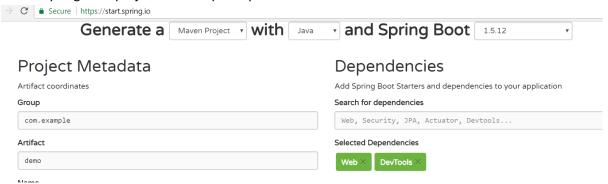
## **Spring boot with Request Validation**

## Check this out on github >> <a href="https://github.com/fishyimm/spring-boot-request-validation">https://github.com/fishyimm/spring-boot-request-validation</a>

1. Create spring boot project with simple dependencies.



2. You can add swagger dependencies to test your project, add this in your pom.xml

```
<!-- swagger -->

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger-ui</artifactId>

<version>2.7.0</version>

</dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version2.7.0</p>
</dependency>

</dependency>

</dependency>

</dependency>

</dependency>

</dependency>

</dependency>
```

 Create simple configuration for swagger as below: https://github.com/fishyimm/spring-boot-request

validation/blob/master/src/main/java/com/example/demo/config/

🗏 💈 🦻 🤻 🗖 🗎 🛮 SwaggerConfig.java 🛭 package com.example.demo.config; √ № demo [boot] [devtools] 3. import org.springframework.context.annotation.Bean; > # com.example.demo > # com.example.demo.annotations 14 @Configuration > # com.example.demo.bean 15 @EnableSwagger2 w the com.example.demo.config 16 public class SwaggerConfig { > 🛂 SwaggerConfig.java private static final String GROUP\_NAME = "TestApi"; > 🛂 WebConfig.java 19 \[
\begin{align\*}
\begin{align\*} 20⊜ > 🔠 com.example.demo.exceptions 21 public Docket productApi() { > # com.example.demo.res 22 return new Docket(DocumentationType.SWAGGER\_2) > # com.example.demo.validators 23 .groupName(GROUP\_NAME) 24 .select() static 25 .apis(RequestHandlerSelectors.any()) .paths(PathSelectors.any()) beta templates
 beta templates
 complete templates
 26 27 .build().apiInfo(metaData()) application.properties 28 .useDefaultResponseMessages(false); > # src/test/iava 29 > A JRE System Library [JavaSE-1.8] 30 > 🛋 Maven Dependencies 31⊜ private ApiInfo metaData() { > 🗁 bin 32 return new ApiInfoBuilder() .title("test api") 33 > 😂 src .description("description") > 🗁 target 35 .license("test license") mvnw 36 .version("0.1") mvnw.cmd .build(); 37 pom.xml 38 } > 🗁 Servers 39 40 } E S S WebConfig.java ⊠ ☐ Package Explorer ☐ package com.example.demo.config; 3**⊕import** java.util.List;[ > # com.example.demo # com.example.demo.annotations # com.example.demo.bean # com.example.demo.config @EnableWebMvc
public class WebConfig extends WebMvcConfigurerAdapter { > La SwaggerConfig.java public void addResourceHandlers(ResourceHandlerRegistry registry) {
 registry.addResourceHandler("swagger-ui.html").addResourceLocations("classpath:/META-INF/resources/");
 registry.addResourceHandler("/webjars/\*\*").addResourceLocations("classpath:/META-INF/resources/webjars/");
} # com.example.demo.controller # com.example.demo.exceptions # com.example.demo.res # com.example.demo.validators public void configureMessageConverters(List<a href="http://distable.com/bullet/">http://distable.com/bullet/</a> converters) {
 final MappingJackson2HttpMessageConverter converter = new MappingJackson2HttpMe
 final ObjectMapper objectMapper = new ObjectMapper();
 objectMapper.setSerializationInclusion(JsonInclude.Include.NON\_NULL);
 objectMapper.setSerializationInclusion(JsonInclude.Include.NON\_EMPTY);
 objectMapper.configure(SerializationFeature.FAIL\_ON\_EMPTY\_BEANS, false);
} static templates = new MappingJackson2HttpMessageConverter(); application.properties > @ src/test/java
> m JRE System Library [JavaSE-1.8] Maven Dependencies converter.setObjectMapper(objectMapper); > 🗁 bin converters.add(converter);
super.configureMessageConverters(converters); > 😂 src > 🗁 target mvnw mvnw.cmd

