

## creating a project using spring approach(maven way)

---

Following are the important folders and files of the spring project

- 1) Src/Main/Java -> Entire java code for my business logic , we need to write here.
- 2) src/main/resources -> Any configuration code like prop file , database connection
- 3) src/test/java -> Unit test cases like Junit etc..
- 4) src/test/resources -> Configuration code for Unit test cases
- 5) Maven Classpath ->  
Entries mentioned in pom.xml file -> it downloads all jar file for the frameworks and keep the jars in class path.
- 6) **pom.xml ==> We will write configuration to download the framework jars**

to download any specific jar	we will mention them in pom.xml file
	it is the heart of the project

to create a springboot application, we need springboot jars

- we need to download springboot jar files,
- in pom.xml we need to write following section

```

https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>com.nareshit.helloworld</groupId>
4   <artifactId>HelloWorldInSpring</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6
7   <parent>
8     <groupId>org.springframework.boot</groupId>
9     <artifactId>spring-boot-starter-parent</artifactId>
10    <version>2.1.6.RELEASE</version>
11  </parent>
12
13  <dependencies>
14    <dependency>
15      <groupId>org.springframework.boot</groupId>
16      <artifactId>spring-boot-starter-web</artifactId>
17    </dependency>
18  </dependencies>
19 </project>

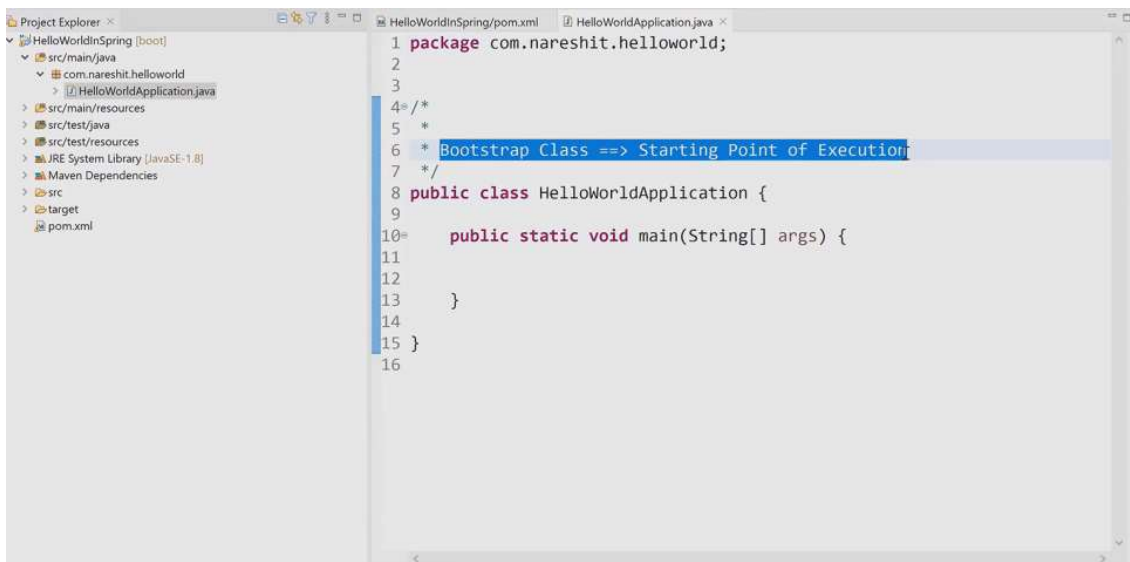
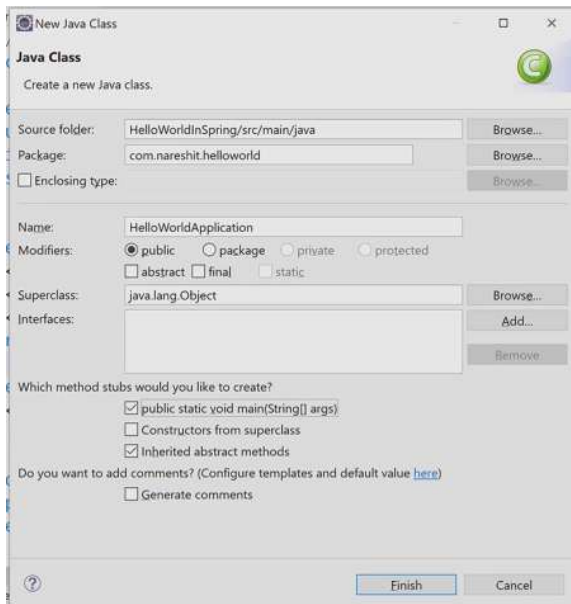
```

we have written the pom.xml to download web-jars

- when we use parent tag and mentioned a version in it --> the same version is applied for child dependencies and respective jars will be downloaded
- in the above parent version is 2.1.6 --> so for child dependencies also same version will be downloaded.

- 2) Need to create a class under a particular package and need to write the following code
  - a) `@SpringBootApplication` -> Annotation -> Bootstrap Class for spring boot project -> Starting Point of Spring boot application-> It will load all the packages into IOC Container i.e. ApplicationContext.
- 3) `@RestController` -> This the entry point for all external communications. Spring boot can be considered as middle ware service can be integrated with lot of front ends like `Angular/React/NodeJS/Spring MVC/Go/Vue`. This is similar `@Controller` annotation in Spring MVC.

## Bootstrap class



Bootstrap class	starting point of execution
-----------------	-----------------------------

How to know a class is a bootstrap class or not ?

- if we write a annotation '@SpringBootApplication'
- A Java class which is having '@SpringBootApplication' is called as Bootstrap class
- Every Springboot project should have only one bootstrap class.
- Bootstarp --> boot --> booting --> starting --> starting point

what is a annotation?

- A predefined keyword with "@" symbol is called as annotation.

Functionality of Bootstarp class

- 1. It will create IoC Container --> Inversion of Control Container.
- 2. It will perform memory management for both the system classes, developer class.
  - management --> creating the object, maintaing the state of the object, destroying the object.
- 3. It will create a web application context