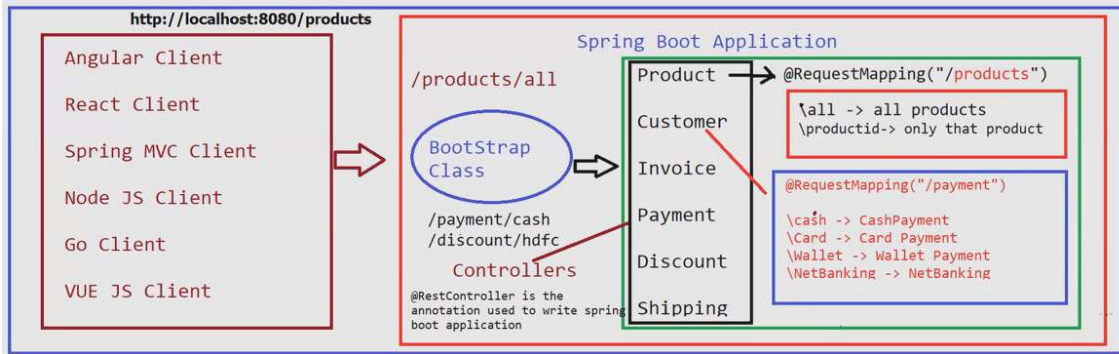


What is @RestController

3) @RestController -> This is 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.



```

10 *
11 * Every Spring Boot Project should have only one bootstrap class
12 *
13 * Any predefined keyword with "@" Symbol is called as annotation
14 *
15 * Functionality of Bootstrap Class
16 *
17 * 1) It will create IOC Container ==> Inversion of Control Container
18 * 2) It will Manage all memory management for both System Classes , Developer Classes
19 * Managing ==> Creating the object, maintaining the state of an object, and destroying the object
20 * 3) It will create a web application context==>
21 * All the classes written by developer moved to IOC Container in temporary web application memory
22 * 4) It will create dispatcher servlet ==> It will register all rest controllers and all requestmappings in the memory
23 * Any request comes ==> It will redirect to a specific method
24 * Rules ==> Every URL Mapping should be unique for application
25 * ==> @RequestMapping + @RestController Mapping ==> Unique through out application
26 *
27 *
28 */

```

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

What is the meaning of Dispatch Servlet ?

