We have used Exception Handling code into java code. If we have one exception handling code and same exception we have in many files then many time we have to write that code so to remove this problem we create other file for that code and extends in those classes where we have that exception but here is also one problem this is boiler plate code and not related to business logic so we should create Aspect for that and use in application as aspect and to achive this we using here **@ControllerAdvice.**

**Like**

@ControllerAdvice

**public** **class** ExceptionHandlingController {

@ExceptionHandler(StudentNotFoundException.**class**)

**public** ModelAndView handleStudentNotFoundException(StudentNotFoundException ex) {

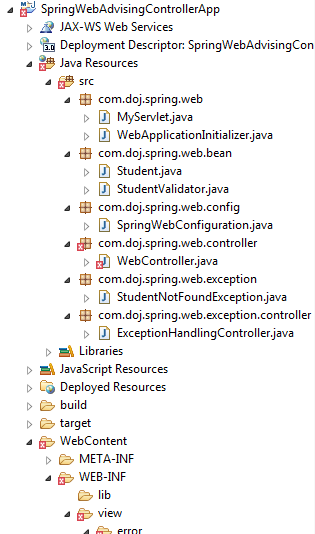
Map<String, StudentNotFoundException> model = **new** HashMap<String, StudentNotFoundException>();

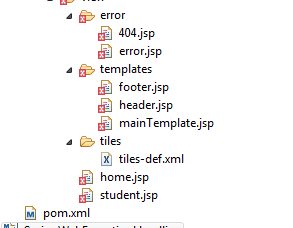
model.put("exception", ex);

**return** **new** ModelAndView("student.error", model);

}

**Now we do not have any code in any file related to this exception but this is executed because we used @ControllerAdvice. Whenever Exception occurred at that time this advice is executed.**





**package** com.doj.spring.web;

**import** javax.servlet.MultipartConfigElement;

**import** javax.servlet.ServletRegistration.Dynamic;

**import** org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

**import** com.doj.spring.web.config.SpringWebConfiguration;

//this file is equivalent to web.xml

**public** **class** WebApplicationInitializer **extends** AbstractAnnotationConfigDispatcherServletInitializer{

//Configuration for non web components like services, daos, repos, etc.

@Override

**protected** Class<?>[] getRootConfigClasses() {

**return** **null**;

}

//Specifying Spring MVC configuration class "SpringWebConfiguration.class" it equivalent to \*-servlet.xml file

@Override

**protected** Class<?>[] getServletConfigClasses() {

**return** **new** Class<?>[]{SpringWebConfiguration.**class**};

}

//Mapping dispatcher server to "/" i.e. Servlet Mapping in the web.xml

@Override

**protected** String[] getServletMappings() {

**return** **new** String[]{"/"};

}

/\*@Override

public void onStartup(ServletContext servletContext)

throws ServletException {

Dynamic myServlet = servletContext.addServlet("myServlet", MyServlet.class);

myServlet.addMapping("/custom-servlet/");

}\*/

@Override

**protected** **void** customizeRegistration(Dynamic registration) {

registration.setMultipartConfig(

**new** MultipartConfigElement("E:\\Shiksha\\Programming\\DOJWorkspace", 204834l, 4937353l,0));

}

}

**package** com.doj.spring.web.bean;

**public** **class** Student {

String fname;

String lname;

String address;

String course;

String age;

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return** lname;

}

**public** **void** setLname(String lname) {

**this**.lname = lname;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getCourse() {

**return** course;

}

**public** **void** setCourse(String course) {

**this**.course = course;

}

**public** String getAge() {

**return** age;

}

**public** **void** setAge(String age) {

**this**.age = age;

}

}

**package** com.doj.spring.web.bean;

**import** org.springframework.validation.Errors;

**import** org.springframework.validation.ValidationUtils;

**import** org.springframework.validation.Validator;

**public** **class** StudentValidator **implements** Validator{

@Override

**public** **boolean** supports(Class<?> clazz) {

**return** Student.**class**.isAssignableFrom(clazz);

}

@Override

**public** **void** validate(Object object, Errors errors) {

ValidationUtils.rejectIfEmpty(errors, "fname", "required", "Enter First name");

ValidationUtils.rejectIfEmpty(errors, "lname", "required", "Enter Last name");

ValidationUtils.rejectIfEmpty(errors, "address", "required", "Enter address");