4. Create simple bean and simple controller for person.

https://github.com/fishyimm/spring-boot-request-

validation/blob/master/src/main/java/com/example/demo/bean/Person.java

https://github.com/fishyimm/spring-boot-request-

validation/blob/master/src/main/java/com/example/demo/controller/PersonController.java

```
Person,java 22  PersonController,java

1 package com.example.demo.bean;
♣ Package Explorer ≅

✓ 

demo [boot] [det

✓ 

src/main/java

src/
                                                                                                                                               > # com.example.demo
           > # com.example.demo.annotations
> # com.example.demo.annotations
> # com.example.demo.bean
> D Person.java
> # Com.example.demo.config
# com.example.demo.controller
> # com.example.demo.exceptions
# com.example.demo.exceptions
                                                                                                                                                                          @NotNull
private int age;
public String getFirstname() {
    return firstname;
                                                                                                                                                15° 16 17 18° 19 20 21° 22 23 24° 25 26 27° 28 29 30° 31 32 33° 34 35 36 37
                 > # com.example.demo.res
                      # com.example.demo.validators

✓ 

Src/main/resources

                                                                                                                                                                               public void setFirstname(String firstname) {
    this.firstname = firstname;

Strict Manning Strict

static
templates
application, properties
Strictest/java
M.RE System Library [JavaSE-1.8]

Mayen Dependencies
                                                                                                                                                                              public String getLastname() {
   return lastname;
                                                                                                                                                                               public void setLastname(String lastname) {
    this.lastname = lastname;

    bin

                                                                                                                                                                               public int getAge() {
    return age;
             ₽ src
              target
                                                                                                                                                                                public void setAge(int age) {
   this.age = age;
                                                                                                                                                                               // Governide
public String toString() {
   return "Person [firstname=" + firstname + ", lastname=" + lastname + ", age=" + age + "]";
Package Explorer ⋈
                                                                                🗏 💲 💆 🗖 🖟 Person.java 🖟 PersonController.java 🛭
                                                                                                                                                                            1 package com.example.demo.controller;

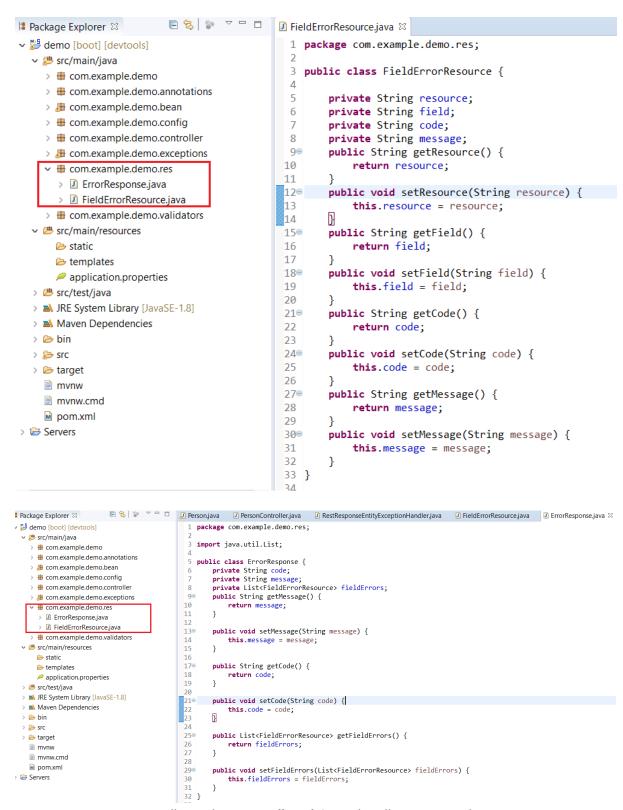
√ № demo [boot] [devtools]

                                                                                                                                                               1 package com.example.demo.controll
2
3* import javax.validation.Valid;
13
14
16
17* @RestController
15 public class PersonController {
16
17* @RequestMapping(value = "/per
18 public String person(@Valid @
19 return "person";
20 }
21
22* @RequestMapping(value = "/per
23 public String person(@Request
24 return "name value = " + +
25 }
26
27* @RequestMapping(value = "/api
28 public String path(@PathVaria
29 return "path value = " +
30 }
31 }
31
                 > # com.example.demo
                 > # com.example.demo.annotations
> # com.example.demo.bean
                        # com.example.demo.config
                                                                                                                                                                                               @RequestMapping(value = "/person", method = RequestMethod.POST)
public String person(@Valid @RequestBody Person person) {
    return "person";

    → 
    □ com.example.demo.controller
    → 
    □ PersonController.java

                          > La TestController.java
                                                                                                                                                                                                @RequestMapping(value = "/person", method = RequestMethod.GET)
public String person(@RequestParam(value = "name", required = true) String name ) {
    return "name value = " + name;
}
                > # com.example.demo.exceptions
> # com.example.demo.res
                     = com.example.demo.validators
                       templates
                                                                                                                                                                                                   @RequestMapping(value = "/api/(path)", method = RequestMethod.GET)
public String path(@PathVariable(value = "path", required = true) String path ) {
    return "path value = " + path;
                         application.properties
           > 🕮 src/test/java
           > M JRE System Library [JavaSE-1.8]
           > 🗁 bin
          > 🔑 src
```

