Date : 16/02/2021
Spring Boot 9AM
Mr. RAGHU

Web MVC : Controller

*) Rules to write controller:-

1. URL(Path) is case-sensitive. ie /show, /SHOW, /Show are different.

```
@RequestMapping("/show") --> m1()
@RequestMapping("/SHOW") --> m2()
```

--full code-------

a) Name : SpringBoot2ContorllerRules
   Dep : Spring Web

b) add in pom.xml

```
<dependency>
        <groupId>org.apache.tomcat.embed</groupId>
        <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
```

c) folder system

```
 src
  |- main
      |- webapp
           |- WEB-INF
               |- pages
```

d) properties file

```
server.port=9090
spring.mvc.view.prefix=/WEB-INF/pages/
spring.mvc.view.suffix=.jsp
```

f) Controller code

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

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

@Controller
public class EmpController {

        @RequestMapping("/show")
        public String showA() {
                return "Home";
        }

        @RequestMapping("/SHOW")
        public String showB() {
                return "Setup";
        }
}
```

g) UI Pages

```
--Home.jsp---
<html>
<body>
<h3>WELCOME TO HOME PAGE!</h3>
</body>
</html>
--Setup.jsp--
<html>
<body>
<h3>WELCOME TO SETUP PAGE!</h3>
</body>
</html>
```

h) Run app and Enter URLs

```
http://localhost:9090/show
http://localhost:9090/SHOW
```

i)** Stop server before running app.

```
If we try to run again without stopping app:
Web server failed to start. Port 9090 was already in use.
```

---

2. Duplicates paths are valid and allowed with unique combination of Method Type

```
Ex(Valid)
   /show + GET   -- m1()
   /show + POST  -- m2()
```

```
--code--
a) Controller
package in.nareshit.raghu.controller;

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

@Controller
public class EmpController {

        @RequestMapping(value = "/show",method = RequestMethod.POST)
//POST
        public String showA() {
                return "Home";
        }

        //@RequestMapping("/show") //GET
        @RequestMapping(value = "/show",method = RequestMethod.GET)
//GET
        public String showB() {
                return "Setup";
        }
}
```

b) UI Pages:

```
------Home.jsp-------
```

```
<html>
<body>
<h3>WELCOME TO HOME PAGE!</h3>
</body>
</html>
--Setup.jsp-----------
<html>
<body>
<h3>WELCOME TO SETUP PAGE!</h3>
<form action="/show" method="POST">
        <input type="submit" value="View Home"/>
</form>
</body>
</html>

*) URL: http://localhost:9090/show
-------------------------------------
Ex(Invalid)
   /show + POST   -- showA()
   /show + POST   -- showB()

a) Controller
package in.nareshit.raghu.controller;

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

@Controller
public class EmpController {

        @RequestMapping(value = "/show") //GET
        public String showA() {
                return "Home";
        }

        @RequestMapping("/show") //GET
        public String showB() {
                return "Setup";
        }
}
--------------------------

IllegalStateException: Ambiguous mapping.
    Cannot map 'empController' method
in.nareshit.raghu.controller.EmpController#showB()
to { [/show]}: There is already 'empController' bean method
in.nareshit.raghu.controller.EmpController#showA() mapped.
```

```
3. We can provide multiple paths at one method even.

@RequestMapping(value={
    "/login", "/home", "/logout"
})
Come to HomePage, for all above URL.

--code--
a)Controller
```

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

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

@Controller
public class UserController {

	@RequestMapping(value = {
			"/login",
			"/home",
			"/logout"
			})
	public String showHome() {
		return "HomePage";
	}
}
```

b) UI Pages

```
--HomePage.jsp--
<html>
<body>
<h3>WELCOME TO USER HOME/LOGIN PAGE!</h3>
</body>
</html>
```

c) Request URL:

```
http://localhost:9090/login
http://localhost:9090/home
http://localhost:9090/logout
```

****

4. For One method we can provide - Multiple Method Types(GET/POST)

```
@RequestMapping(
    value="/home",
    method={
        RequestMethod.GET,
        RequestMethod.POST
    })
```

If we make request /home with GET/POST goto HomePage only.

--code--

a) Controller

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

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

@Controller
public class EmpController {

	@RequestMapping(
			value = "/home",
```

```
                        method = {
                                        RequestMethod.GET,
                                        RequestMethod.POST
                        })
        public String showA() {
                System.out.println("IT IS CALLED...");
                return "Home";
        }

}
```

b) UI Page

```
--Home.jsp--
<html>
<body>
<h3>WELCOME TO HOME PAGE!</h3>
<form action="/home" method="POST">
        <input type="submit" value="View Home"/>
</form>
</body>
</html>
-------------------
```

****

5. We can even provide multiple URLs and multiple method types.
   [Not used in realtime, technically valid, has no meaning]

   Combination of Point(c) and Point(d)

```
@RequestMapping(
   value={
    "/login", "/home", "/logout"
  },
  method={
        RequestMethod.GET,
        RequestMethod.POST
  }
)
```

*)For all above URL with GET/POST always show Inbox Page.

