Spring boot First Example with configurations.

THE KIRAN ACADEMY

We need to follow some steps to configure spring boot project.

Pre-Requisite:

- Jdk 17
- Latest eclipse
- MySQL – required in future.

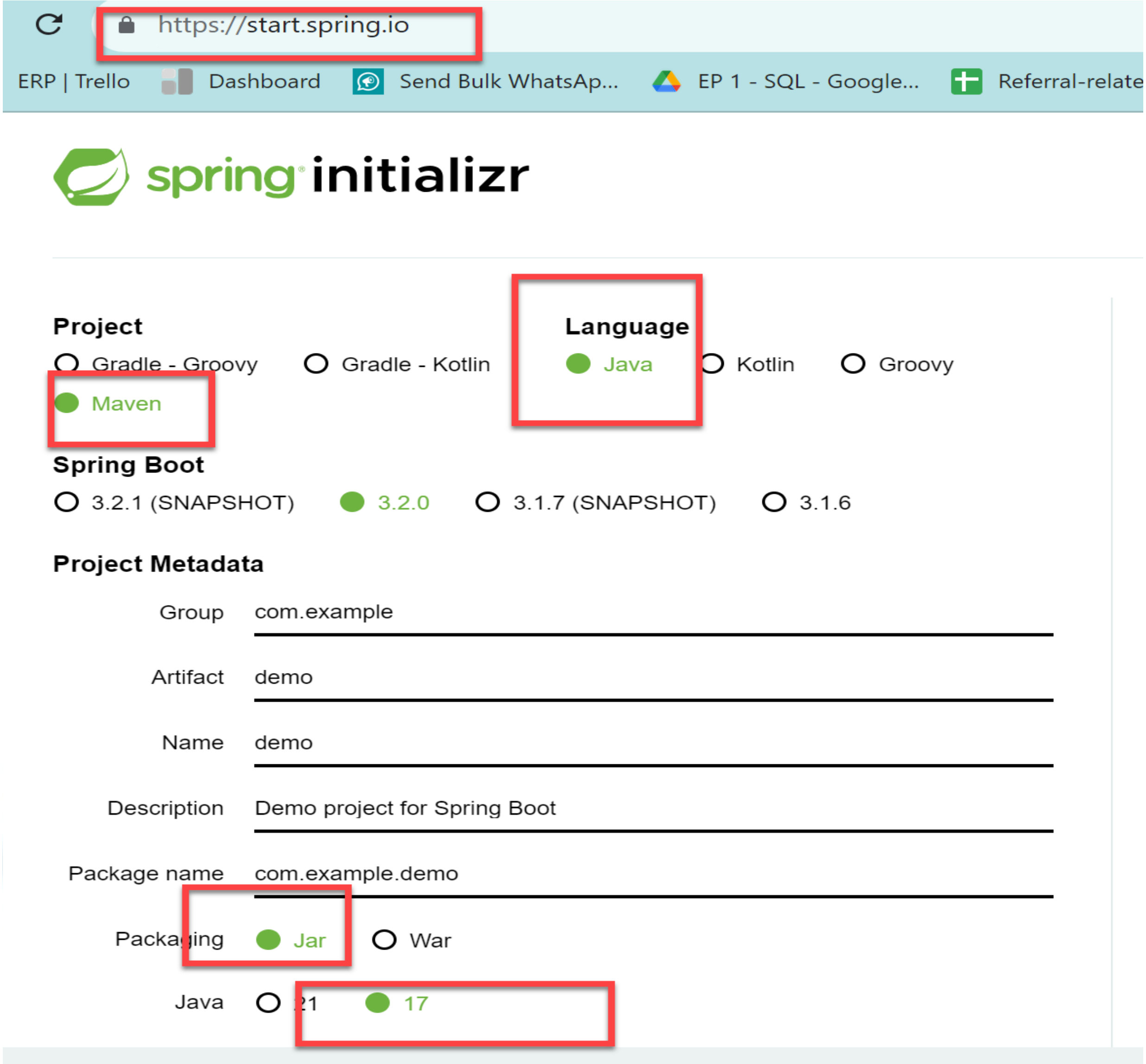
To install java and eclipse, follow this document, if you already have latest eclipse and java 8 skip this step.

https://javabykiran.in/core-java/JBKSETUP001-java-eclipse-setup.pdf

Step #1

Open website https://start.spring.io/

Select below details.



Project
○ Gradle - Groovy ○ Gradle - Kotlin
● Maven

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 3.2.1 (SNAPSHOT) ● 3.2.0 ○ 3.1.7 (SNAPSHOT) ○ 3.1.6