 Create simple response error bean as below: <a href="https://github.com/fishyimm/spring-boot-request-validation/tree/master/src/main/java/com/example/demo/res">https://github.com/fishyimm/spring-boot-request-validation/tree/master/src/main/java/com/example/demo/res</a>



6. Create RestExceptionHandler with **@ControllerAdvice** to handle exception that occur in controller as below:

https://github.com/fishyimm/spring-boot-request-

validation/blob/master/src/main/java/com/example/demo/exceptions/RestResponseEntity ExceptionHandler.java

```
🖹 🚼 🔝 💟 🗖 🖟 Person.java 🖟 PersonController.java 🖟 RestResponseEntityExceptionHandler.java 🖂
Package Explorer ⋈
                                                 1 package com.example.demo.exceptions;
## demo [boot] [devtools]

√ # src/main/java

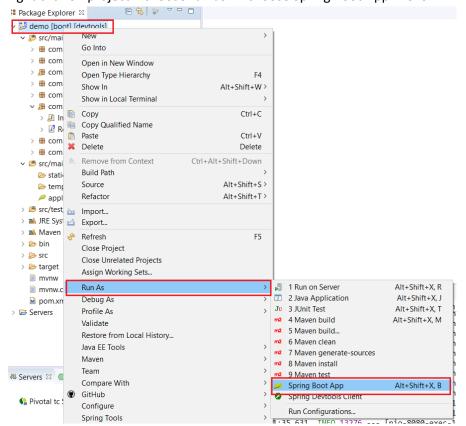
                                               3® import java.util.ArravList;
     > # com.example.demo
     > # com.example.demo.annotations
                                               24 @ControllerAdvice
     > 🚜 com.example.demo.bean
                                                    public class RestResponseEntityExceptionHandler extends ResponseEntityExceptionHandler {
     > # com.example.demo.config
       # com.example.demo.controller
                                                        public RestResponseEntityExceptionHandler() {

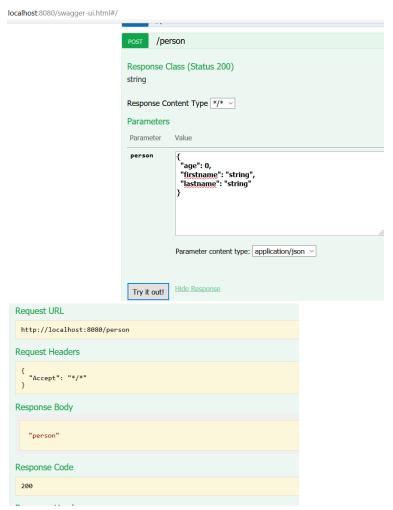
    # com.example.demo.exceptions

       > 🔝 InvalidRequestException.java
                                                        RestResponseEntityExceptionHandler.java
       com.example.demo.res
       # com.example.demo.validators
                                                             ListListfieldErrors = ex.getintln("handleMethodArgumentNotValid fieldErrors();
   for (FieldError fieldError : fieldErrors) {
   FieldErrorResource fieldErrorResource = new FieldErrorResource();
      static
      templates
       application.properties
                                                                 fieldErrorResource.setResource(fieldError.getObjectName());
                                                                 fieldErrorResource.setField(fieldError.getField());
fieldErrorResource.setCode(fieldError.getCode());
   > 👛 src/test/java
  > IRF System Library [JavaSF-1.8]
                                                                 fieldErrorResource.setMessage(fieldError.getDefaultMessage());
fieldErrorResources.add(fieldErrorResource);
  > Maven Dependencies
                                                            ErrorResponse error = new ErrorResponse();
error.setFieldErrors(fieldErrorResources);
  > 🗁 target
    mvnw
                                                             headers.setContentType(MediaType.APPLICATION_JSON);
    Imx.moq
                                                             return super.handleExceptionInternal(ex, error, headers, status, request);
> 🗁 Servers
```