```
--code--
a) Controller
package in.nareshit.raghu.controller;

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

@Controller
public class EmpController {

        @RequestMapping(
                        value = {
                                        "/home",
                                        "/login",
                                        "/logout",
```

```
                                        },
                        method = {
                                        RequestMethod.GET,
                                        RequestMethod.POST
                        })
        public String showA() {
                System.out.println("IT IS CALLED...");
                return "Home";
        }

}
```

b) UI Page

--Home.jsp--

```
<html>
<body>
<h3>WELCOME TO HOME PAGE!</h3>
<form action="/home" method="POST">
        <input type="submit" value="View Home"/>
</form>
</body>
</html>
```

---

6. We can even provide path(URL) as '/'.
   (If we do not provide any path, default is /)
   Default Method type is : GET

```
@RequestMapping("/")
String m1() {

}
```

*) @RequestMapping("/") is equals to
   @RequestMapping is equals to
   @RequestMapping(value="/",method=RequestMethod.GET) is equals to
   @RequestMapping(method=RequestMethod.GET) [no path - default /]

*)It behaves like welcome page(default) setup in servlets.
   If we do not provide any method type default is GET.

--code--

a) Controller

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

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

@Controller
public class UserController {

        //@RequestMapping("/")
        //@RequestMapping
        //@RequestMapping(value = "/")
        //@RequestMapping(value = "/",method = RequestMethod.GET)
```

```
        @RequestMapping(method = RequestMethod.GET)
        public String showHome() {
                return "HomePage";
        }
}

b) UI Page:
--HomePage.jsp--
<html>
<body>
<h3>WELCOME TO USER HOME PAGE!</h3>
</body>
</html>

Req URL:
http://localhost:9090/

Q) Can we define two or more method in in controller with path '/'?
A) No. Any path is used only once with one method Type.
  But / with GET and / with POST is valid.
```

```
7. We can provide path at controller level also (But not method type).

*) If we have multiple controllers, then this is used.
*) It is recomanded in realtime applications.

@Controller
@RequestMapping("/emp")
class  EmpController {
  /save - POST -- saveEmp()
}

@Controller
@RequestMapping("/admin")
class AdminController {
   /save - POST -- saveAdmin()
}

--code---
a) Controllers
package in.nareshit.raghu.controller;

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

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

        @RequestMapping("/save")
        //@RequestMapping("/emp/save")
        public String saveEmp() {
                return "EmpSave";
        }

}
---
```

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

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

@Controller
@RequestMapping("/admin")
public class AdminController {

        @RequestMapping("/save")
        //@RequestMapping("/admin/save")
        public String saveAdmin() {
                return "AdminSave";
        }

}
```

b) UI Pages

--EmpSave.jsp--

```
<html>
<body>
<h3>WELCOME TO EMP SAVE!</h3>
</body>
</html>
```

--AdminSave.jsp--

```
<html>
<body>
<h3>WELCOME TO ADMIN SAVE!</h3>
</body>
</html>
```

*) Req URL

http://localhost:9090/emp/save

http://localhost:9090/admin/save

---

8. We can use even same path at method level which is given at class level. (Technically valid, has no meaning).

```
@Controller
@RequestMapping("/data")
class  EmpController {
  @RequestMapping("/data")
  String show(){}

}
```

Req URL looks like : http:// ..../data/data

--code--

a) Controller

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

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

@Controller
@RequestMapping("/emp")
```

```
public class EmpController {

        @RequestMapping("/emp")
        public String saveEmp() {
                return "EmpSave";
        }

}
```

b) UI Page

```
--EmpSave.jsp--
<html>
<body>
<h3>WELCOME TO EMPLOYEE SAVE!</h3>
</body>
</html>
-------
```

*) Req URL:

```
http://localhost:9090/emp/emp
```

Q) What is @RequestMapping ?
A) Link One URL(Path) and Method Type with one java class and method
  If browser makes request, compare those details with our class
  and method, if matching execute them.

Q) If we did not specify @Controller at class level ?
A) No Object is created and HTTP protocol not supported.
  You will see error 404 - NOT FOUND.

Q) Can we define a method in controller without @RequestMapping?
A) YES, 100% valid java method, but can never be executed by FC
   for any request.

```
@Controller
class  Sample {

  String show(){ //valid java method. never gets executed by FC.
     ...
  }
}
```

****
---

9. We can use advanced RequestMapping Annotations,

```
GET     @GetMapping("/path")    --
@RequestMapping(value="/path",method=RequestMethod.GET)

POST    @PostMapping("/path")    --
@RequestMapping(value="/path",method=RequestMethod.POST)

PUT     @PutMapping("/path")    --
@RequestMapping(value="/path",method=RequestMethod.PUT)

DELETE  @DeleteMapping("/path")    --
@RequestMapping(value="/path",method=RequestMethod.DELETE)
```

PATCH   @PatchMapping("/path")   --
@RequestMapping(value="/path",method=RequestMethod.PATCH)
---
10. If we set to server.port=80 then need not to provide
   port number in request URL.

ex:
  http://localhost:80/home   is equals to

  http://localhost/home

---
*) Note: Output: Whitelabel Error Page(type=Not Found, status=404).

Might be,
a) Entered URL in browser not matching with any controller method.
b) Forgot to add Tomcat Embedded JASPER (that will not convert JSP
pages)
c) Provided invalid folder names
   like
     WEBAPP instedof webapp
     WEB_INF, Web-Inf, WebINF instedof WEB-INF

e) Not providing / at prefix (at begining and ending)
(all are invalid)
   spring.mvc.view.prefix=WEB-INF/mypages
   spring.mvc.view.prefix=/WEB-INF/mypages
   spring.mvc.view.prefix=WEB-INF/mypages/
   spring.mvc.view.prefix=/WEB-INF/mypages/   (spaces added here)

   Insted of (valid)
   spring.mvc.view.prefix=/WEB-INF/mypages/

f) Controller is not under basepackage

  Starter class : in.nit.app

  Controller class:
      in.nit.app.abc (valid)
      in.nit.controller (invalid)

g) Forgot to write @RequestMapping at method.
