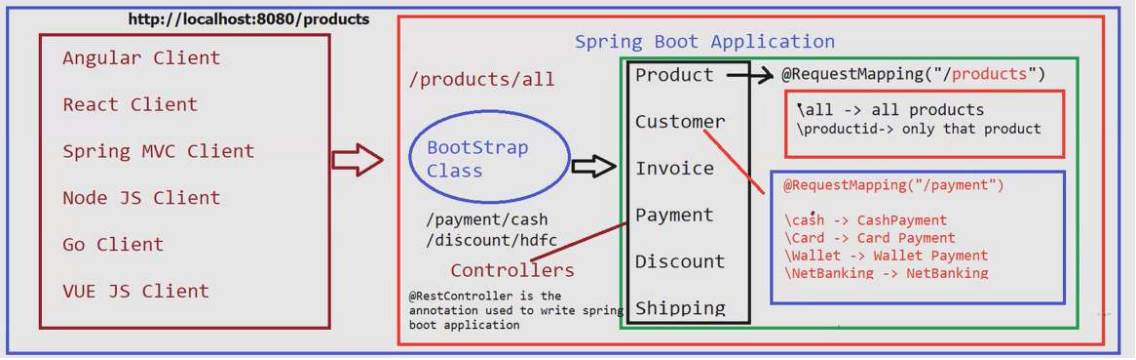
What is @RestController

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.



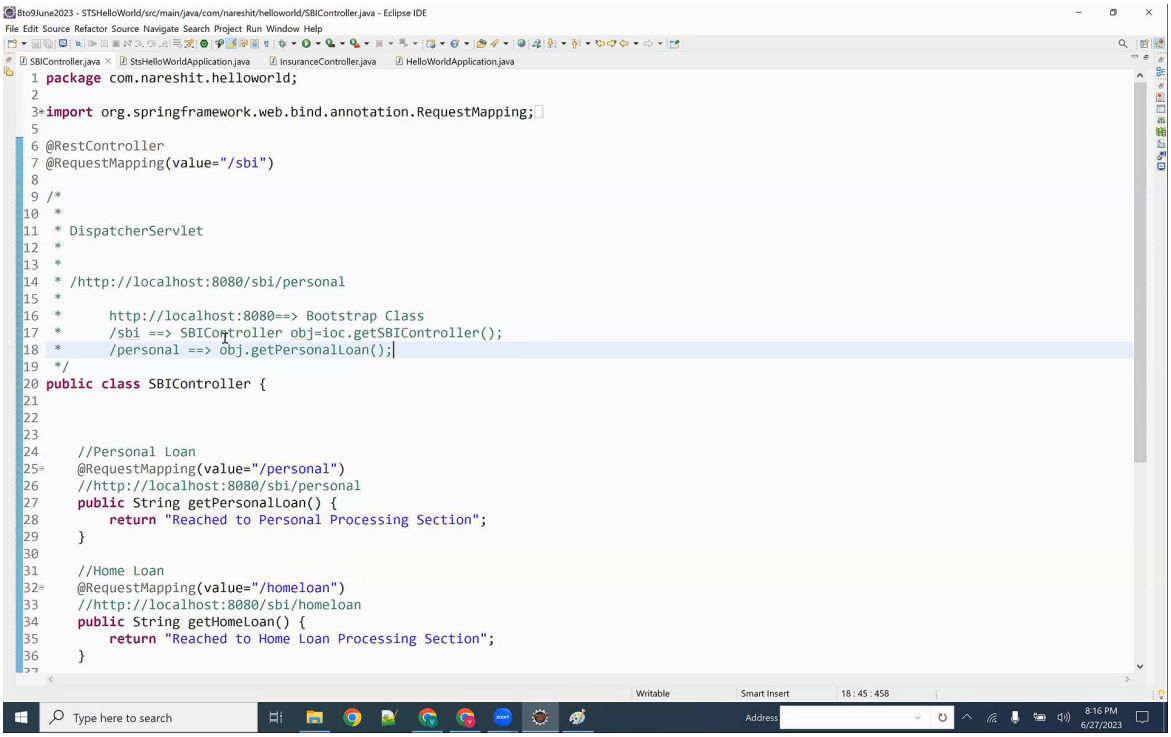

```
*
* Every Spring Boot Project should have only one bootstrap class
*
* Any predefinedkeyword with "@" Symbol is called as annotation
*
* Functionality of Bootstrap Class
*
* 1) It will create IOC Container ==> Inversion of Control Container
* 2) It will Manage all memory management for both System Classes , Developer Classes
*      Managing ==> Creating the object, maintaing the state of an object, and destroying the object
* 3) It will create a web application context==>
*       All the classes written by developer moved to IOC Container in temporary web application memory
* 4) It will create dispatcherservlet ==> It will register all rest controllers and all requestmappings in the memory
*        Any request comes ==> It will redirect to a specific method
*        Rules ==> Every URL Mapping should be unique for application
*               ==> @RequestMapping + @RestController Mapping ==> Unique through out application
*
*
*/
```