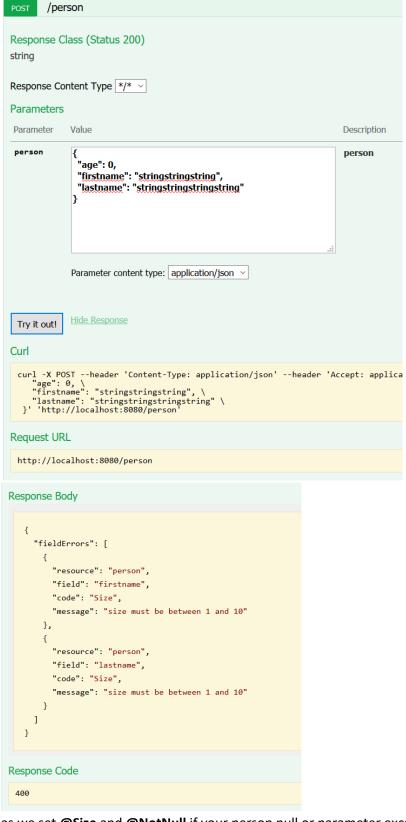
Now test it with swagger-ui >> start your project and go to <a href="http://localhost:8080/swagger-ui.html">http://localhost:8080/swagger-ui.html</a>

right click on project > choose run as >> choose Spring Boot App in STS IDE





with normal request it won't occur any error but try as below and see the result



as we set **@Size** and **@NotNull** if your person null or parameter exceed the length error will be occurred but how about **parameter age(int)** how to validate the size of it, since **@Size** cannot validate it so we need to create new validation for **int value** 

8. Create new annotation for validation you can check Size for example

```
2⊕* JBoss, Home of Professional Open Source.
 17 package javax.validation.constraints;
 18
 19 import java.lang.annotation.Documented;
 20 import java.lang.annotation.Retention;
 21 import java.lang.annotation.Target;
 22 import javax.validation.Constraint;
 23 import javax.validation.Payload;
 25 import static java.lang.annotation.ElementType.ANNOTATION_TYPE;
 26 import static java.lang.annotation.ElementType.CONSTRUCTOR;
 27 import static java.lang.annotation.ElementType.FIELD;
 28 import static java.lang.annotation.ElementType.METHOD;
 29 import static java.lang.annotation.ElementType.PARAMETER;
 30 import static java.lang.annotation.RetentionPolicy.RUNTIME;
33	ext{@} * The annotated element size must be between the specified boundaries (included).\Box
 47 @Target({ METHOD, FIELD, ANNOTATION_TYPE, CONSTRUCTOR, PARAMETER })
 48 @Retention(RUNTIME)
 49 @Documented
 50 @Constraint(validatedBy = { })
 51 public @interface Size {
 53
        String message() default "{javax.validation.constraints.Size.message}";
        Class<?>[] groups() default { };
        Class<? extends Payload>[] payload() default { };
 58
 60⊕
         * @return size the element must be higher or equal to...
 62
        int min() default 0;
         * @return size the element must be lower or equal to \( \)
 65⊕
        int max() default Integer.MAX_VALUE;
 68
         * Defines several {@link Size} annotations on the same element.
        @interface List { ...
81 }
 82
```