- Bootstrap class will create dispatcher servlet.
- while boot strap class is running, dispatcher 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 functionalities to be loaded on startup/which beans to be start on startup.
- It is going to identify which jars to be loaded on startup.

```
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 }
37 }
```

```
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
12 @RequestMapping(value="/personal")
13 //http://localhost:8080/insurance/personal
14 /*
15 * http://localhost:8080==> Bootstrap Class
16 * /insurance ==> InsuranceController insObj=ioc.getInsuranceController();
17 * /personal ==> insObj.getPersonalLoan();
18 */
19 public String getPersonalLoan() {
20     return "Personal Insurance";
21 }
22
23 }
24 }
```

```

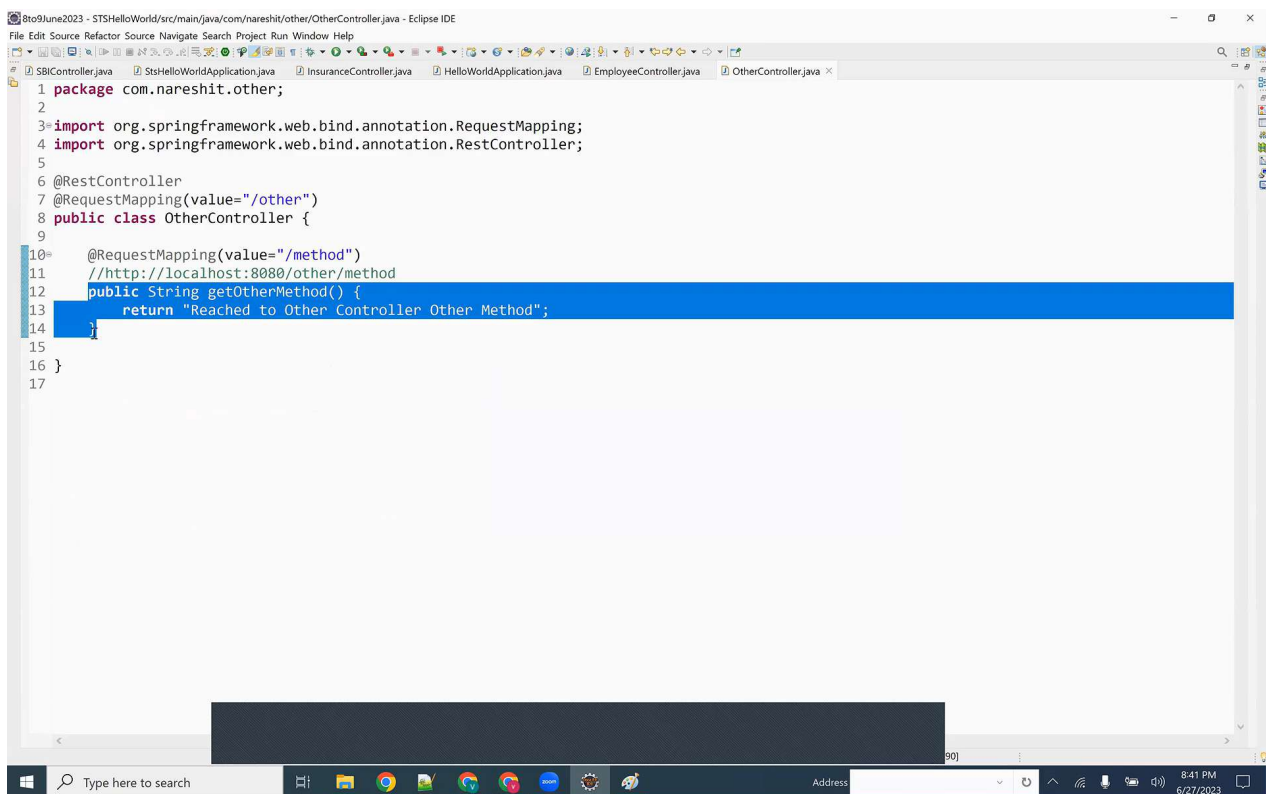
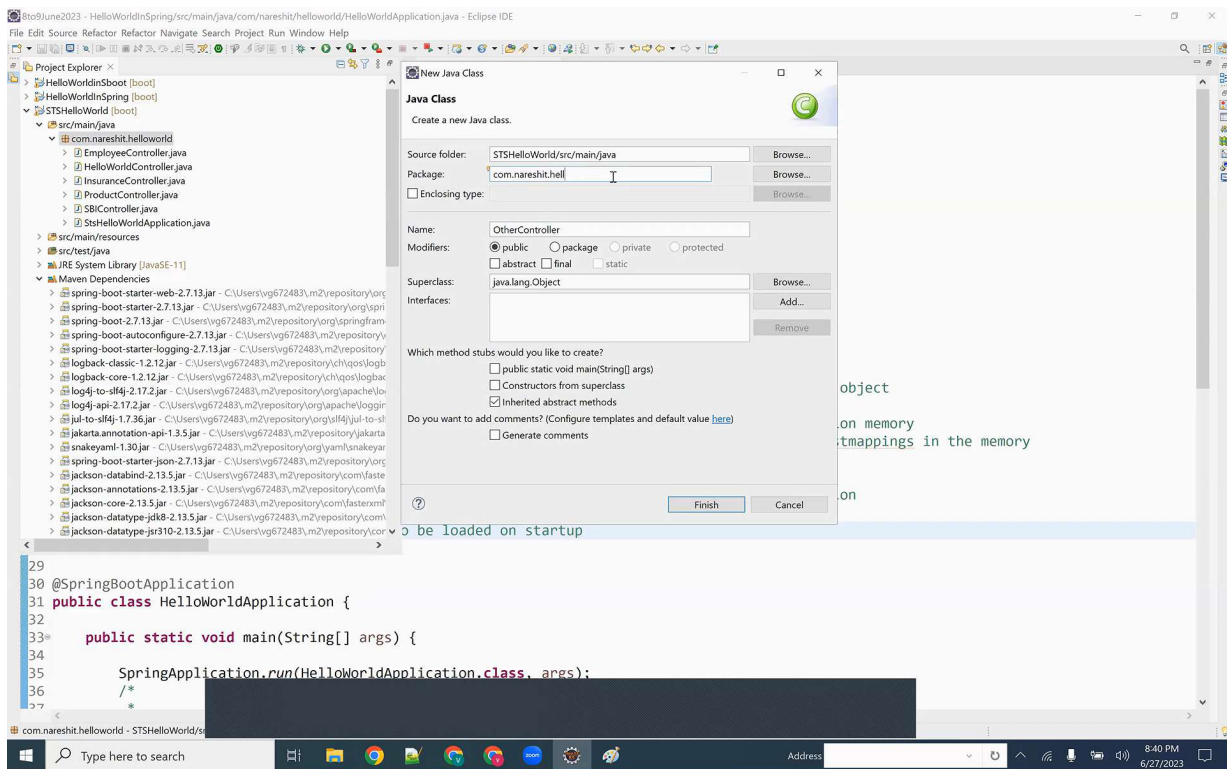
Bto9June2023 - STSHelloWorld/src/main/java/com/nareshit/helloworld/InsuranceController.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
SBIController.java STSHelloWorldApplication.java InsuranceController.java HelloWorldApplication.java
