We are using in built validation(@NotNull, @NotEmpty) here not custom. All validation will work if we will use @valid annotation before that bean where we are going to use validation like here

We are using validations in student bean (i.e. @NotNull

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

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

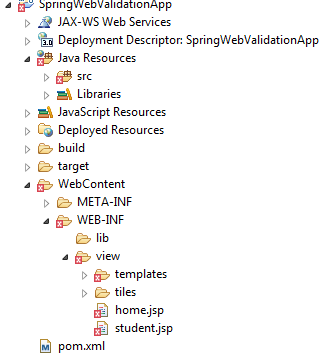
String fname;

)

For this we have to use @valid annotation then above validation will work (like @RequestMapping(value="/dojstudent", method=RequestMethod.POST)

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

If any validation fail then form will not submit and errors assign to Errors class from this class we can extract information about error.



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

**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[]{"/"};

}

}

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

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

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

**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();

}

}

**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.servlet.ModelAndView;

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

@Controller

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

//multiple mapping with one remote method

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

**public** String home(){

**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){

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

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

**return** "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";

}

}