Project Metadata

Group com.example

Artifact demo

Name demo

Description Demo project for Spring Boot

Package name com.example.demo

Packaging ● Jar ○ War

Java ○ 21 ● 17

ACADEMY

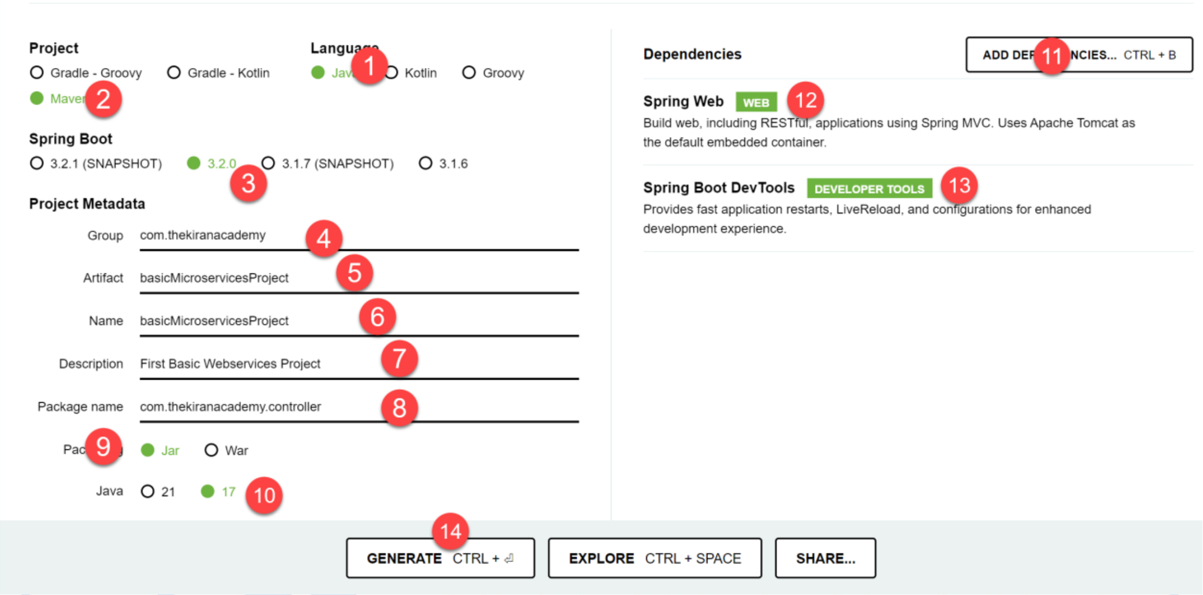
Project Metadata

- Group
  - This is a folder where all other spring boot project you want to store.
  - enter **com.javabykiran**
- Artifact
  - Your current project name.
  - enter **basicMicroserviceProject**
- Name
  - Any name
- Description
  - First basic microservice or REST webservice project
- Package name

  - com.javabykiran.controller
- Packaging
  - war: This is to create web project.
  - jar: This is to create API project.
  - select **jar.**
- Java
  - version 17 we will be using.
  - **version 17**

Click on dependencies.

Add web , devtools



Project
○ Gradle - Groovy ○ Gradle - Kotlin ● Maven

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 3.2.1 (SNAPSHOT) ● 3.2.0 ○ 3.1.7 (SNAPSHOT) ○ 3.1.6

Project Metadata

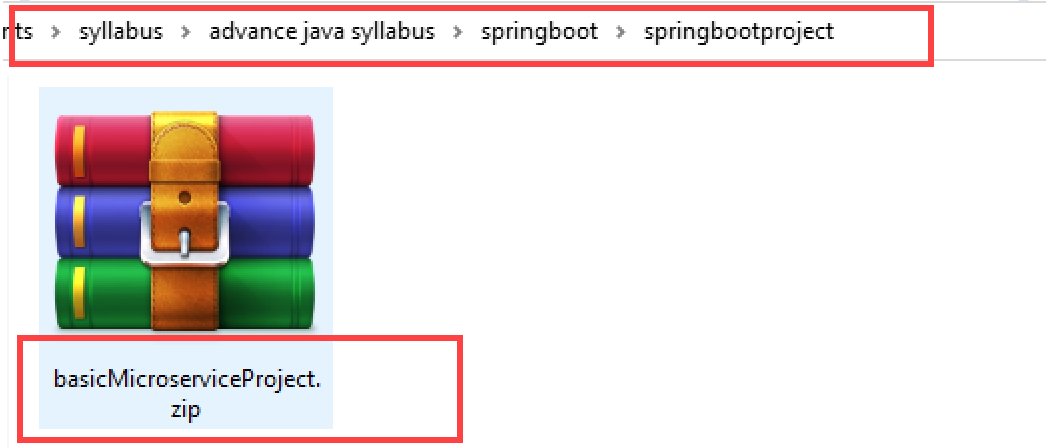
Group	com.thekiranacademy
Artifact	basicMicroservicesProject
Name	basicMicroservicesProject
Description	First Basic Webservices Project
Package name	com.thekiranacademy.controller
Packaging	● Jar ○ War
Java	○ 21 ● 17

Dependencies

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

ACADEMY

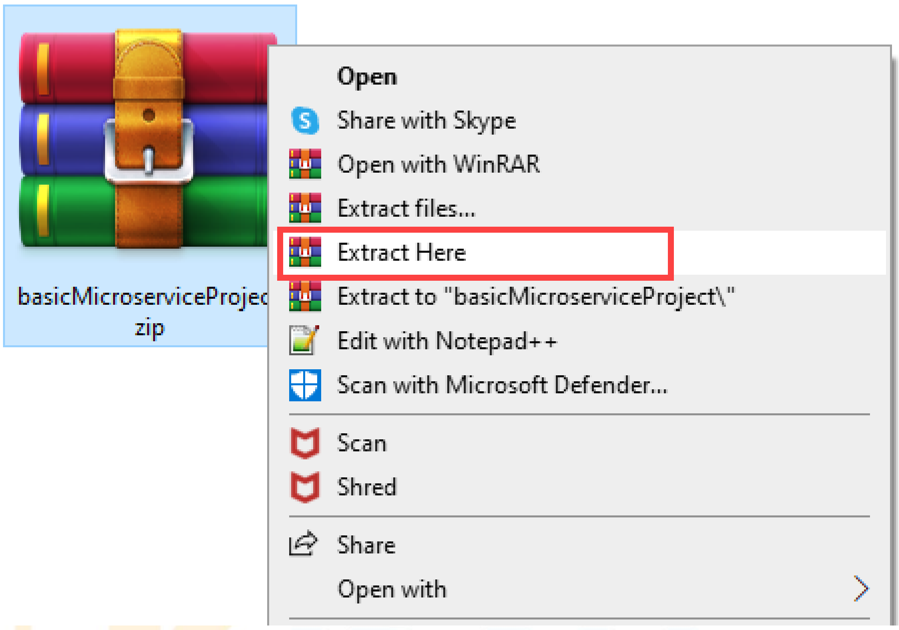


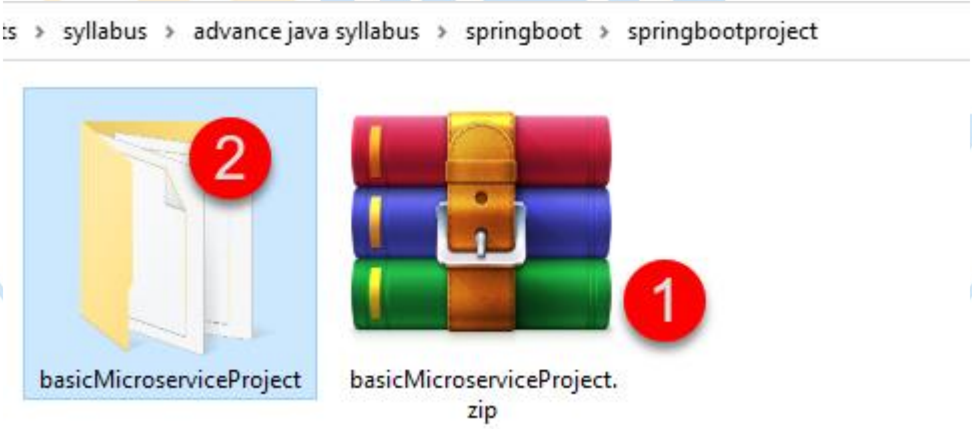
rts › syllabus › advance java syllabus › springboot › springbootproject