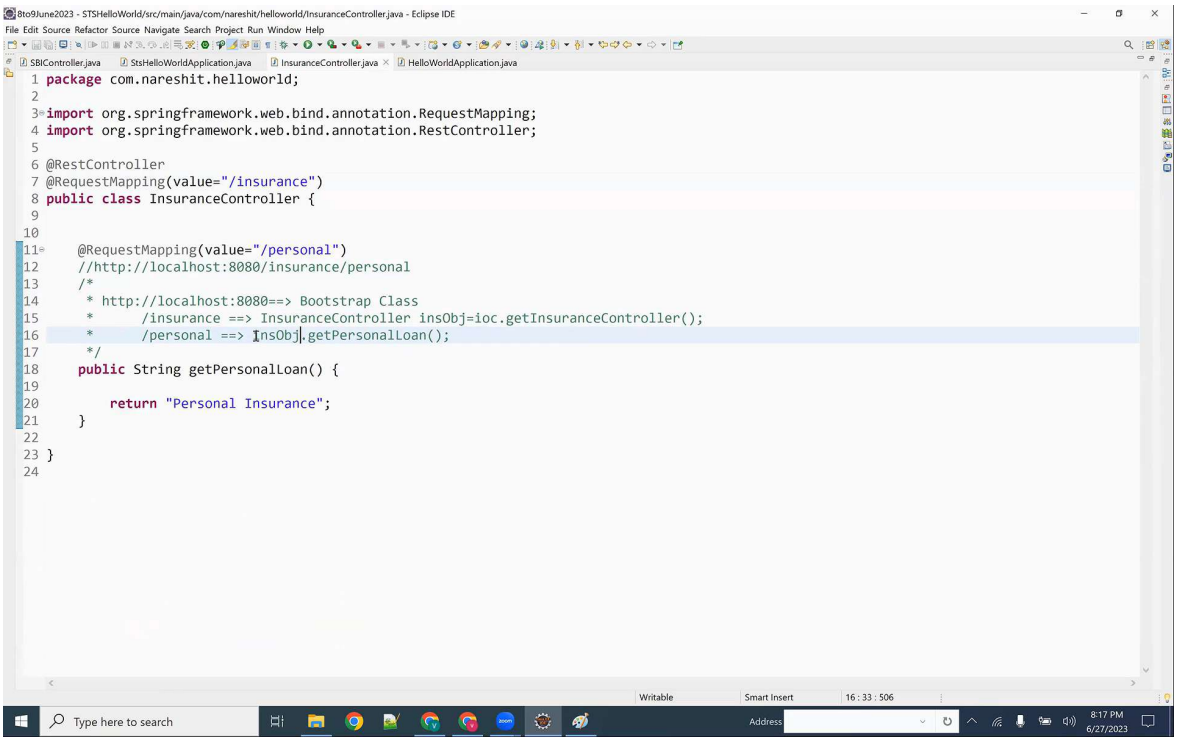
@RestController --> it is going to expose functionality to outside

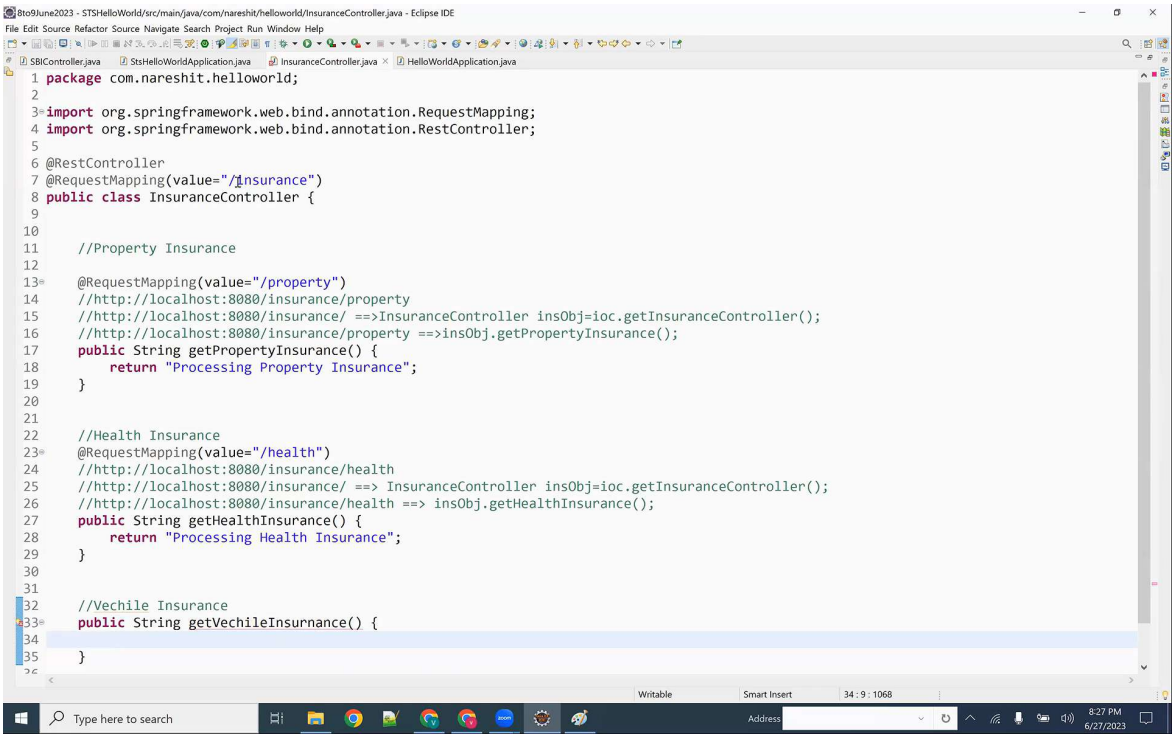
What is the meaning of Dispatch Servlet ?

