Date : 19/02/2021
Spring Boot 9AM
Mr. RAGHU

Spring Boot with Docker:-
https://www.youtube.com/c/NareshIT/search?query=docker%20raghu

Web MVC : Model

=> Model is a memory created and handled by Spring container.

=> This is used to share data from Controller to UI.
[only one direction, Controller->UI].

=> Model Stores data in Map Format (key,val)
Key--String type
Value--Object (Primitive, Class-Obj, Collection...etc)

=> Use this model Controller#method Param and call method
model.addAttribute(key,val).

Then data is added to Model memory.

=> At UI, use EL(Expression Language) : ${key}

=> keys are case-sensitive, must be used at controller and UI with matching names, else data can not be rendered.

Q) Is below code valid? How?
```
Object ob = 100;
```

A) Yes. 100 -> int -(Autoboxing)->
Integer -(Upcasting)-> Number-(Upcasting)->Object

By using AutoBoxing and Upcasting.
// object can store any data in java.

-----Servlets code-----

```
1--(source)Servlet  -->
RequestDispatcher rd = request.getRequestDispatcher("/WEB-
INF/pages/EmpHome.jsp");
request.setAttribute("eid",10);
rd.forward(req,resp);

1-- Servlet (destination)

Object ob = request.getAttribute("eid");
int id = (Integer)ob; //downcast and auto-unbox
```

===Example#1 Working with Primitive Data ==========

Name : SpringBoot2WebMvcModelEx
Dep : Spring Web, Lombok

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

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class EmployeeController {

	@GetMapping("/data")
	public String showData(Model model) {
		System.out.println(model.getClass().getName());

		//adding data to model
		model.addAttribute("eid", 100);
		model.addAttribute("ename", "ABC");
		model.addAttribute("esal", 3.3);

		return "EmpData";
	}
}
```

2. EmpData.jsp
```
<html>
<body>
<h3>
Data is : ${eid}, ${ename}, ${esal}
</h3>
</body>
</html>
```

3. application.properties
```
server.port=9698
spring.mvc.view.prefix=/WEB-INF/pages/
spring.mvc.view.suffix=.jsp
```

*) Enter URL; http://localhost:9698/data

*) Model(I) is a inteface impl class also given by Spring F/w
  BindingAwareModelMap(C), is if data exist then allocate memory for
  that at runtime, else clear data if not neededor used already.

=======Ex#2 Sending Object data ================================
*) We can send object data from Controller to UI using Model(I)
  memory, at UI we can read full object using ${key} then it
  internally calls toString() method. We can even read one by one
  value from object using ${key.variable}.

name: SpringBoot2WebMvcModelObjectEx
Dep : Spring web, Lombok

1. Model class
```
package in.nareshit.raghu.model;
```

```
import lombok.Data;

@Data
public class Employee {

        private Integer eid;
        private String ename;
        private Double esal;
}
```

2. Controller

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

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

import in.nareshit.raghu.model.Employee;

@Controller
public class EmployeeController {

        @GetMapping("/data")
        public String showData(Model model) {
                //System.out.println(model.getClass().getName());

                Employee emp = new Employee();
                emp.setEid(5896);
                emp.setEname("ABC-ZYZ");
                emp.setEsal(50000.10);

                model.addAttribute("employee", emp);

                return "EmpData";
        }
}
```

3. UI : EmpData.jsp

```
<html> <body>
<h3>
Data is : ${employee}
<br/>
Hello Mr.Ms/Mrs:  ${employee.ename}, your Id is : ${employee.eid},
  You got package of : ${employee.esal }
</h3>
</body> </html>
```

Req URL: http://localhost:9698/data

JSTL (JSP Standard Tag Library):
  Writing Java code in tags Format, looks like (HTML tags).

*) Wiring Loop using Core tags(Core Java For Each Loop at JSP Page)

Code Expected:

```
 List<T> list =.....
```

```
 for(T t:list) {
   //...
 }

Tags:
<c:forEach items="${keyFromModel}" var="tempLoopvariable">
  //...
</c:forEach>

Add Import Line:
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>


===Ex#3: Working with Collection Data=========================
1. Model
package in.nareshit.raghu.model;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@AllArgsConstructor
@NoArgsConstructor
@Data
public class Employee {

        private Integer eid;
        private String ename;
        private Double esal;
}

2. Controller
package in.nareshit.raghu.controller;

import java.util.List;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

import in.nareshit.raghu.model.Employee;

@Controller
public class EmployeeController {

        @GetMapping("/data")
        public String showData(Model model) {
                //System.out.println(model.getClass().getName());

                List<Employee> emps = List.of(
                                        new Employee(10, "A", 3.3),
                                        new Employee(11, "B", 4.3),
                                        new Employee(12, "C", 5.6),
                                        new Employee(13, "D", 5.8)
                                );

                model.addAttribute("employees", emps);
```

```
                return "EmpData";
        }
}

3. UI: EmpData.jsp (NO UI DESIGN)
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<html>
<body>
<h3> WELCOME TO EMPLOYEE DATA PAGE!! </h3>
 <%-- ${employees}  --%>

<table border="1">
        <tr>
                <th>ID</th>
                <th>NAME</th>
                <th>SALARY</th>
        </tr>

        <c:forEach items="${employees}" var="ob">
                <tr>
                        <td>${ob.eid}</td>
                        <td>${ob.ename}</td>
                        <td>${ob.esal}</td>
                </tr>
        </c:forEach>

</table>

</body>
</html>


3. UI: EmpData.jsp (WITH UI DESIGN)
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>
<head>
<link rel="stylesheet"
 href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.m
in.css" />
</head>
<body>
        <div class="container">
                <h3>WELCOME TO EMPLOYEE DATA PAGE!!</h3>
                <%-- ${employees}  --%>

                <table class="table table-hover">
                        <tr class="bg-primary text-white">
                                <th>ID</th>
                                <th>NAME</th>
                                <th>SALARY</th>
                                <th>OPERATIONS</th>
                        </tr>

                        <c:forEach items="${employees}" var="ob">
```

```
                                    <tr>
                                            <td>${ob.eid}</td>
                                            <td>${ob.ename}</td>
                                            <td>${ob.esal}</td>
                                            <td>
                                                    <a href="#" class="btn
btn-danger">DELETE</a> |
                                                    <a href="#" class="btn
btn-success">EDIT</a>
                                            </td>
                                    </tr>
                            </c:forEach>

                    </table>

            </div>
</body>
</html>
```
