Date : 17/02/2021
Spring Boot 9AM
Mr. RAGHU

Web MVC : Working with Controller

Q) Can we define two method with path '/' in one/different controller ?
A) YES. But method type must be different.

```
   m1() --  /  + GET
   m2() --  /  + POST

   is valid.
```

Q) Who will create Controller class object?
A) Spring Container creates object to our class.

Q) Who will call controller methods and when?
A) FC(DS) calls controller#methods on every request.

```
*) In servlets 3 life cycle methods are there
     init()   : executed only once before first request
   service() : executed for every request
   destory() : executed only once while stopping app.
```

*) In Spring (Core) 2 life-cycle methods :
   init (@PostConstruct), destroy(@PreDestroy)

destory method will : close/clean/resource clearing.

-----Controller Example-----------------------

```
package in.nareshit.raghu.controller;

import javax.annotation.PostConstruct;
import javax.annotation.PreDestroy;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/emp")
public class EmpController {

	public EmpController() {
		System.out.println("OBJ CREATED");
	}

	@PostConstruct
	public void mysetup() {
		System.out.println("FROM INIT");
	}

	@RequestMapping("/save")
```

```
        public String saveEmp() {
                System.out.println("FROM SERVICE METHOD");
                return "EmpSave";
        }

        @PreDestroy
        public void clearsetup() {
                System.out.println("FROM DESTROY");
        }
}
-------------------------------------------------
```

*) init() and destory() are executed only once.

*) init() method is called as after creating object so annotated with @PostConstruct (POST-After, Construct-creating object)

*) When we start server, then container creates object and calls init() method.

*) When even request made by browser, then FC executes methods in controller.

*) When we are about to stop server. Then before closing ApplicationContext (Spring Container) destory() method is executed.

*) destory method clears the resources and closes the connections/external links It is also executed by containe itself.

```
----------------------------------------------------------------------
-
*) Spring boot has provided 3 embedded webservers.
  Apache  - Tomcat    [deful]
  JBoss   - Undertow
  Eclipse - Jetty.
----------------------------------------------------------------
EJB -- ??--- Maintances App (old apps) ---> App Servers.
---------------------------------------------------------
```

*) By default [Default Servlet ie] path with / is mapped to index.html exist in static folder

```
ProjectName
  |-src/main/resources
     |-static
        |-index.html
```

URL: http://localhost:8080

*) If we define a Controller method with path '/' then it wont.

```
http://localhost:80    => http://localhost
http://Sys:80          => http://Sys
http://192.168.0.1:80  => http://192.168.0.1
```

(cmd > ipconfig)

server.port=80 in Boot WEB MVC App.

=> To run Spring Boot web application on other embedded servers

a. Exclude Tomcat server from spring-boot-starter-web
b. Add other embeeded server dependency

Step#1
From:

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

To:

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
        <exclusions>
                <exclusion>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-
tomcat</artifactId>
                </exclusion>
        </exclusions>
</dependency>
```

Step#2 Add any one below

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-undertow</artifactId>
</dependency>

<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-jetty</artifactId>
</dependency>
```

Q) How can we convert Spring Boot App to war?
   How can we do external deployment ? (any server)

A) Create Project as WAR and run where you want.

```
> While creating project Choose Packing as : war
> Right click on project (After coding) > Run As > maven install
> Refresh project
> Goto targte folder
> Copy war file into Server folder
```

Ex: Place war file under location:
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps

```
> Starter External server.
```

```
C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin
Click on Tomcat9.exe
----------------------------------------------------------
Q) How can we convert we project into jar/war and run?
A)
-> Define Spring Boot Code
-> Right click on project > Run as > Maven clean and install
-> Check jar under target folder
-> Execute below commands
   java -jar  <jarFileName>.jar
-> Enter URL in browser
  http://localhost:8080/emp
--------------------------------------------------------------
```