1 package com.nareshit.helloworld;
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     //Property Insurance
11
12
13     @RequestMapping(value="/property")
14     //http://localhost:8080/insurance/property
15     //http://localhost:8080/insurance/ ==>InsuranceController insObj=ioc.getInsuranceController();
16     //http://localhost:8080/insurance/property ==>insObj.getPropertyInsurance();
17     public String getPropertyInsurance() {
18         return "Processing Property Insurance";
19     }
20
21
22     //Health Insurance
23     @RequestMapping(value="/health")
24     //http://localhost:8080/insurance/health
25     //http://localhost:8080/insurance/ ==> InsuranceController insObj=ioc.getInsuranceController();
26     //http://localhost:8080/insurance/health ==> insObj.getHealthInsurance();
27     public String getHealthInsurance() {
28         return "Processing Health Insurance";
29     }
30
31
32     //Vechile Insurance
33     public String getVechileInsurance() {
34
35     }
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

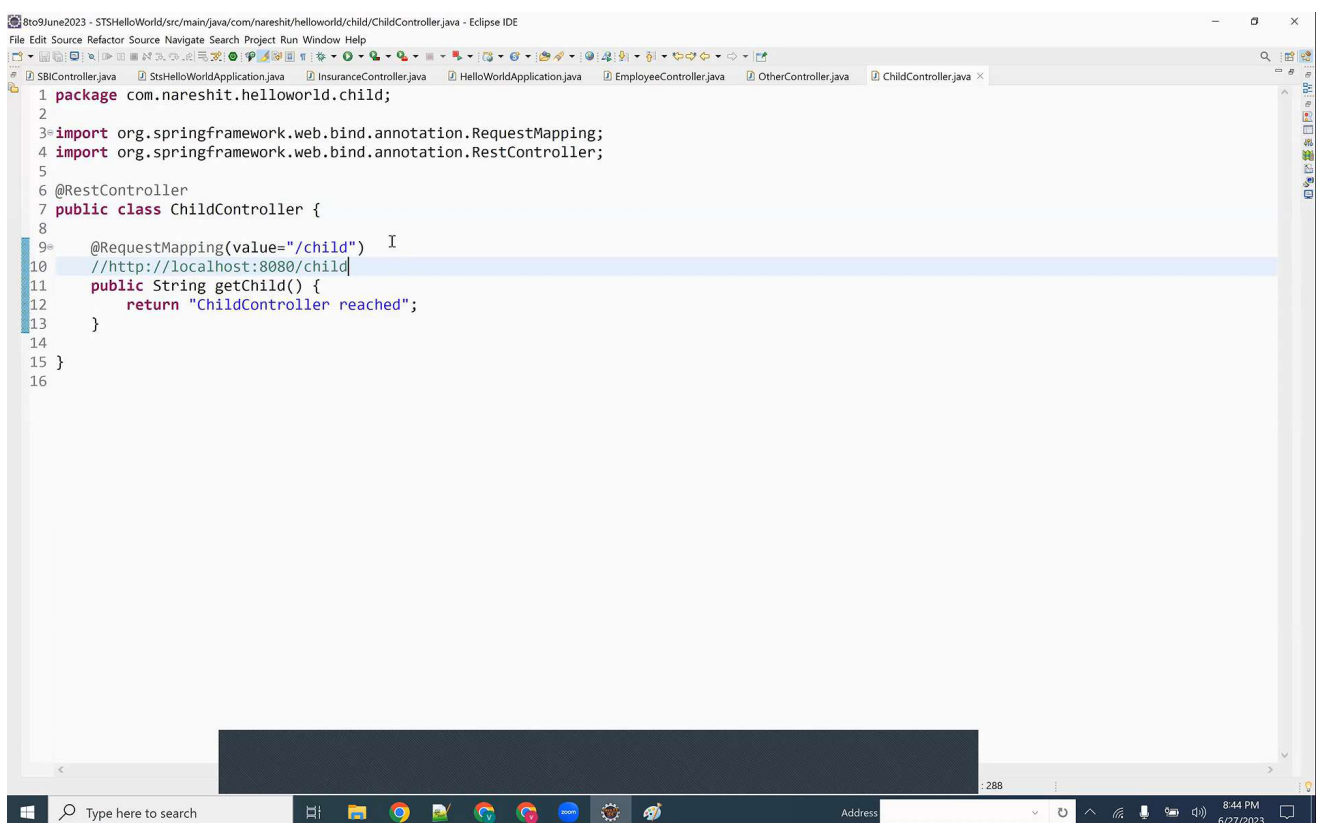
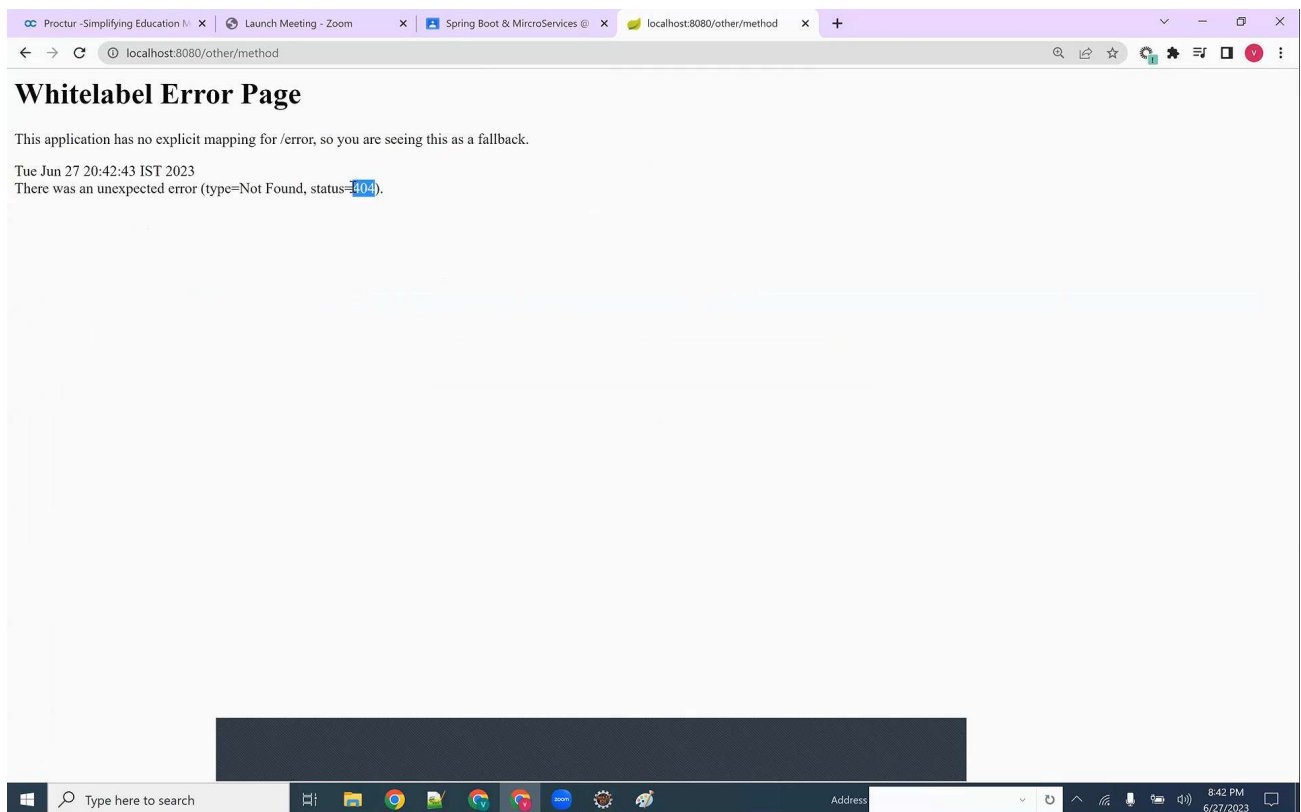
```

