Date : 08/03/2021
Spring Boot 9AM
Mr. RAGHU

Code:
https://www.mediafire.com/file/92tx87hckkly1i0/SpringBoot2WebMvcMySQlCrud08032021.zip/file

---Task-----------------------
a. Define Custom page for 500 (Unhandled Exceptions)
b. Goto Log4J Videos
https://www.youtube.com/c/NareshIT/search?query=log4j%20raghu
Cache:
https://www.youtube.com/c/NareshIT/search?query=redis%20cache%20raghu

---

Mini Project Stages

Stage#1 Register Employee [Done]
Stage#2 Display Data [Done]
Stage#3 Delete By Id [Done]
Stage#4 Edit Page and Update Data [DONE]
Stage#5 UI BootStrap Design [done]
Stege#6 UI-JQuery Validation [done]
Stage#7 AJAX validation [DONE]
Stage#8 Error Page Handling [done]
Stage#9 Exception Handling [done]
Stage#10 Log Implementation

Stage#7 AJAX Validation

EmailId pattern:-

_____ @ _____ . _____

[a-zA-Z0-9\.\-_] \@ [a-z] \. [a-z\.]{2,10}
at least one char
means + symbols

SQL:
SELECT COUNT(EMAIL) FROM EMPLOYEE WHERE EMAIL=?

a. Repository

```
@Query("SELECT COUNT(empMail) FROM Employee WHERE empMail=:empMail")
Integer getEmployeeEmailCount(String empMail);
```

b. Service Interface

```
boolean isEmployeeEmailExist(String empMail);
```

c. SeviceImpl

```
public boolean isEmployeeEmailExist(String empMail) {
        /*
        Integer count = repo.getEmployeeEmailCount(empMail);
```

```
                boolean isExist = count > 0 ? true:false;
                return isExist;
                */
                return repo.getEmployeeEmailCount(empMail)>0;
 }
```

d. Controller

```
        @GetMapping("/validateMail")
        public @ResponseBody String validateEmail(
                        @RequestParam("email")String empMail
                        )
        {
                String message = "";

                if(service.isEmployeeEmailExist(empMail)) {
                        message = empMail+", already exist";
                }

                return message;
        }
```

e. At Register page

```
--syntax---
$.ajax({
  url : '/wheretoMakeRequest',
  data : { key:val what to send input to method},
  success:function(paramName) {
    //what logic should we execute on controller response
  }
})
-------------
        $.ajax({
                url : 'validateMail',
                data: { 'email' : val},
                success:function(resTxt) {
                        if(resTxt!='') {
                                $("#empMailError").show();
                                $("#empMailError").html(resTxt);
                                $("#empMailError").css('color','red');
                                empMailError = false;
                        } else {
                                $("#empMailError").hide();
                                empMailError = true;
                        }
                }
       });
--------------------------
@ResponseBody : It indicates return type is not a viewName, this is
                final response need to be sent back to same page.


--Controller#method-----
@GetMapping("/show")
String showPage() {
```

```
  return "HomePage"; ==> HomePage.html
}

@GetMapping("/validate")
@ResponseBody String showMsg() {
  //This is not viewName
  return "Invalid Data! Try Again";
}
```

Stage#8 Error Page Handling

*) Spring Web MVC has provided one pre-defined error contoller
Link: https://www.youtube.com/c/NareshIT/search?query=error%20raghu

ie BasicErrorController, which handled all errors and exceptions
and gives finally white label error pages.

*) Here, I want to replace 'white label error pages' with our custom
   error pages, for that follow below folder system

```
--For thymeleaf---
ProjectFolder
 |--src/main/resources
     |--templates
         |--error
             |--404.html
             |--405.html

--For JSP---
ProjectFolder
 |-src
    |-main
       |-webapp
          |-WEB-INF
             |-mypages
                  |--error
                      |--404.jsp
                      |--405.jsp
```

```
---404.html-----------
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css" />
<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
<script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd
/popper.min.js"></script>
<script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstra
p.min.js"></script>
</head>
<body>
```

```
        <div class="container">

                <div class="card">
                        <div class="card-header bg-primary text-white
text-center">
                                <h2>EMPLOYEE SERVICE NOT EXIST</h2>
                        </div>
                        <div class="card-body">
                                <img th:src="@{/images/nit.png}"
width="160" height="50" />
                                <b>HYDERABAD(M): 040-23746666 (M): 9000
994 007 / 8</b>
                                <br/> <br/>
                                Goto <a th:href="@{/employee/all}">
                                <img th:src="@{/images/home.png}"
width="20" height="25"/>
                                </a>
                        </div>

                </div>

        </div>

</body>
</html>

---405.html-----------
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css" />
<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
<script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd
/popper.min.js"></script>
<script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstra
p.min.js"></script>
</head>
<body>
        <div class="container">

                <div class="card">
                        <div class="card-header bg-primary text-white
text-center">
                                <h2>WRONG ACCESS TYPE!!</h2>
                        </div>
                        <div class="card-body">
                                <img th:src="@{/images/nit.png}"
width="160" height="50" />
                                <b>HYDERABAD(M): 040-23746666 (M): 9000
994 007 / 8</b>
                                <br/> <br/>
```

```
                                Goto <a th:href="@{/employee/all}">
                                <img th:src="@{/images/home.png}"
width="20" height="25"/>
                                </a>
                        </div>
                </div>
        </div>

</body>
</html>
```

Stage#9 Exception Handling

#1. Define one custom exception

```
package in.nareshit.raghu.exception;

public class EmployeeNotFoundException
        extends RuntimeException
{

        private static final long serialVersionUID = 1L;

        public EmployeeNotFoundException() {
                super();
        }
        public EmployeeNotFoundException(String message) {
                super(message);
        }
}
```

#2. Throw exception at service if not exist

```
     public Employee getOneEmployee(Integer id) {
                /*Optional<Employee> opt = repo.findById(id);
                if(opt.isEmpty()) {
                        throw new EmployeeNotFoundException("Employee
'"+id+"' Not exist");
                } else {
                        return opt.get();
                }*/
                return repo.findById(id)
                        .orElseThrow(
                                ()-> new EmployeeNotFoundException(
                                        "Employee '"+id+"' Not exist")
                                );
        }
```

#3. handle exception at controller and return custom error msg to UI

```
        @GetMapping("/delete")
        public String deleteEmp(
                        @RequestParam("id") Integer empId,
```

```
                        Model model
                        )
        {
                String page = null;
                try {
                        service.deleteEmployee(empId);
                        model.addAttribute("msg", "Employee '"+empId+"'
Deleted");

                } catch (EmployeeNotFoundException enfe) {
                        page = "EmployeeData";
                        model.addAttribute("msg", enfe.getMessage());
                } catch (Exception e) {
                        e.printStackTrace();
                }
                page="EmployeeData";
                //fetch latest data after delete
                getUiEmployeeTableData(model);


                return page;
        }
```