basicMicroserviceProject.zip

Step #2

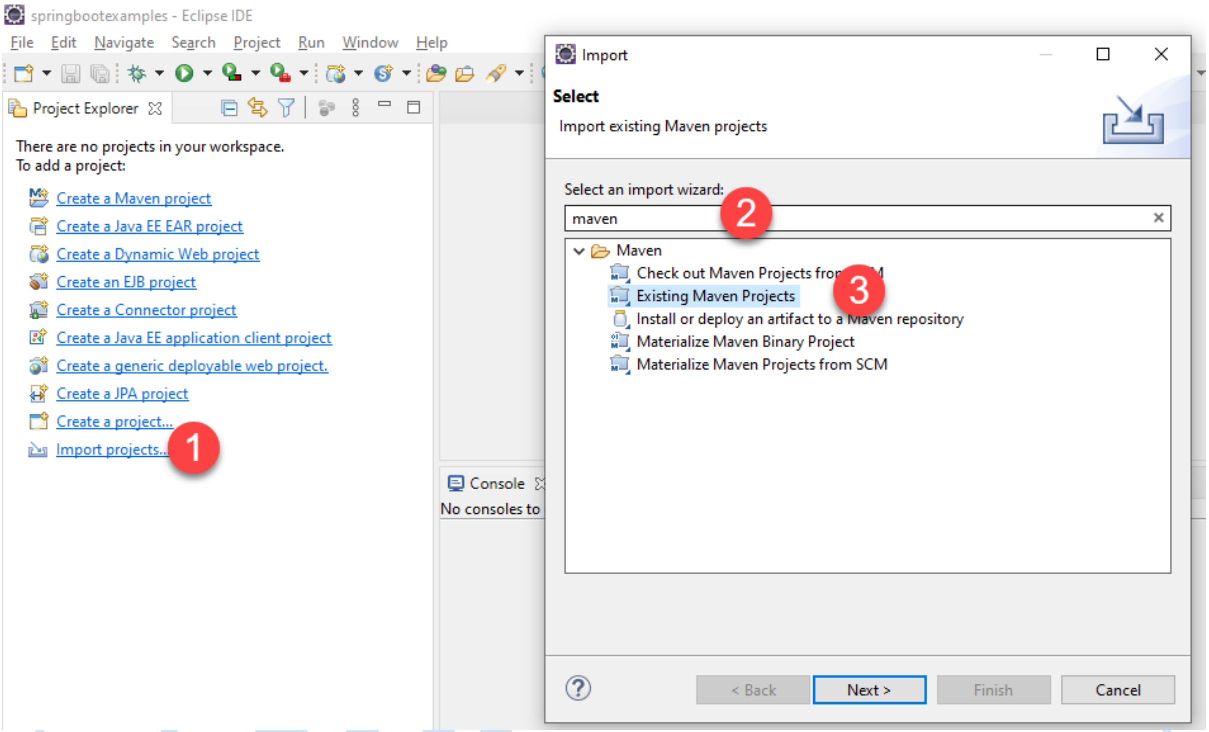
Extract project




ts › syllabus › advance java syllabus › springboot › springbootproject
2
basicMicroserviceProject
1
basicMicroserviceProject.
zip

Step #3

Import project to eclipse.



springbootexamples - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
Console
Import
Select
Back
Next >
Finish
Cancel
1
2
3

There are no projects in your workspace.
To add a project:

- Create a Maven project
- Create a Java EE EAR project
- Create a Dynamic Web project
- Create an EJB project
- Create a Connector project
- Create a Java EE application client project
- Create a generic deployable web project.
- Create a JPA project
- Create a project...
- Import projects...