- Boostrap class will create dispatcher servlet.
- while boot strap class is running, disptacher servlet will register all rest controllers and all requestmapping in the memory.
- rule: any request comes, dispatcher servlet redirect to a specific method
- @RestController + @RequestMapping (url pattern/mapping) --> should be unique through out the application

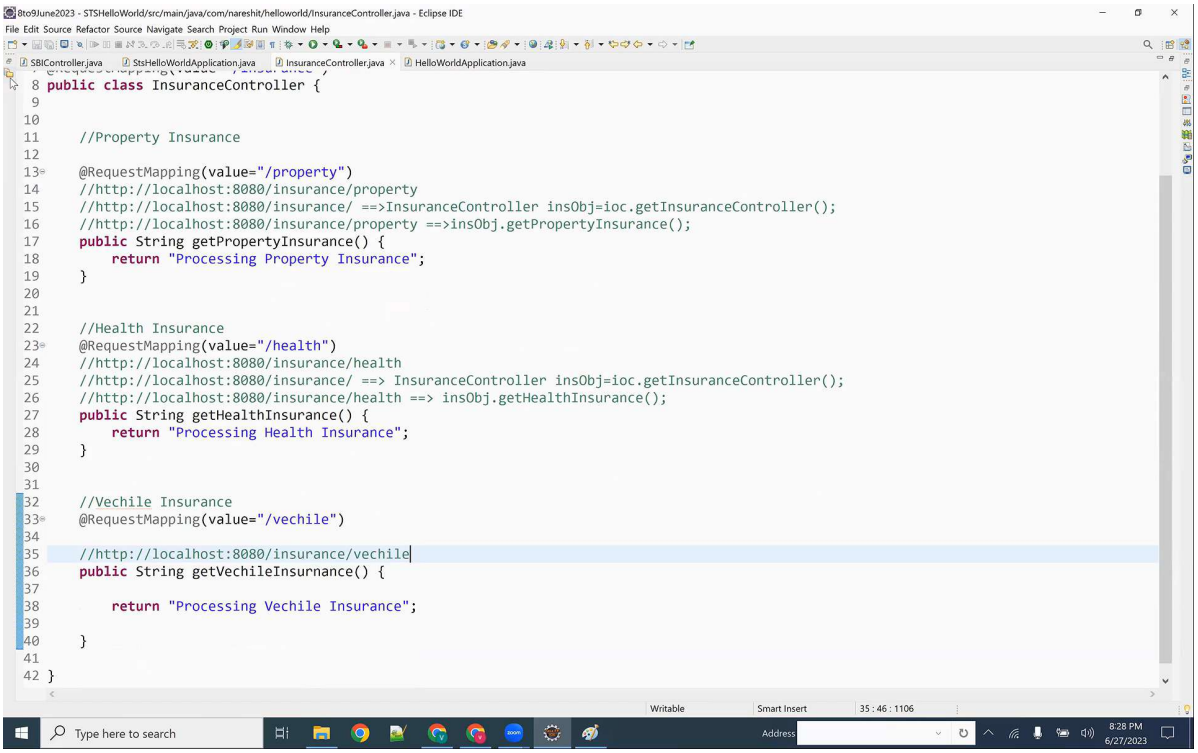
- bootstrap class is going identify which functaionalities to be loaded on startup/which beans to be start on startup.
- It is going to idenity which jars to be loaded on startup.


8to9June2023 - STSHelloWorld/src/main/java/com/nareshit/helloworld/SBIController.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
SBIController.java
StsHelloWorldApplication.java
InsuranceController.java
HelloWorldApplication.java
1 package com.nareshit.helloworld;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4
5
6 @RestController
7 @RequestMapping(value="/sbi")
8
9 /*
10 *
11 * DispatcherServlet
12 *
13 *
14 * /http://localhost:8080/sbi/personal
15 *
16 * http://localhost:8080==> Bootstrap Class
17 * /sbi ==> SBIController obj=ioc.getSBIController();
18 * /personal ==> obj.getPersonalLoan();
19 */
20 public class SBIController {
21
22
23
24 //Personal Loan
25 @RequestMapping(value="/personal")
26 //http://localhost:8080/sbi/personal
27 public String getPersonalLoan() {
28 return "Reached to Personal Processing Section";
29 }
30
31 //Home Loan
32 @RequestMapping(value="/homeloan")
33 //http://localhost:8080/sbi/homeloan
34 public String getHomeLoan() {
35 return "Reached to Home Loan Processing Section";
36 }
Writable
Smart Insert
18 : 45 : 458
Type here to search
Address
8:16 PM
6/27/2023


