



When we have exception in our application then our application does not run and then through exception that has standard message. Now we are going to make custom Exception and will through according our requirement.

If we search student number that does not exist then will will through message student not found.

**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;

}

}

This is code where we are throughing exception

@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";

}

It will print simple message with error code if we want to print our jsp page with error code and message then use ExceptionHandler. It will bind to it when exception occurred.

@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);

}

**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;

**import** javax.validation.constraints.Max;

**import** javax.validation.constraints.Min;

**import** javax.validation.constraints.NotNull;

**import** javax.validation.constraints.Size;

**import** org.hibernate.validator.constraints.NotEmpty;

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

@NotNull

@NotEmpty(message="Your first name is mandatory")

@Size(max=20, min=5, message="Your first name must be between 5 to 20 characters")

String fname;

@NotNull

@NotEmpty(message="Your last name is mandatory")

@Size(max=20, min=5, message="Your last name must be between 5 to 20 characters")

String lname;

@NotNull

@NotEmpty(message="Your address is mandatory")

@Size(max=50, min=10, message="Your address must be between 10 to 50 characters")

String address;

@NotNull

@NotEmpty(message="Your course is mandatory")

@Size(max=10, min=3, message="Your course must be between 3 to 10 characters")

String course;

@NotNull

@NotEmpty(message="Your age is mandatory")

@Max(value=50, message="Your age should be less than or equal to 50")

@Min(value=5, message="Your age should be greater than or equal to 5")

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.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("com.doj.spring.web.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.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("com.doj.spring.web.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();

}

}

404.jsp

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

**Footer.jsp**

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

**Header.jsp**

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

**mainTemplate.jsp**

<!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>

</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"*></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>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.DOJSpring</groupId>

<artifactId>SpringWebExceptionHandling</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>SpringWebExceptionHandling</name>

<description>SpringWebExceptionHandling is about Exception Handling in ths Spring MVC</description>

<properties>

<spring.version>4.0.0.RELEASE</spring.version>

<aspect.version>1.6.11</aspect.version>

<tiles.version>3.0.5</tiles.version>

<jstl.version>1.2</jstl.version>

</properties>

<dependencies>

<!-- Spring Core dependency -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- Spring Core dependency -->

<!-- Spring AOP dependency -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${aspect.version}</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>${aspect.version}</version>

</dependency>

<!-- Spring AOP dependency -->

<!-- Spring MVC dependency -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- Spring MVC dependency -->

<!-- Tiles dependency -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-core</artifactId>

<version>${tiles.version}</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-jsp</artifactId>

<version>${tiles.version}</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-api</artifactId>

<version>${tiles.version}</version>

</dependency>

<!-- Tiles dependency -->

<!-- JSTL dependency -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>${jstl.version}</version>

</dependency>

<!-- JSTL dependency -->

<dependency>

<groupId>javax.validation</groupId>

<artifactId>validation-api</artifactId>

<version>1.0.0.GA</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>4.2.0.Final</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>1.7</source>

<target>1.7</target>

</configuration>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>2.3</version>

<configuration>

<warSourceDirectory>WebContent</warSourceDirectory>

<failOnMissingWebXml>false</failOnMissingWebXml>

</configuration>

</plugin>

</plugins>

</build>

</project>