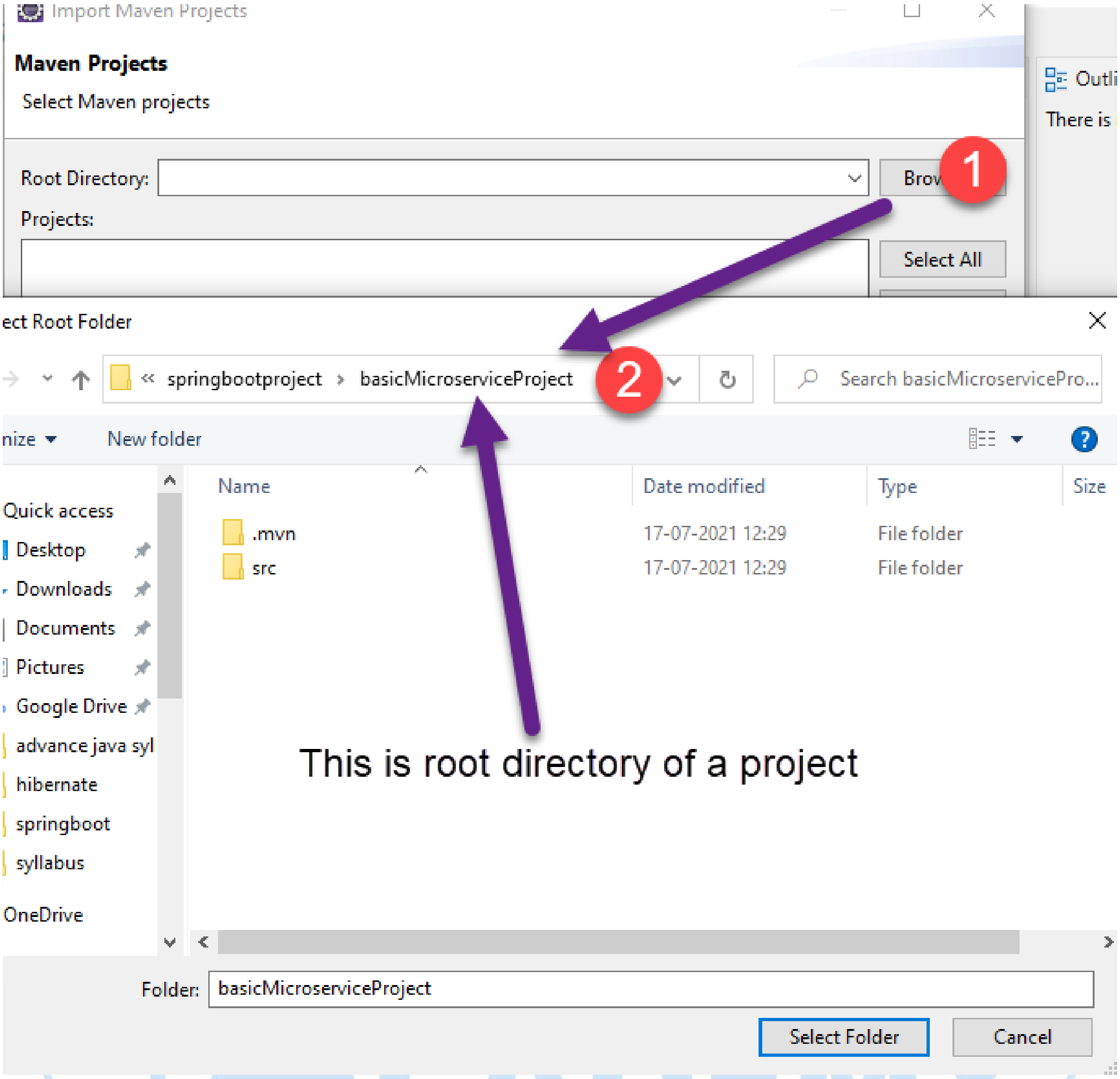
No consoles to

Import existing Maven projects

Select an import wizard: maven

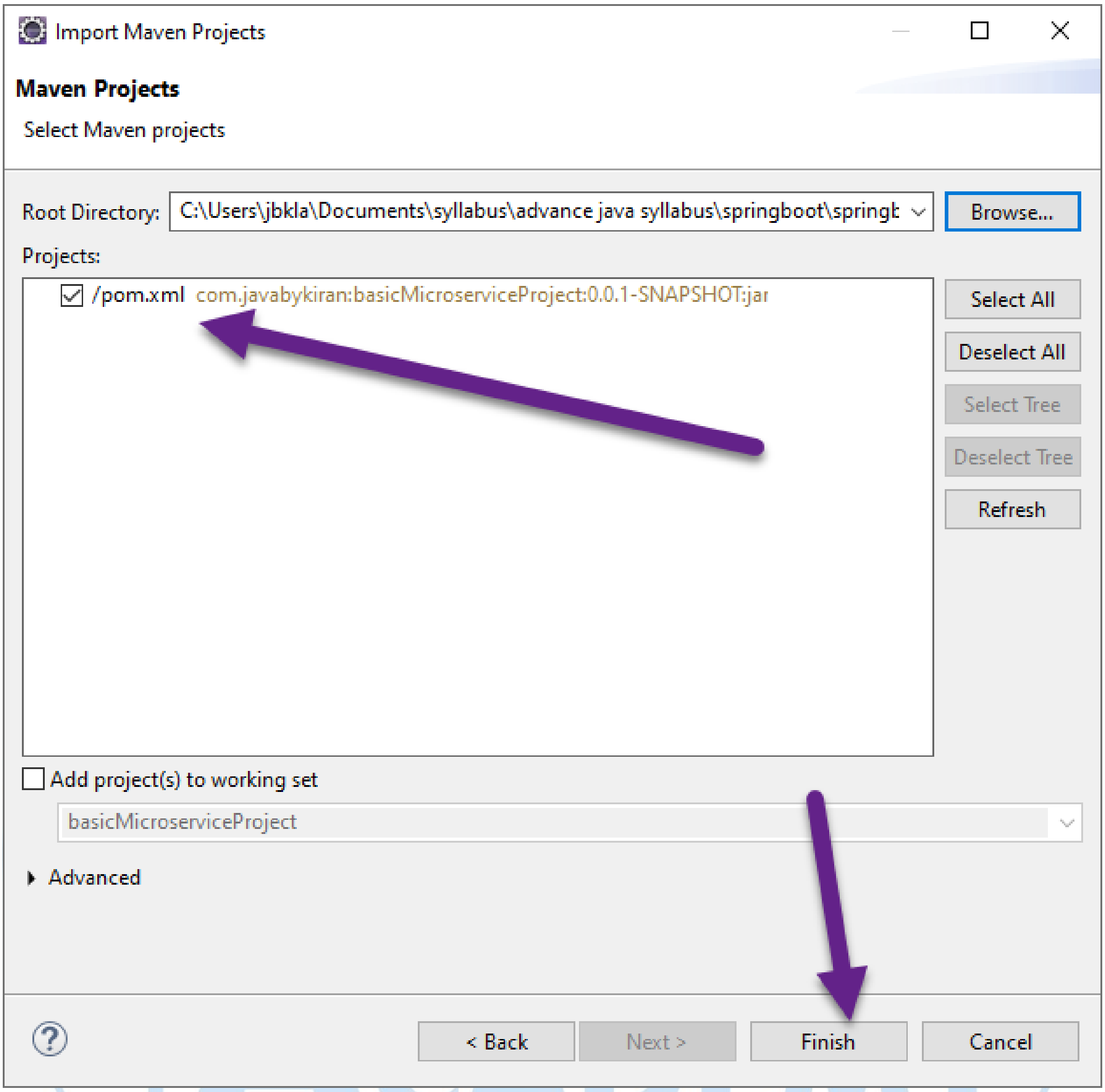
- Maven
  - Check out Maven Projects from SCM
  - Existing Maven Projects
  - Install or deploy an artifact to a Maven repository
  - Materialize Maven Binary Project
  - Materialize Maven Projects from SCM

THE KIRAN ACADEMY



This is root directory of a project

Once you click on Select Folder you can see pom.xml will be imported.