8to9June2023 - STSHelloWorld/src/main/java/com/nareshit/helloworld/InsuranceController.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
SBIController.java
StsHelloWorldApplication.java
InsuranceController.java
HelloWorldApplication.java
1 package com.nareshit.helloworld;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 @RequestMapping(value="/insurance")
8 public class InsuranceController {
9
10
11 @RequestMapping(value="/personal")
12 //http://localhost:8080/insurance/personal
13 /*
14 * http://localhost:8080==> Bootstrap Class
15 * /insurance ==> InsuranceController insObj=ioc.getInsuranceController();
16 * /personal ==> insObj.getPersonalLoan();
17 */
18 public String getPersonalLoan() {
19
20 return "Personal Insurance";
21 }
22
23 }
24
Writable
Smart Insert
16 : 33 : 506
Type here to search
Address
8:17 PM
6/27/2023

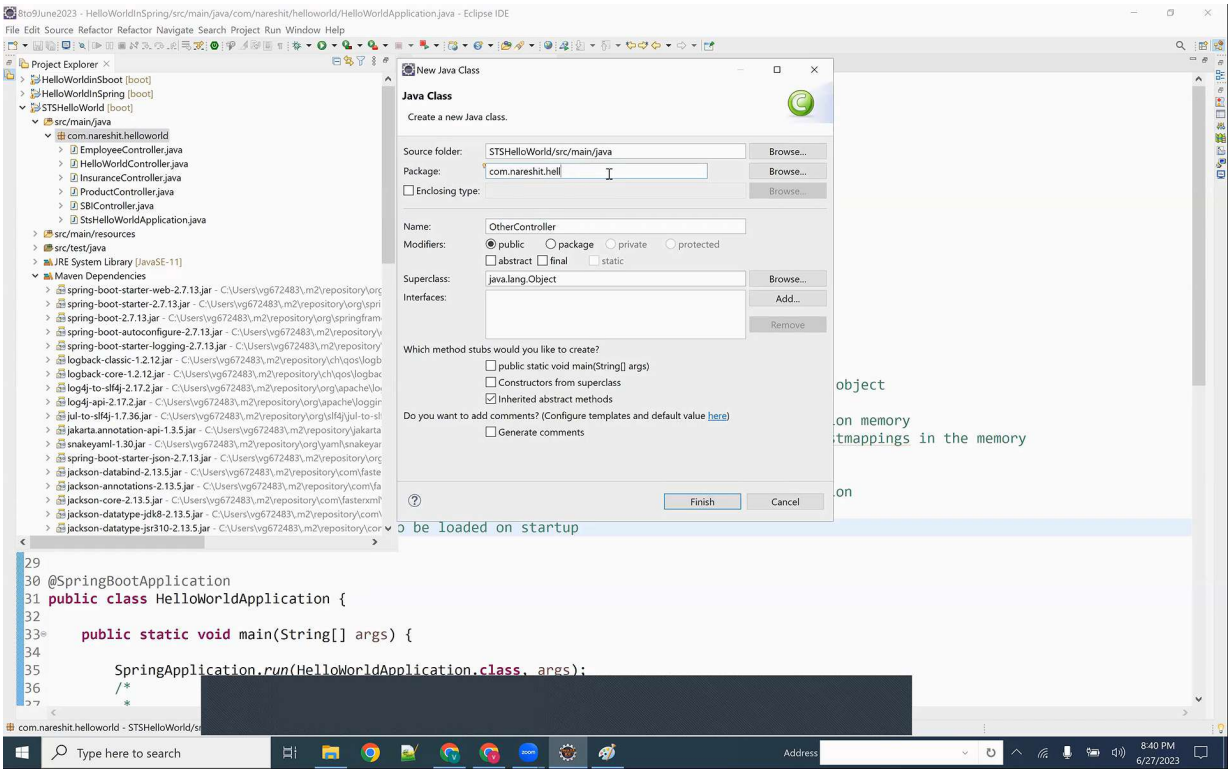


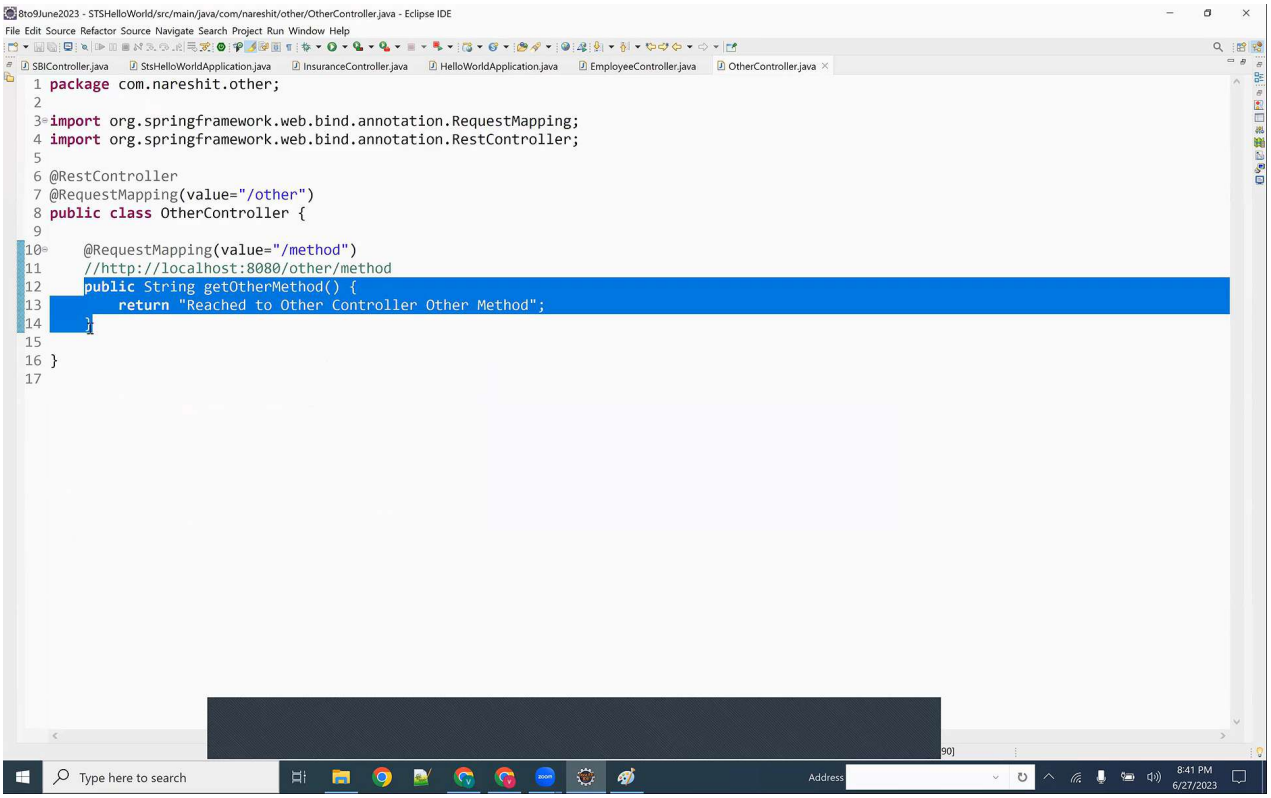
```java
package com.nareshit.helloworld;

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

@RestController
@RequestMapping(value="/insurance")
public class InsuranceController {


    //Property Insurance

    @RequestMapping(value="/property")
    //http://localhost:8080/insurance/property
    //http://localhost:8080/insurance/ ==>InsuranceController insObj=ioc.getInsuranceController();
    //http://localhost:8080/insurance/property ==>insObj.getPropertyInsurance();
    public String getPropertyInsurance() {
        return "Processing Property Insurance";
    }


    //Health Insurance
    @RequestMapping(value="/health")
    //http://localhost:8080/insurance/health
    //http://localhost:8080/insurance/ ==> InsuranceController insObj=ioc.getInsuranceController();
    //http://localhost:8080/insurance/health ==> insObj.getHealthInsurance();
    public String getHealthInsurance() {
        return "Processing Health Insurance";
    }


    //Vechile Insurance
    public String getVechileInsurnance() {

    }
```



