WAY2LEARN

LAB- Spring Mvc - Exception Handling

In this lab you will be understanding how to handle Exceptions in Spring mvc

STEP -1

In this step, you will be working on **08-mvc-exceptions-start** project

1) In com.way2learnonline.exceptions package Observe that we have defined two exception classes **CourseNotFoundException** and **InvalidRequestException**

2) Open CourseController.java and observe that viewIndividualCourse method is throwing 2 exceptions.

3)Now Deploy the application and visit home page. In Home Page, in the top menu, click on Course-> Hadoop link. You will see the Hadoop course details. Now the URL should be as below : http://localhost:8080/08-mvc-exceptions-start/viewIndividualCourse.htm?courseId=hadoop

Now change the courseId to xyz and give a request. You should see Exception stack trace as we have not handled exceptions yet.

4) GOTO mvc-config.xml and configure SimpleMappingExceptionResolver as shown below :

```
<bean id="simpleMappingExceptionResolver"
      class="org.springframework.web.servlet.handler.SimpleMappingExceptionResolver">
      <property name="exceptionMappings">
            <map>
                  <entry key="com.way2learnonline.exceptions.CourseNotFoundException"
                        value="coursenotfound" />
                  <entry key="com.way2learnonline.exceptions.InvalidRequestException"
                        value="invalidrequest" />
            </map>
      </property>
      <property name="exceptionAttribute" value="ex" />
      <property name="defaultErrorView" value="defaultErrorPage" />
</bean>
```

In the configuration Observe how Exceptions are mapped to logical view names.

Exception Attribute name is configured as ex. So, We can use "ex" to refer to exception Object.

Open tiles.xml and observe that we have already configured tile definitions for **coursenotfound** and **invalidrequest**

Open `WEB-INF/jsp/application/coursenotfound.jsp` **and** `/WEB-INF/jsp/application/invalidrequest.jsp"` **and observe the code .**

Now deploy the application and try to view course details for courseid =xyz.

You should observe that coursenotfound.jsp is shown.

Try giving request to
localhost:8080/08-mvc-exceptions-start/viewIndividualCourse.htm?courseId=hadoop&**showerror=true**
You should Observe that invalidrequest.jsp is shown.

So, in this step, you understood how to use SimpleMappingExceptionResolver.

5) **Handling Exceptions using @ResponseStatus**

Goto mvc-config.xml and comment the bean definition of SimpleMappingExceptionResolver.

open CourseNotFoundException.java and annotate it with

```
@ResponseStatus(value=HttpStatus.NOT_FOUND,reason="Course Not Found!!!!")
```

Now deploy the application and give request to
http://localhost:8080/08-mvc-exceptions-start/viewIndividualCourse.htm?courseId=xyz

Now you should see 404 status code with message "Course Not Found !!" instead of 500 error code.

Can you tell which HandlerExceptionResolver is mapping the exception to status code?

6) Handling Exceptions with @ExceptionHandler

Open CourseNotFoundException.java and remove the @ResponseStatus on top of it.

Open CourseController and write following exception handler methods :

```
@ExceptionHandler(value={CourseNotFoundException.class})
      public ModelAndView handleCourseNotFound(){

            ModelAndView mav= new ModelAndView("coursenotfound");
            mav.addObject("someData", "If you are seeing this, then this exception is
mapped by @ExceptionHandler(value={CourseNotFoundException.class})");
```

```
            return mav;
    }


    @ExceptionHandler(value={InvalidRequestException.class})
    public ModelAndView handleInvalidRequest(){

            ModelAndView mav= new ModelAndView("invalidrequest");
            mav.addObject("someData", "If you are seeing this, then this exception is 
mapped by @ExceptionHandler(value={InvalidRequestException.class})");

            return mav;
    }
```

Now Deploy the application and give a request to
http://localhost:8080/08-mvc-exceptions-start/viewIndividualCourse.htm?courseId=xyz

and

http://localhost:8080/08-mvc-exceptions-start/viewIndividualCourse.htm?courseId=hadoop&showerror=true

Did you observe the corresponding error pages shown?

Can you tell which HandlerExceptionResolver is used ?

6) Handling Exceptions using @ControllerAdvice

Now remove both the @ExceptionHandler annotated methods and write them in a class with name MyControllerAdvice and annotated it with @ControllerAdvice .

Now deploy the application and give request to same urls.

Observe that same error pages are shown.

Congratulations !! You know how to handle Exceptions in Spring MVC