Import Maven Projects

Maven Projects

Select Maven projects

Root Directory: C:\Users\jbkla\Documents\syllabus\advance java syllabus\springboot\springb

Projects:

☑ /pom.xml com.javabykiran:basicMicroserviceProject:0.0.1-SNAPSHOT:jar

□ Add project(s) to working set

basicMicroserviceProject

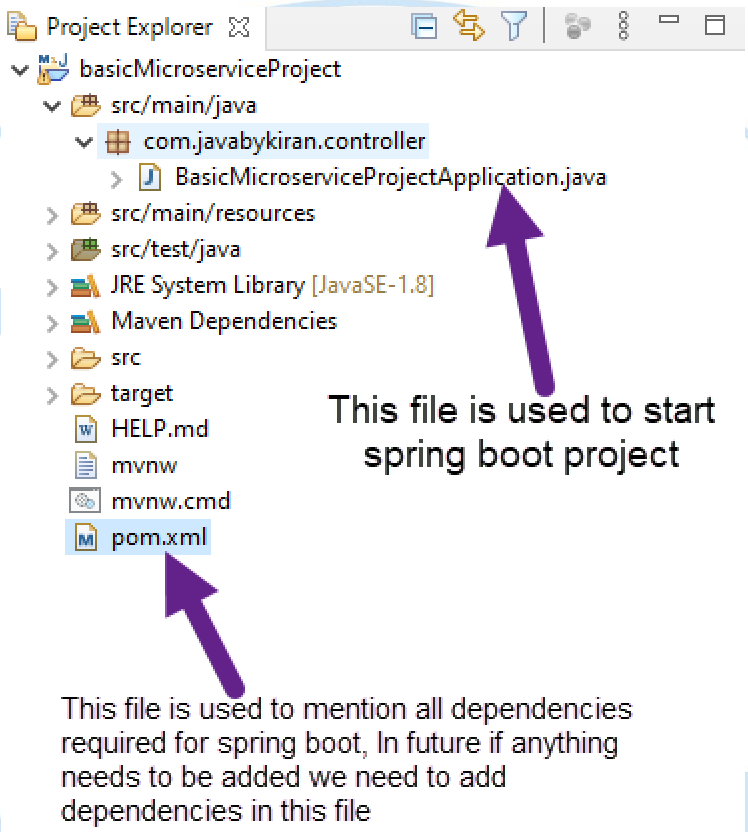
Once click on finish. Wait for dependencies to get downloaded in our project. It may take few minutes depending upon your internet speed.

Below is project structure after downloading all dependencies. Make sure you do not have any errors in problem windows. Most of the errors occur if we do not have proper version of java or maven or eclipse.

Open problem window – window – show view - problems.

Step #4

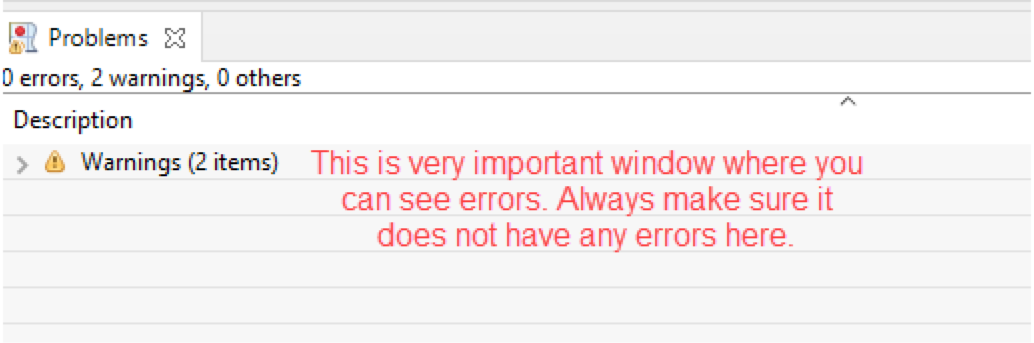
Check if out setup is correct.



This file is used to start spring boot project

This file is used to mention all dependencies required for spring boot, In future if anything needs to be added we need to add dependencies in this file

This is important window as below. For everything you do in eclipse always keep this window open for seeing any errors.

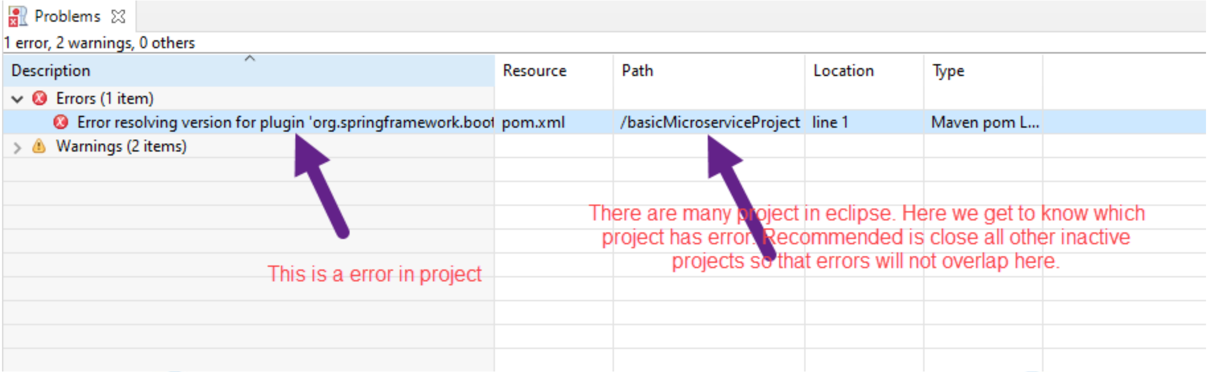


0 errors, 2 warnings, 0 others

This is very important window where you can see errors. Always make sure it does not have any errors here.

KIRAN ACADEMY

Just for time being we will see how errors will look like.