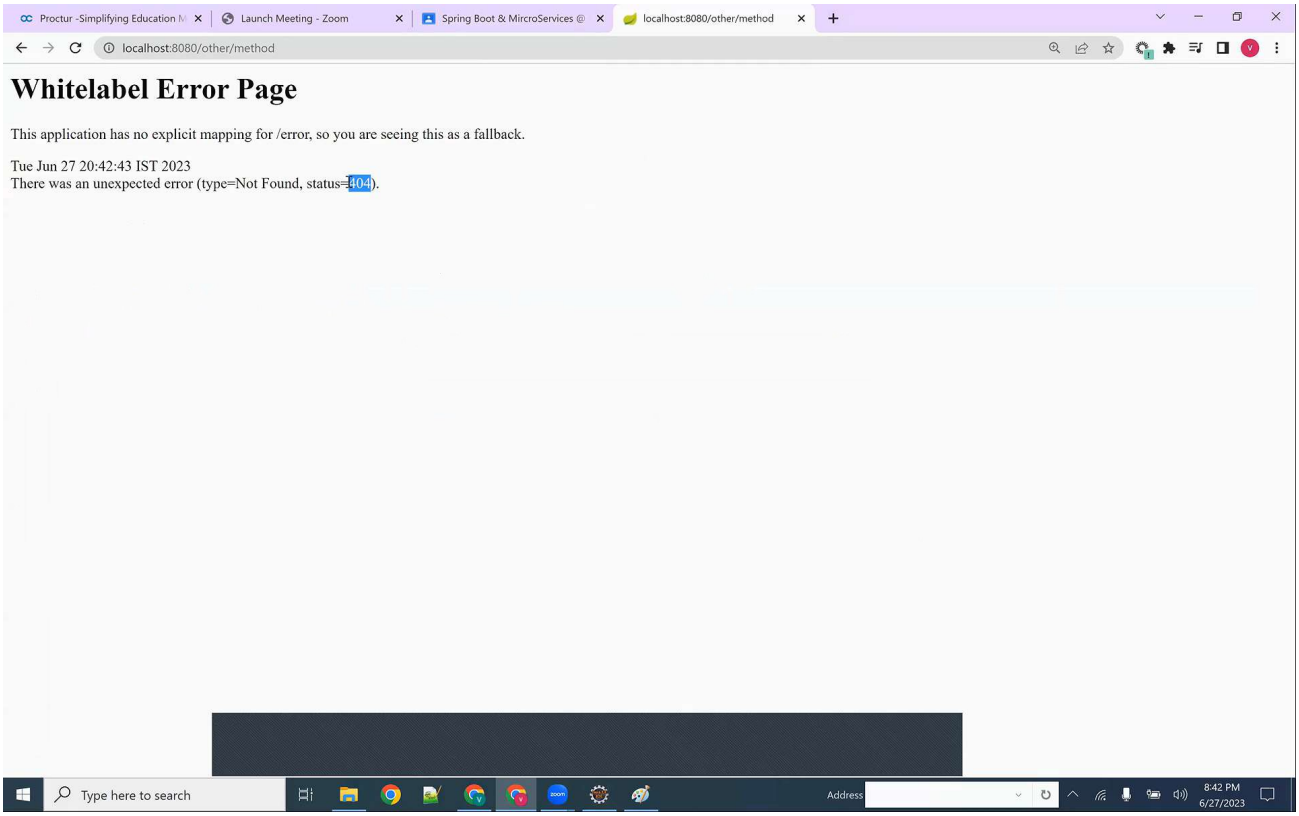
```java
public class InsuranceController {


    //Property Insurance

    @RequestMapping(value="/property")
    //http://localhost:8080/insurance/property
    //http://localhost:8080/insurance/ ==>InsuranceController insObj=ioc.getInsuranceController();
    //http://localhost:8080/insurance/property ==>insObj.getPropertyInsurance();
    public String getPropertyInsurance() {
        return "Processing Property Insurance";
    }


    //Health Insurance
    @RequestMapping(value="/health")
    //http://localhost:8080/insurance/health
    //http://localhost:8080/insurance/ ==> InsuranceController insObj=ioc.getInsuranceController();
    //http://localhost:8080/insurance/health ==> insObj.getHealthInsurance();
    public String getHealthInsurance() {
        return "Processing Health Insurance";
    }


    //Vechile Insurance
    @RequestMapping(value="/vechile")

    //http://localhost:8080/insurance/vechile
    public String getVechileInsurnance() {

        return "Processing Vechile Insurance";

    }

}
```