ValidationUtils.rejectIfEmpty(errors, "course", "required", "Enter course name");

ValidationUtils.rejectIfEmpty(errors, "age", "required", "Enter Age");

}

}

**package** com.doj.spring.web.config;

**import** java.io.IOException;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.web.multipart.MultipartResolver;

**import** org.springframework.web.multipart.support.StandardServletMultipartResolver;

**import** org.springframework.web.servlet.ViewResolver;

**import** org.springframework.web.servlet.config.annotation.DefaultServletHandlerConfigurer;

**import** org.springframework.web.servlet.config.annotation.EnableWebMvc;

**import** org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

**import** org.springframework.web.servlet.view.tiles3.TilesConfigurer;

**import** org.springframework.web.servlet.view.tiles3.TilesViewResolver;

@Configuration

@EnableWebMvc

@ComponentScan(basePackages={"com.doj.spring.web.controller","com.doj.spring.web.exception.controller"})

**public** **class** SpringWebConfiguration **extends** WebMvcConfigurerAdapter{

//Apache tiles

//Configuring and creating bean for view resolver

/\*@Bean

public ViewResolver viewResolver() {

InternalResourceViewResolver resolver = new InternalResourceViewResolver();

resolver.setPrefix("/WEB-INF/view/");

resolver.setSuffix(".jsp");

resolver.setExposeContextBeansAsAttributes(true);

return resolver;

}\*/

@Bean

**public** ViewResolver viewResolver() {

**return** **new** TilesViewResolver();

}

@Bean

**public** TilesConfigurer tilesConfigurer() {

TilesConfigurer tiles = **new** TilesConfigurer();

tiles.setDefinitions(**new** String[] {

"/WEB-INF/view/tiles/tiles-def.xml"

});

tiles.setCheckRefresh(**true**);

**return** tiles;

}

//Configure for default static content handling

@Override

**public** **void** configureDefaultServletHandling(DefaultServletHandlerConfigurer configurer) {

configurer.enable();

}

@Bean

**public** MultipartResolver multipartResolver() **throws** IOException {

**return** **new** StandardServletMultipartResolver();

}

}

**package** com.doj.spring.web.controller;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.validation.Valid;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.ModelMap;

**import** org.springframework.validation.Errors;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.bind.annotation.RequestPart;

**import** org.springframework.web.multipart.MultipartFile;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.doj.spring.web.bean.Student;

**import** com.doj.spring.web.exception.StudentNotFoundException;

@Controller

**public** **class** WebController{

**static** Map<String, Student> map = **new** HashMap<String, Student>();

//multiple mapping with one remote method

@RequestMapping(value={"/", "/index","/home","/welcome"})

**public** String home(){

Student student = **new** Student();

student.setFname("Dinesh");

student.setLname("Rajput");

student.setAge("29");

student.setAddress("Noida");

student.setCourse("MS");

map.put("1000", student);

**return** "home";

}

//With Model and Model Name to View Resolver

@RequestMapping("/indexc")

**public** ModelAndView welcome(){

Map<String, String> model = **new** HashMap<>();

model.put("name", "Sumit");

**return** **new** ModelAndView("home","model", model);

}

//We are using Spring ModelMap for return the model value

@RequestMapping("/hello")

**public** String index(ModelMap model){

model.put("name", "Sumit");

**return** "home";

}

//We are using Spring ModelMap and fetching request parameter here

@RequestMapping("/doj")

**public** String hello(ModelMap model, HttpServletRequest request){

String name = request.getParameter("name");

model.put("name", name);

**return** "home";

}

//We are using Spring ModelMap and Mapping the attribte with request param annotation with attributes whatever is ur requirement

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam(defaultValue = "DOJ Students", required=**true**, value="fname") String name,

@RequestParam(required = **false**, value="lname") String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

@RequestMapping("/doj-student-{fname}-{lname}")

**public** String dojStudent(ModelMap model, @PathVariable(value="fname") String name, @PathVariable(value="lname") String sname,

@RequestParam String address){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

@RequestMapping(value="/doj-student", method=RequestMethod.GET)

**public** String getDojStudent(ModelMap model, Student student){

String name = **null**;

**if**(student.getFname() != **null**){

name = student.getFname();

}

**if**(student.getLname() != **null**){

name = name + " " +student.getLname();

}

**if**(student.getAddress() != **null**){

name = name+" "+student.getAddress();