Bto9June2023 - 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
8 public class InsuranceController {
9
10
11     //Property Insurance
12
13     @RequestMapping(value="/property")
14     //http://localhost:8080/insurance/property
15     //http://localhost:8080/insurance/ ==>InsuranceController insObj=ioc.getInsuranceController();
16     //http://localhost:8080/insurance/property ==>insObj.getPropertyInsurance();
17     public String getPropertyInsurance() {
18         return "Processing Property Insurance";
19     }
20
21
22     //Health Insurance
23     @RequestMapping(value="/health")
24     //http://localhost:8080/insurance/health
25     //http://localhost:8080/insurance/ ==> InsuranceController insObj=ioc.getInsuranceController();
26     //http://localhost:8080/insurance/health ==> insObj.getHealthInsurance();
27     public String getHealthInsurance() {
28         return "Processing Health Insurance";
29     }
30
31
32     //Vechile Insurance
33     @RequestMapping(value="/vechile")
34
35     //http://localhost:8080/insurance/vechile
36     public String getVechileInsurance() {
37
38         return "Processing Vechile Insurance";
39     }
40
41
42 }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```



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.



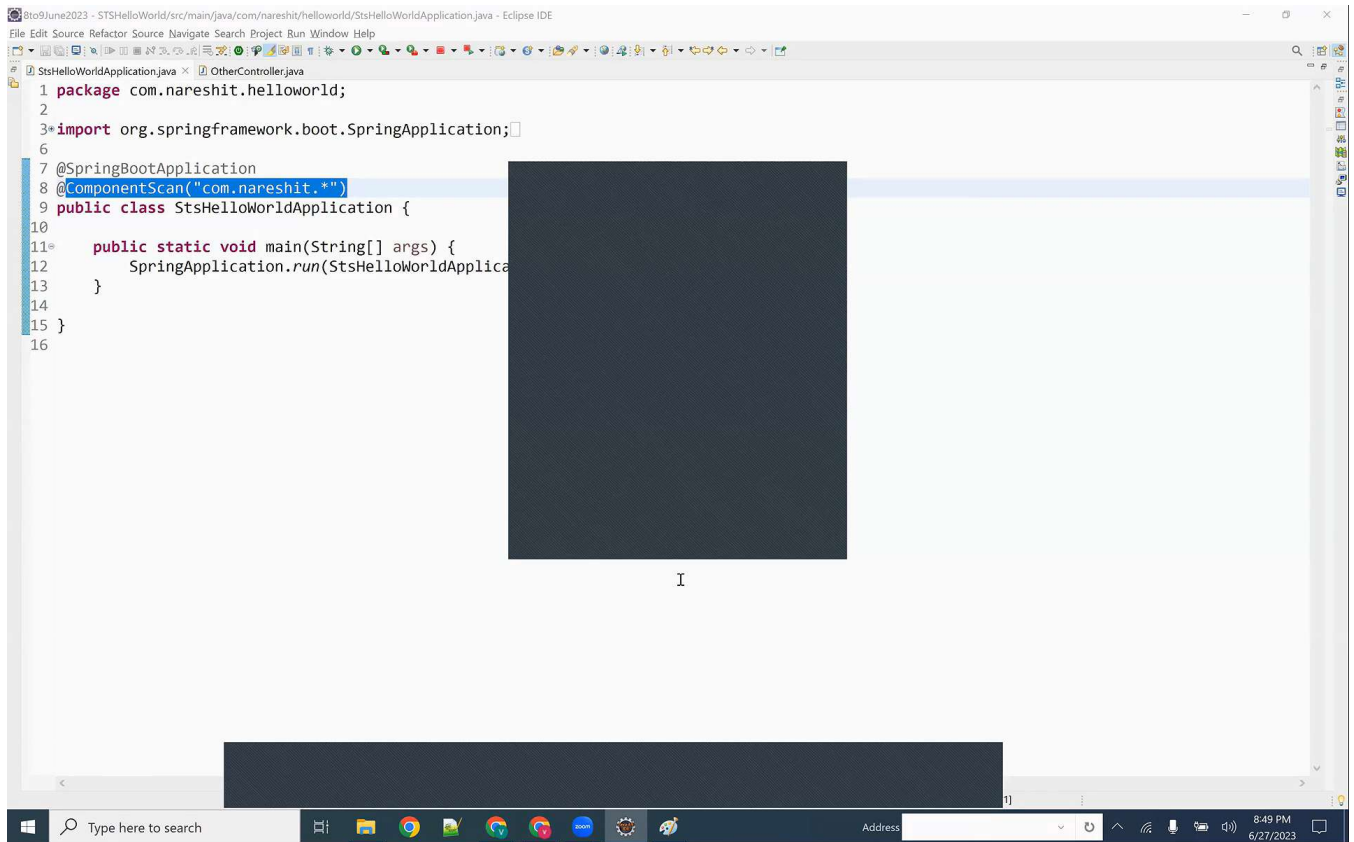
for this url we will get the output, because this is created inside the bootstrap class package (subpackage of bootstrap class package.)