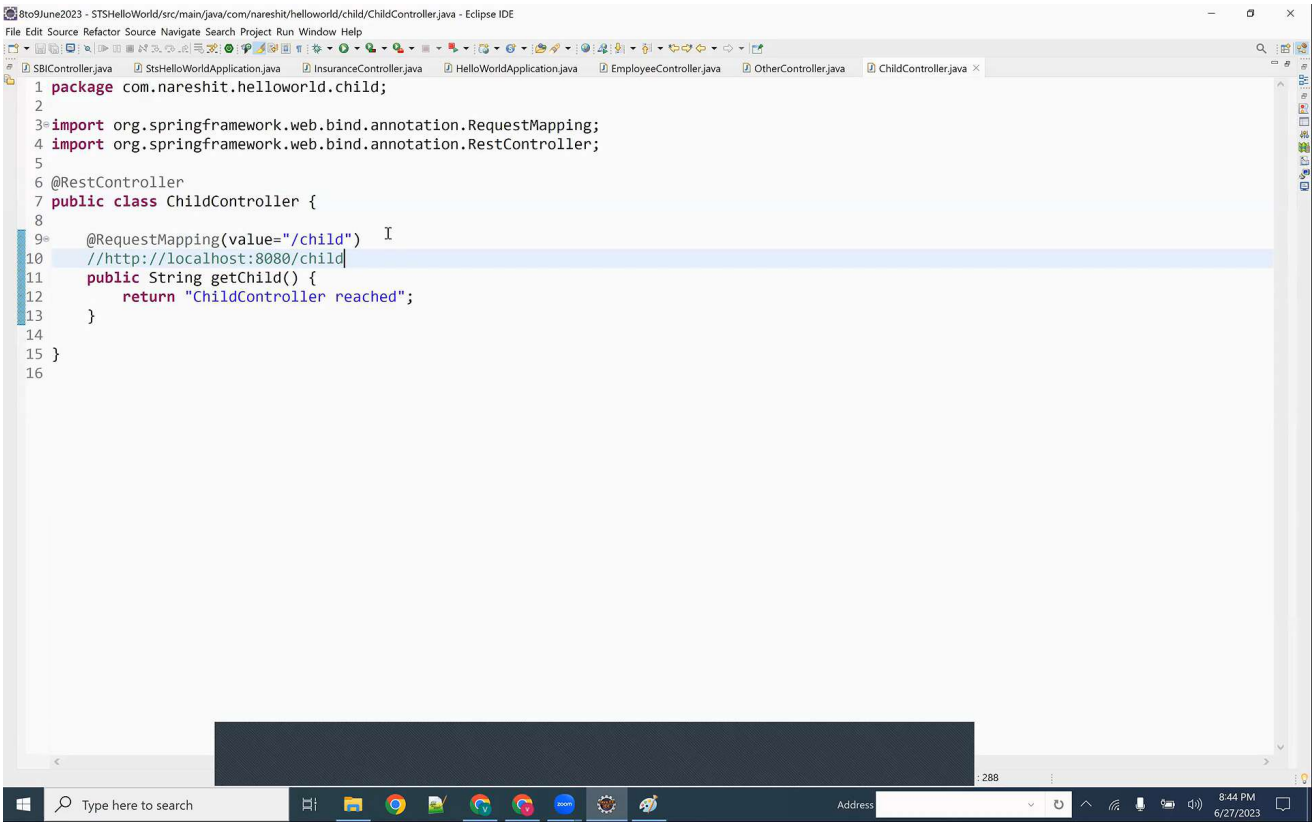
==========





when we run the application
and hit the above mentioned url,
it won't work or don't give the output.
because, the above class is created outside of the bootstrap class package.





for this url we will get the output, because this is created inside the bootstrap class package (subpackage of bootstrap class package.)

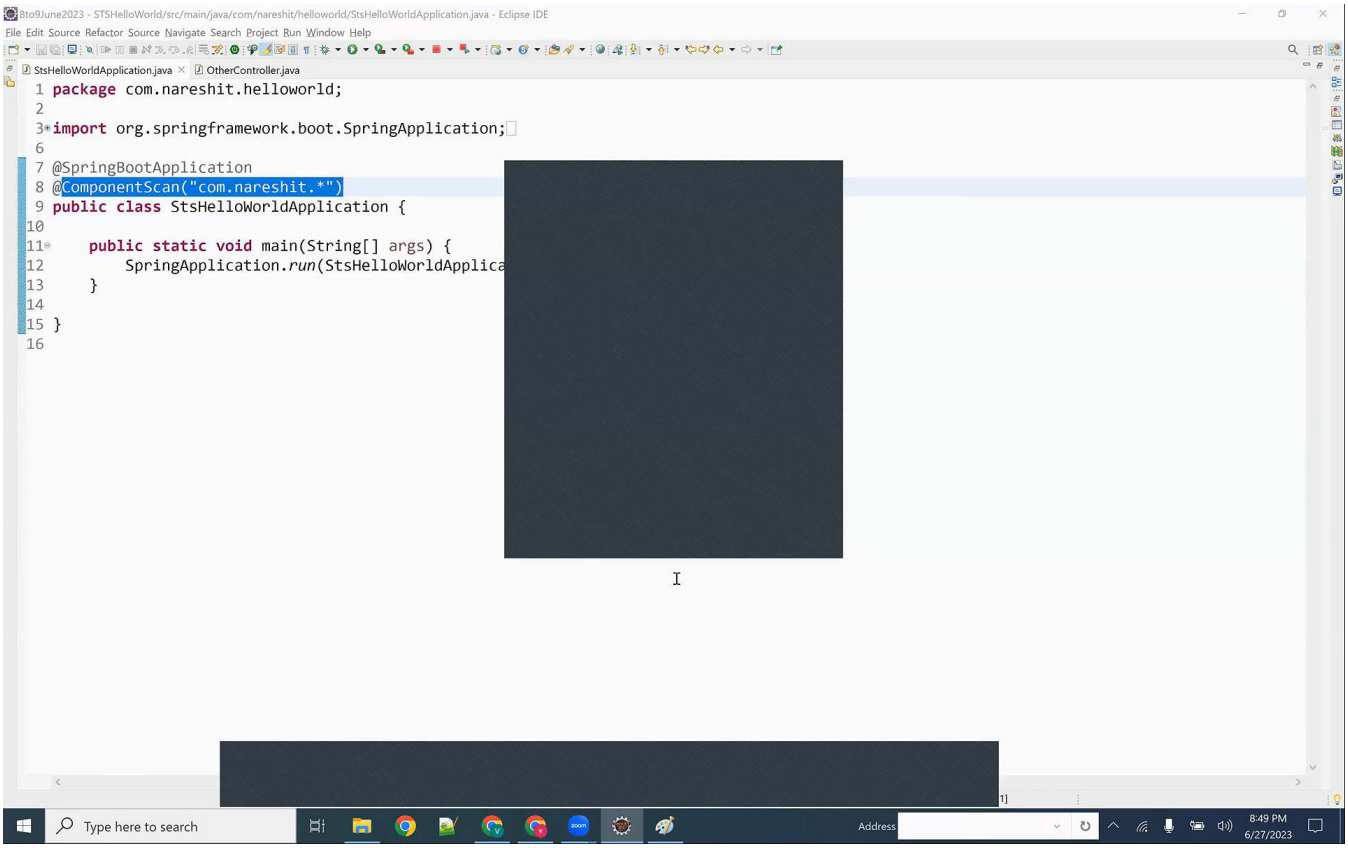
Inteview Question ?