Problems

1 error, 2 warnings, 0 others

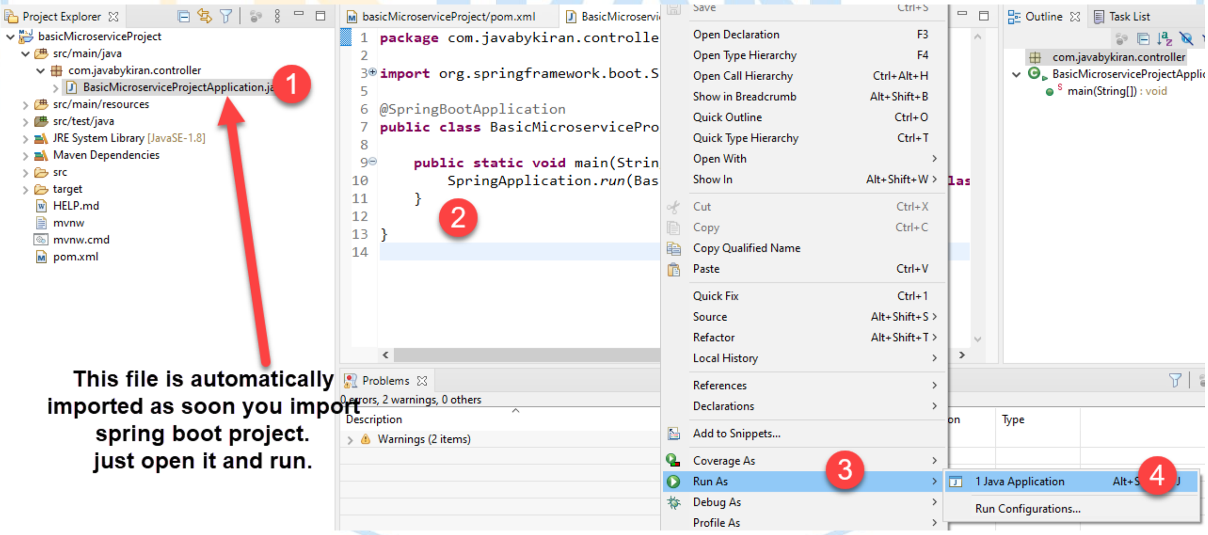
Description	Resource	Path	Location	Type
Errors (1 item)				
Error resolving version for plugin 'org.springframework.boot	pom.xml	/basicMicroserviceProject	line 1	Maven pom L...
Warnings (2 items)				

This is a error in project

There are many project in eclipse. Here we get to know which project has error. Recommended is close all other inactive projects so that errors will not overlap here.

Step #5

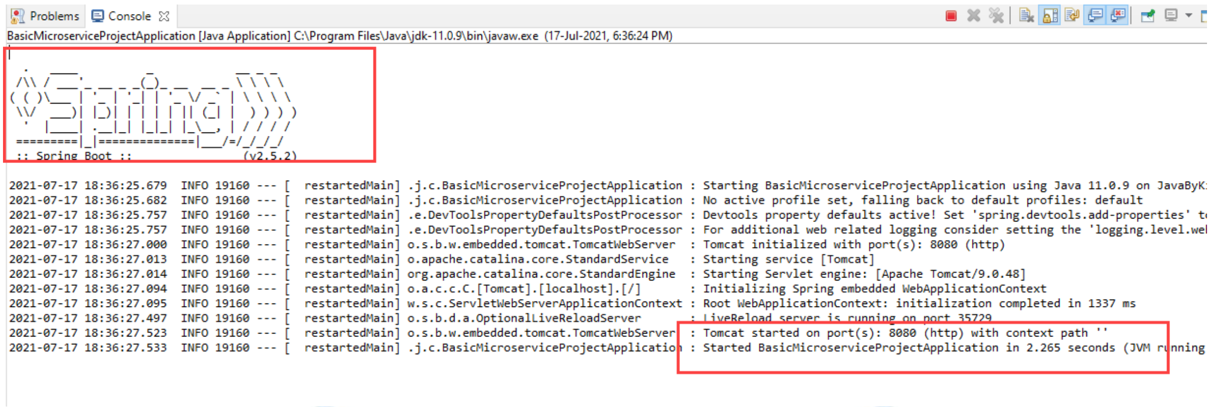
It's time to check if our spring boot project is running correctly.



```
package com.javabykiran.controlle

import org.springframework.boot.S

@SpringBootApplication
public class BasicMicroservicePro

    public static void main(Strin
        SpringApplication.run(Bas
    }

}
```

This file is automatically imported as soon you import spring boot project. just open it and run.

After running this console should have below output.



```
BasicMicroserviceProjectApplication [Java Application] C:\Program Files\Java\jdk-11.0.9\bin\javaw.exe (17-Jul-2021, 6:36:24 PM)

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.5.2)

2021-07-17 18:36:25.679  INFO 19160 --- [  restartedMain] .j.c.BasicMicroserviceProjectApplication : Starting BasicMicroserviceProjectApplication using Java 11.0.9 on JavaByK
2021-07-17 18:36:25.682  INFO 19160 --- [  restartedMain] .j.c.BasicMicroserviceProjectApplication : No active profile set, falling back to default profiles: default
2021-07-17 18:36:25.757  INFO 19160 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to
2021-07-17 18:36:25.757  INFO 19160 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.we
2021-07-17 18:36:27.000  INFO 19160 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2021-07-17 18:36:27.013  INFO 19160 --- [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2021-07-17 18:36:27.014  INFO 19160 --- [  restartedMain] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.48]
2021-07-17 18:36:27.094  INFO 19160 --- [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2021-07-17 18:36:27.095  INFO 19160 --- [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1337 ms
2021-07-17 18:36:27.497  INFO 19160 --- [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2021-07-17 18:36:27.523  INFO 19160 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2021-07-17 18:36:27.533  INFO 19160 --- [  restartedMain] .j.c.BasicMicroserviceProjectApplication : Started BasicMicroserviceProjectApplication in 2.265 seconds (JVM running
```