}

**if**(student.getCourse() != **null**){

name = name+" "+student.getCourse();

}

model.put("name", name);

**return** "home";

}

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

**public** String student(){

**return** "student";

}

@RequestMapping(value="/dojstudent", method=RequestMethod.POST)

**public** String getStudent(ModelMap model, @Valid Student student, Errors errors, @RequestPart("profilePicture") MultipartFile profilePicture){

**if**(errors.hasErrors()){

System.out.println("Errors "+errors.getErrorCount());

**return** "student";

}

**try** {

profilePicture.transferTo(**new** File("E:\\Shiksha\\Programming\\DOJWorkspace\\"+profilePicture.getOriginalFilename()));

} **catch** (IllegalStateException | IOException e) {

e.printStackTrace();

}

String name = **null**;

**if**(student.getFname() != **null**){

name = student.getFname();

}

**if**(student.getLname() != **null**){

name = name + " " +student.getLname();

}

**if**(student.getAddress() != **null**){

name = name+" "+student.getAddress();

}

**if**(student.getCourse() != **null**){

name = name+" "+student.getCourse();

}

model.put("name", name);

**return** "home";

}

@RequestMapping("/doj-student-{rollNumber}")

**public** String dojStudentByRollNumber(ModelMap model, @PathVariable(value="rollNumber") String rollNumber){

Student student = map.get(rollNumber);

**if** (student == **null**) {

**throw** **new** StudentNotFoundException("Student Not Found", "MYAPPERROR-404");

}

String name = student.getFname()+" "+student.getLname()+" "+student.getAddress()+" "+student.getCourse();

model.put("name", name);

**return** "home";

}

}

**package** com.doj.spring.web.exception;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value=HttpStatus.NOT\_FOUND, reason="Student Not Found")

**public** **class** StudentNotFoundException **extends** RuntimeException{

/\*\*

\*

\*/

**private** **static** **final** **long** serialVersionUID = -2581975292273282583L;

String errorMessage;

String errorCode;

**public** StudentNotFoundException(String errorMessage, String errorCode) {

**super**();

**this**.errorMessage = errorMessage;

**this**.errorCode = errorCode;

}

**public** String getErrorMessage() {

**return** errorMessage;

}

**public** **void** setErrorMessage(String errorMessage) {

**this**.errorMessage = errorMessage;

}

**public** String getErrorCode() {

**return** errorCode;

}

**public** **void** setErrorCode(String errorCode) {

**this**.errorCode = errorCode;

}

}

**package** com.doj.spring.web.exception.controller;

**import** java.util.HashMap;

**import** java.util.Locale;

**import** java.util.Map;

**import** org.springframework.web.bind.WebDataBinder;

**import** org.springframework.web.bind.annotation.ControllerAdvice;

**import** org.springframework.web.bind.annotation.ExceptionHandler;

**import** org.springframework.web.bind.annotation.InitBinder;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.doj.spring.web.bean.Student;

**import** com.doj.spring.web.bean.StudentValidator;

**import** com.doj.spring.web.exception.StudentNotFoundException;

@ControllerAdvice

**public** **class** ExceptionHandlingController {

@ExceptionHandler(StudentNotFoundException.**class**)

**public** ModelAndView handleStudentNotFoundException(StudentNotFoundException ex) {

Map<String, StudentNotFoundException> model = **new** HashMap<String, StudentNotFoundException>();

model.put("exception", ex);

**return** **new** ModelAndView("student.error", model);

}

@ExceptionHandler(Exception.**class**)

**public** ModelAndView handleException(Exception ex) {

Map<String, Exception> model = **new** HashMap<String, Exception>();

model.put("exception", ex);

**return** **new** ModelAndView("error", model);

}

@InitBinder

**public** **void** initBinder(WebDataBinder binder) {

binder.setDisallowedFields("fname");

binder.setValidator(**new** StudentValidator());

}

@ModelAttribute

**public** Student register(Locale currentLocale){

Student student = **new** Student();

student.setAddress(currentLocale.getDisplayName());

**return** student;

}

**}**

**404.jsp**

**<h1>${exception.errorCode} : ${exception.errorMessage}!!!</h1>**

**Error.jsp**

<**h1>${exception}!!!</h1>**

**Footer.jsp**

**<h1>&copy; DOJ SOFTWARE CONSULTANTS</h1>**

**Header.jsp**

**<h1>DOJ STUDENT FORM</h1>**

**mainTemplate.xml**

<!DOCTYPE h1 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<%@taglib uri=*"http://tiles.apache.org/tags-tiles"* prefix=*"tiles"* %>