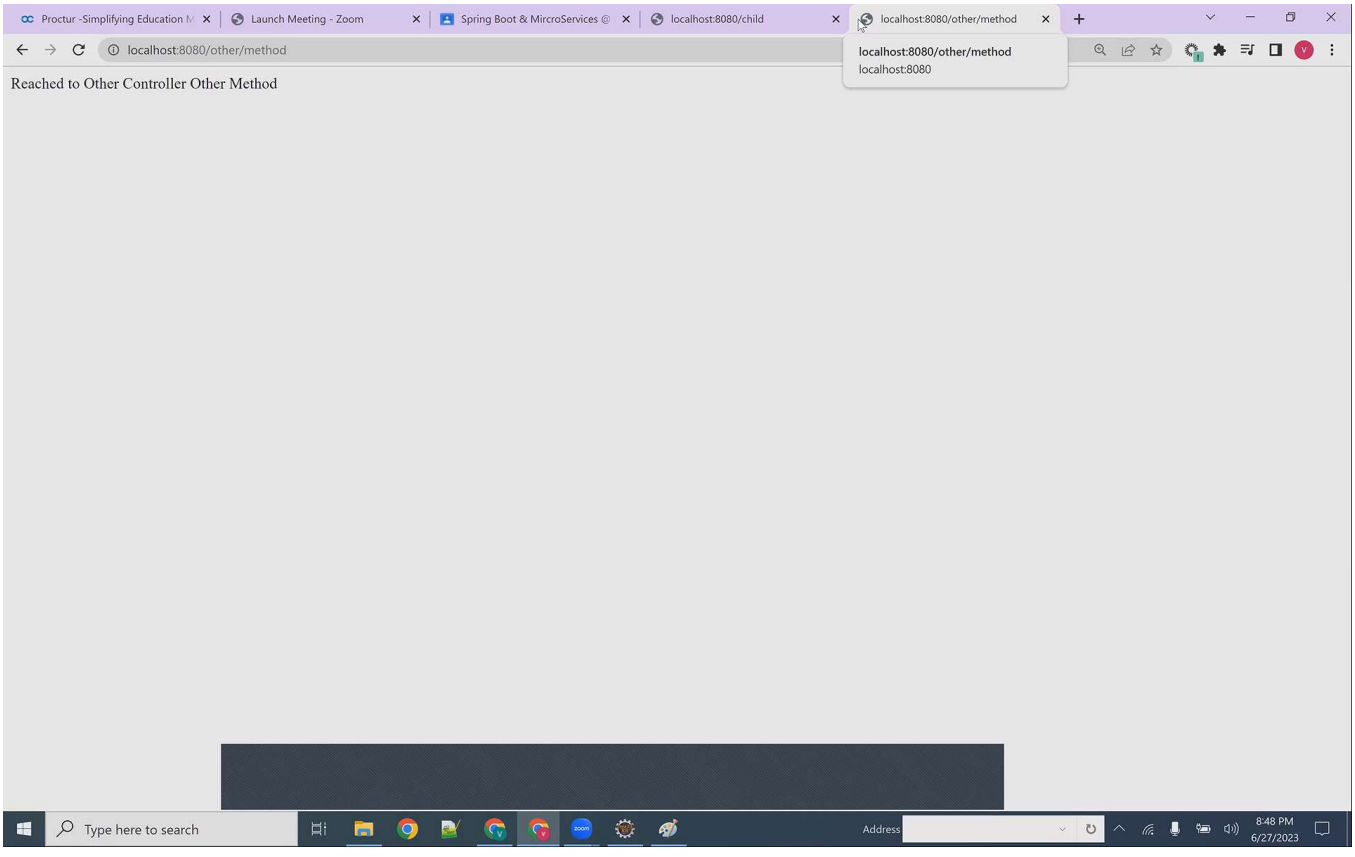
If I want to load/register a class, which is created outiside of the bootstarp class package. is it possible or not ?

- yes it is possible
- it can be possible by using annotation called "@ComponentScan"

- by default ,
  --> @ComponentScan("com.nareshit.helloworld")

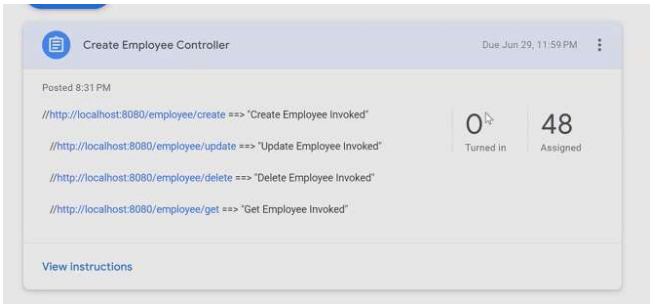
- we have to modify like,
  --> @ComponentScan("com.nareshit.*)





==========

Assignment



==========

- from tomorrow class we will start from -- JPA --> Java Persistance API
- without jdbc can we connect to database ?
    - Java is OOP Language
    - DB language is SQL
    - jdbc bridges between java and db