Interview Question ?

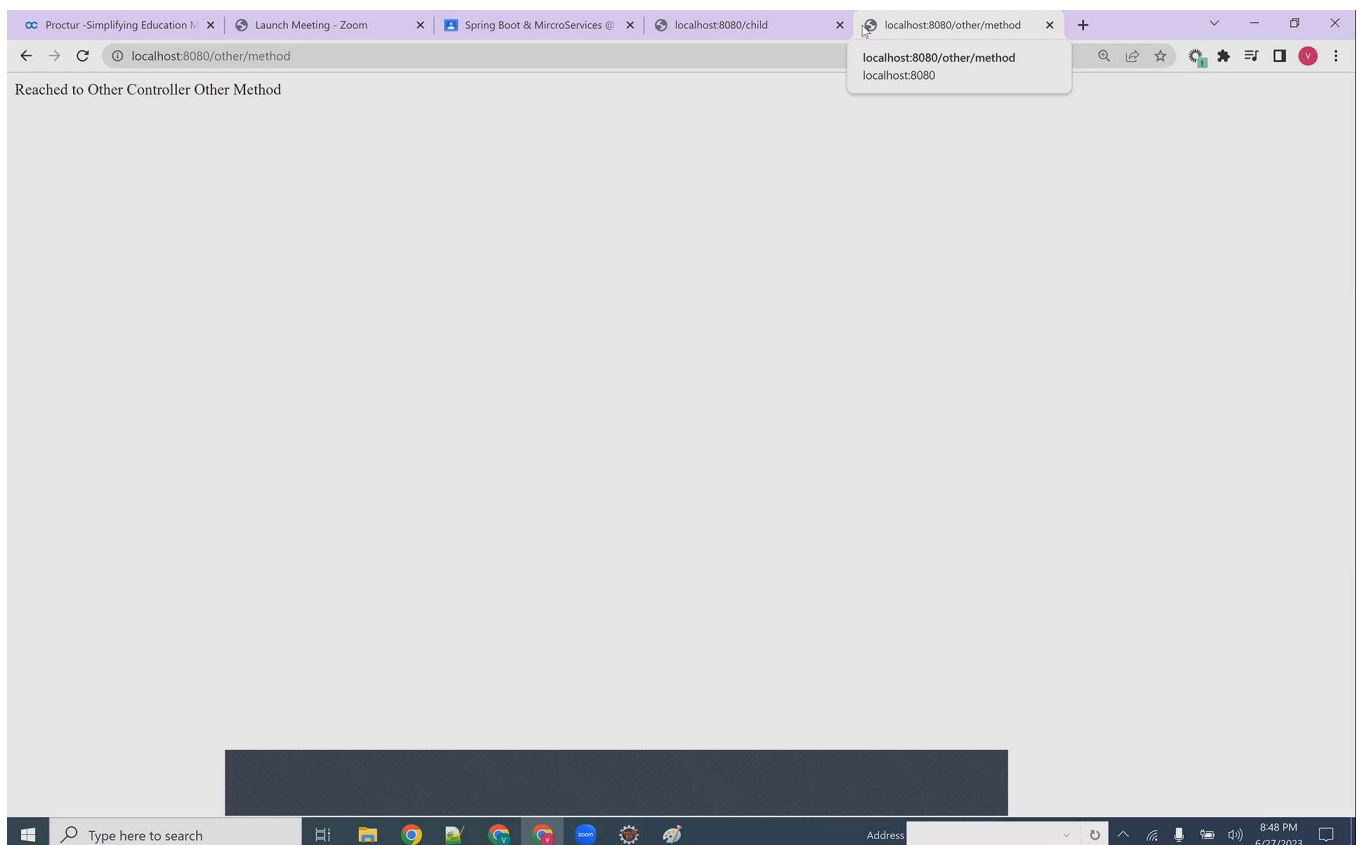
If I want to load/register a class, which is created outside of the bootstrap class package. is it possible or not ?

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

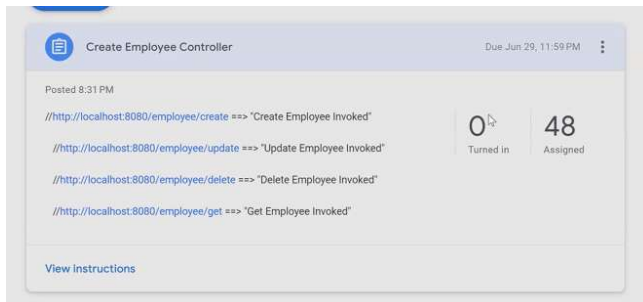
- by default ,
--> `@ComponentScan("com.nareshit.helloworld")`
- we have to modify like,
--> `@ComponentScan("com.nareshit.*")`



```
1 package com.nareshit.helloworld;
2
3 import org.springframework.boot.SpringApplication;
4
5
6
7 @SpringBootApplication
8 @ComponentScan("com.nareshit.*")
9 public class StsHelloWorldApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(StsHelloWorldApplication.class, args);
13     }
14 }
15
16
```



Assignment



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

JPA -> Java Persistence API

In java framework, if we want to work with database -> we need JDBC -> Java Database Connectivity .

JDBC code is tightly coupled to database. All DML,DQL and DDL -> will vary from database to database.

All the JDBC code which i wrote in java -> Needs to be rewritten

ORM -> Object Relational Mapping -> Java Class -----> RDBMS -> Mapping -> **Hibernate**

1) If we change database no impact to my java code
2) Not having any database code in java side.

Data JPA,Hibernate,Ibatis,CrudRepository,JpaRepository ->

Spring is using EntityManager and spring boot created two advaced repo's -> CRUD and JPA

TicketBooking Application

We will use In memory database which willll store all the tickets information.
We will use CRUDRepository for alll database operations
We will user RestController as the front end engine for booking the tickets