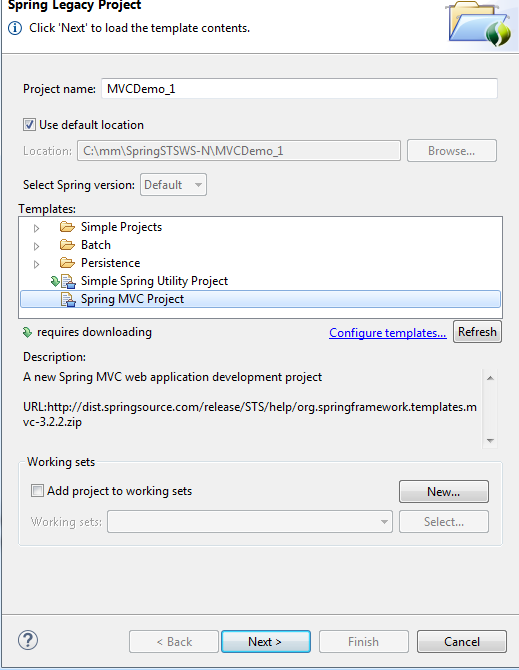
Click On Finish



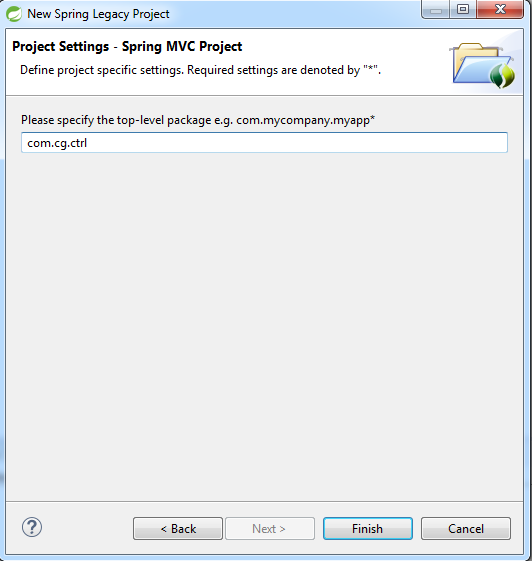
5) Steps To Create Spring MVC project

Step-1)For Creating Spring MVC project Go to New- select “Spring Legacy Project option”

Give a project name and click on nEXT button



Give a package name as follows and click on finish button.



Project structure for MVC with web.xml and pom.xml is created.