<html>

<head>

<title><tiles:insertAttribute name=*"title"*/></title>

<style>

*.error* {

color: *red*; font-weight: *bold*;

}

</style>

</head>

<body style="text-align: *center*;">

<div id=*"header"* style="height: *10%*;background-color: *gray*;"><tiles:insertAttribute name=*"header"*/></div>

<div id=*"body"* style="height: *70%*; background-color: *aqua*;padding-top: *40px*;"><tiles:insertAttribute name=*"body"*/></div>

<div id=*"footer"* style="height: *10%*;background-color: *gray*;"><tiles:insertAttribute name=*"footer"*/></div>

</body>

</html>

**Tiles-def.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE tiles-definitions PUBLIC

"-//Apache Software Foundation//DTD Tiles Configuration 2.0//EN"

"http://tiles.apache.org/dtds/tiles-config\_2\_0.dtd">

<tiles-definitions>

<definition name=*"mainTemplate"* template=*"/WEB-INF/view/templates/mainTemplate.jsp"*>

<put-attribute name=*"title"* value=*""*></put-attribute>

<put-attribute name=*"header"* value=*"/WEB-INF/view/templates/header.jsp"*></put-attribute>

<put-attribute name=*"body"* value=*""*></put-attribute>

<put-attribute name=*"footer"* value=*"/WEB-INF/view/templates/footer.jsp"*></put-attribute>

</definition>

<definition name=*"student"* extends=*"mainTemplate"*>

<put-attribute name=*"title"* value=*"STUDENT FORM"*></put-attribute>

<put-attribute name=*"body"* value=*"/WEB-INF/view/student.jsp"*></put-attribute>

</definition>

<definition name=*"home"* extends=*"mainTemplate"*>

<put-attribute name=*"body"* value=*"/WEB-INF/view/home.jsp"*></put-attribute>

<put-attribute name=*"title"* value=*"WELCOME DOJ CLASS"*></put-attribute>

</definition>

<definition name=*"student.error"* extends=*"mainTemplate"*>

<put-attribute name=*"body"* value=*"/WEB-INF/view/error/404.jsp"*></put-attribute>

<put-attribute name=*"title"* value=*"OOPS Something went wrong | WELCOME DOJ CLASS"*></put-attribute>

</definition>

<definition name=*"error"* extends=*"mainTemplate"*>

<put-attribute name=*"body"* value=*"/WEB-INF/view/error/error.jsp"*></put-attribute>

<put-attribute name=*"title"* value=*"OOPS Something went wrong | WELCOME DOJ CLASS"*></put-attribute>

</definition>

</tiles-definitions>

Home.jsp

<h1>${name} Welcome to DOJ Classes for Spring MVC!!!</h1>

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<form:form action=*"dojstudent"* commandName=*"student"* enctype=*"multipart/form-data"*>

<table style="text-align: *center*; position: *absolute*;padding-left: *500px*;">

<tr>

<td>First Name : </td>

<td><input name=*"fname"* type=*"text"*></td>

<td><form:errors path=*"fname"* cssClass=*"error"*/></td>

</tr>

<tr>

<td>Last Name : </td>

<td><input name=*"lname"* type=*"text"*></td>

<td><form:errors path=*"lname"* cssClass=*"error"*/></td>

</tr>

<tr>

<td>Address : </td>

<td><input name=*"address"* type=*"text"* value=*"*${student.address}*"*></td>

<td><form:errors path=*"address"* cssClass=*"error"*/></td>

</tr>

<tr>

<td>Course : </td>

<td><input name=*"course"* type=*"text"*></td>

<td><form:errors path=*"course"* cssClass=*"error"*/></td>

</tr>

<tr>

<td>Age : </td>

<td><input name=*"age"* type=*"text"*></td>

<td><form:errors path=*"age"* cssClass=*"error"*/></td>

</tr>

<tr>

<td>Profile Picture : </td>

<td><input type=*"file"* name=*"profilePicture"* accept=*"image/jpeg,image/png,image/gif"* /></td>

<td></td>

</tr>

<tr>

<td></td>

<td><input type=*"Submit"* name=*"Submit"* value=*"Submit"*></td>

</tr>

</table>

</form:form>