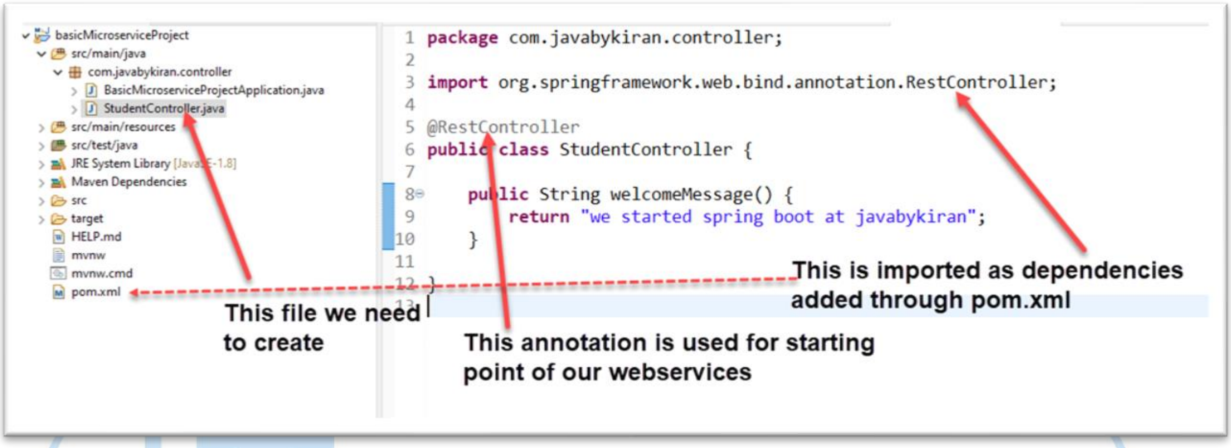
Now it is time to write simple first webservice.

Step #6

1. Create controller class.
   a. This is a class which will be hit by client first.
   b. This class takes a request and send response back to client.
      i. Response can be json or xml format.
      ii. Our focus will be on json format.
2. Use all annotations required for rest API.
   a. @Restcontroller
      i. This annotation is used at class level.
      ii. This is mandatory if we want to create rest webservice.
   b. @Requestmapping
      i. @getMapping
      ii. @postMapping
      iii. @putMapping
      iv. etc.

Step #7

Create CustomerController.java as below.



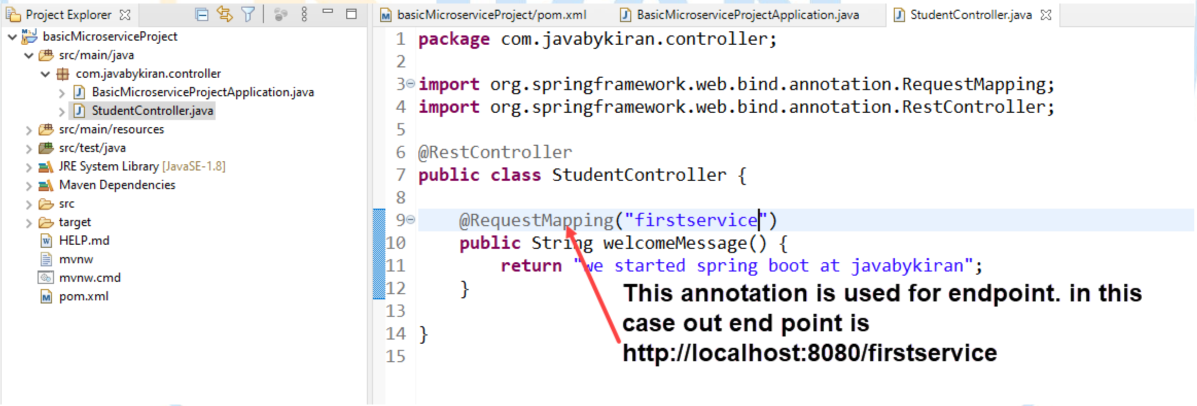
```java
package com.javabykiran.controller;

import org.springframework.web.bind.annotation.RestController;

@RestController
public class StudentController {

    public String welcomeMessage() {
        return "we started spring boot at javabykiran";
    }

}
```

This file we need to create

This annotation is used for starting point of our webservices

This is imported as dependencies added through pom.xml

We have not yet added endpoint for our service. So add at method level @RequestMapping annotation

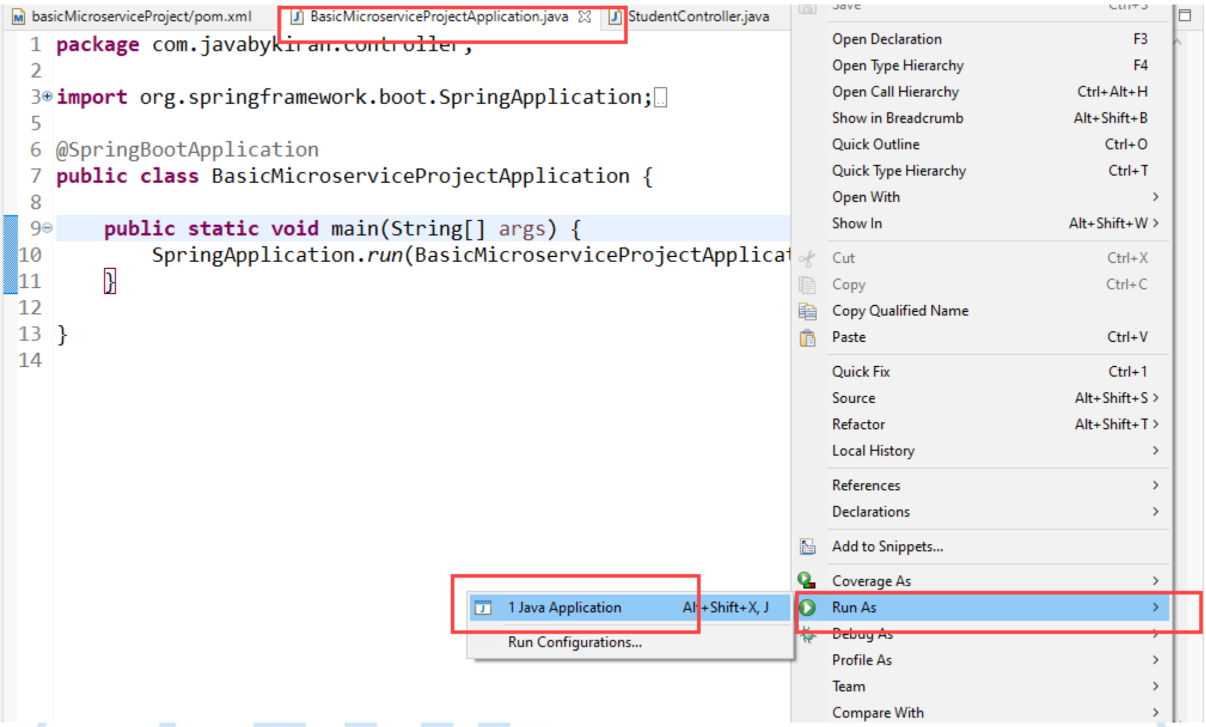


```java
package com.javabykiran.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class StudentController {

    @RequestMapping("firstservice")
    public String welcomeMessage() {
        return "we started spring boot at javabykiran";
    }

}
```

This annotation is used for endpoint. in this case out end point is http://localhost:8080/firstservice

Go to browser. This is simple service we have created we need to run this service.

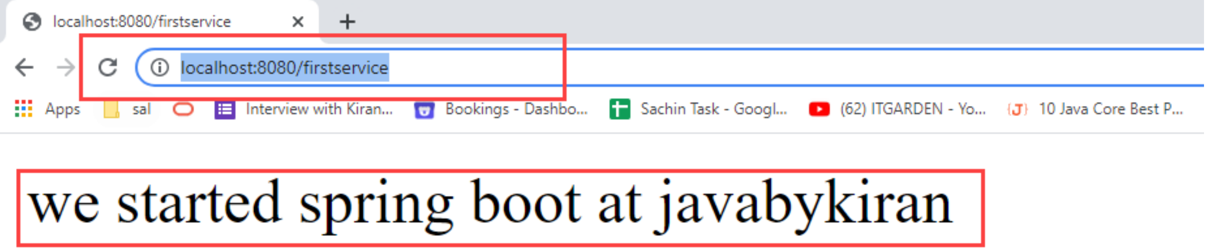
First start a project from main method.



```
package com.javabykiran.controller;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class BasicMicroserviceProjectApplication {

    public static void main(String[] args) {
        SpringApplication.run(BasicMicroserviceProjectApplicat
    }

}
```

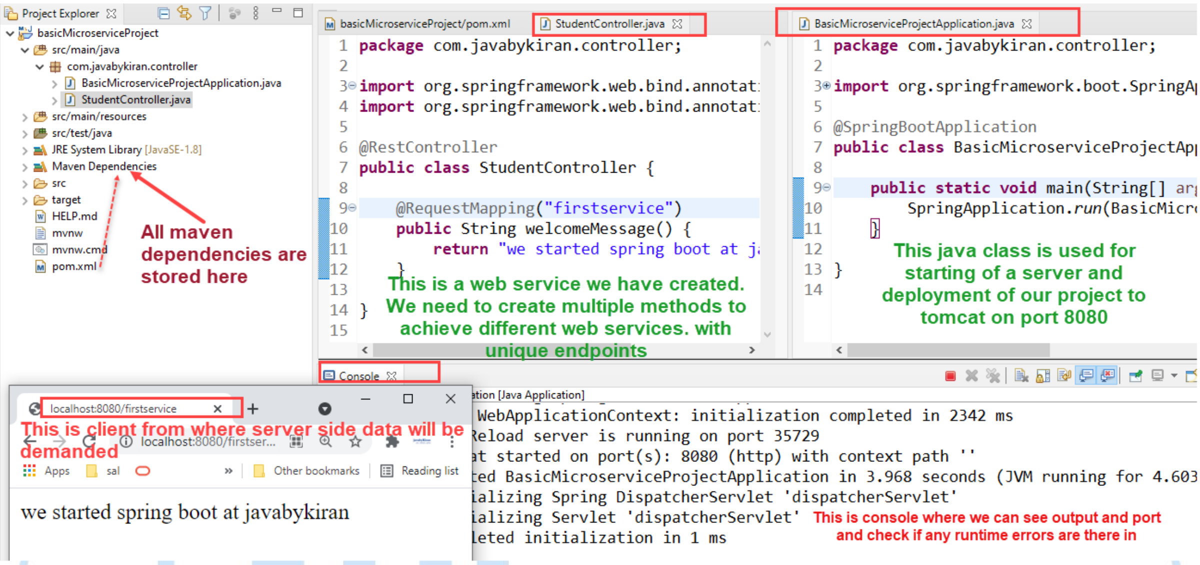
Check console if you see any exceptions if not go to browser.

In address bar hit url http://localhost:8080/firstservice



we started spring boot at javabykiran

Summary


Project Explorer
basicMicroserviceProject
src/main/java
com.javabykiran.controller
BasicMicroserviceProjectApplication.java
StudentController.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
basicMicroserviceProject/pom.xml
StudentController.java
BasicMicroserviceProjectApplication.java
Console
localhost:8080/firstservice
localhost:8080/firstser...
Apps
sal
Other bookmarks
Reading list

All maven dependencies are stored here

```
package com.javabykiran.controller;

import org.springframework.web.bind.annotat
import org.springframework.web.bind.annotat

@RestController
public class StudentController {

    @RequestMapping("firstservice")
    public String welcomeMessage() {
        return "we started spring boot at ja
    }

}
```

This is a web service we have created. We need to create multiple methods to achieve different web services. with unique endpoints

```
package com.javabykiran.controller;

import org.springframework.boot.SpringAp

@SpringBootApplication
public class BasicMicroserviceProjectAp

    public static void main(String[] ar
        SpringApplication.run(BasicMicr
    }

}
```

This java class is used for starting of a server and deployment of our project to tomcat on port 8080

```
ation [Java Application]
 WebApplicationContext: initialization completed in 2342 ms
Reload server is running on port 35729
at started on port(s): 8080 (http) with context path ''
ted BasicMicroserviceProjectApplication in 3.968 seconds (JVM running for 4.603
ializing Spring DispatcherServlet 'dispatcherServlet'
ializing Servlet 'dispatcherServlet'
leted initialization in 1 ms
```

This is console where we can see output and port and check if any runtime errors are there in

This is client from where server side data will be demanded

we started spring boot at javabykiran

Homework:

Read tutorial from below link to know in depth of spring boot. Screens used might be different than this, but concept is same.

https://www.jbktutorials.com/spring-boot/introduction-to-spring-boot.php#gsc.tab=0