and create **inrange annotation** for validate int value what u need to create more is validatedBy as picture below:

```
🖹 💲 💝 😑 🗖 🚺 InRange.java 🏻 👪 Size.class

□ Package Explorer 
□

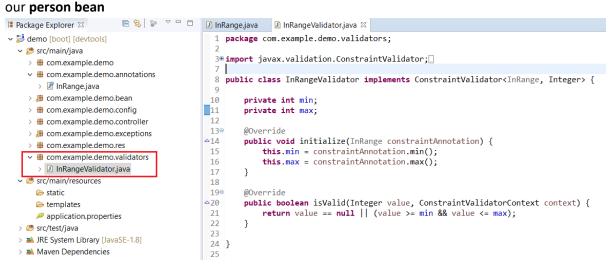
√ № demo [boot] [devtools]

                                               1 package com.example.demo.annotations;
  3 mport java.lang.annotation.Documented:
     > # com.example.demo

w ## com.example.demo.annotations

                                             14 @Target({ElementType.METHOD, ElementType.FIELD, ElementType.PARAMETER})
      > 🗗 InRange.java
                                                 @Retention(RetentionPolicy.RUNTIME)
      # com.example.demo.bean
    > # com.example.demo.config
                                             17 @Constraint(validatedBy = { InRangeValidator.class })
     > # com.example.demo.controller
                                             18
                                                     String message() default "{javax.validation.constraints.Size.message}";
                                             19
    > # com.example.demo.exceptions
                                             20
    > # com.example.demo.res
                                                     Class<?>[] groups() default {};
     > # com.example.demo.validators
   Class<? extends Payload>[] payload() default {};
      static
      int min() default Integer.MIN_VALUE;
      application.properties
                                                     int max() default Integer.MAX_VALUE;
  > # src/test/iava
                                             28 3
   > Mark JRE System Library [JavaSE-1.8]
```

this validator will validate the value of the parameter size now try to add this annotation in





here we go got error already cause of we sending age = 0 and validator give us an error as expected

```
Response Body

{
    "fieldErrors": [
        {
             "resource": "person",
             "field": "age",
             "code": "InRange",
             "message": "size must be between 1 and 10"
        }
    ]
}

Response Code

400
```

9. Now we gonna try to validate nested bean, create another simple bean and add person in it as below

```
🖹 💲 💆 🗖 🔲 TestController,java 🔃 TestBean,java 🛭
■ Package Explorer \( \times \)
1 package com.example.demo.bean;
  import javax.validation.constraints.NotNull;
                                                     3
     > # com.example.demo
     > # com.example.demo.annotations
                                                    5
                                                       public class TestBean {
    v # com.example.demo.bean
                                                     6
         Person.java
                                                     7⊜
                                                            @NotNull
      > 🗓 TestBean.java
                                                     8
                                                            private Person person;
                                                    9
      # com.example.demo.config
                                                   10⊝
                                                            public Person getPerson() {
    v # com.example.demo.controller
                                                   11
                                                                 return person;
         PersonController java
                                                    12
       > 🛂 TestController.java
                                                   13
     > # com.example.demo.exceptions
                                                   14⊜
                                                            public void setPerson(Person person) {
     > # com.example.demo.res
                                                   15
                                                                 this.person = person;
     > # com.example.demo.validators
                                                   16
                                                   17

→ 

B src/main/resources

                                                   18⊜
                                                            @Override
      static
                                                   19
                                                            public String toString() {
      return "TestBean [person=" + person + "]";
                                                   20
       application.properties
                                                   21
  > # src/test/java
                                                   22
  > A JRE System Library [JavaSE-1.8]
                                                   23 }

√ № demo [boot] [devtools]

                                           1 package com.example.demo.controller;
  🗸 🥭 src/main/java
                                         3⊕ import javax.validation.Valid;
    > # com.example.demo
    > # com.example.demo.annotations
                                          15 @RestController

√ 

⊕ com.example.demo.bean

                                          16 public class TestController {
      > Person.java
                                                  @RequestMapping(value = "/test-bean", method = RequestMethod.POST)
public String test1(@Valid @RequestBody TestBean testBean) {
      > 🗾 TestBean.java
                                          18∈
    > # com.example.demo.config
                                           19
                                          20
21
22
    # com.example.demo.controller
                                                      return "test-bean";
      > F PersonController.iava
      > 🛂 TestController.java
    > # com.example.demo.exceptions
    = com example demo res
```

10. Now try it on swagger and get success result but why? Validation does not work?



11. You need a little trick here just add **@Valid** on person bean inside **TestBean** as picture below:

```
☑ TestBean.java 

□ 
 package com.example.demo.bean;
  3⊕ import javax.validation.Valid; ...
 6 public class TestBean {
80
       @NotNull
9 @Valid
10
       private Person person;
11
      public Person getPerson() {
12⊜
13
          return person;
 14
15
 16⊜
      public void setPerson(Person person) {
17
18 }
         this.person = person;
 19
 20⊝
      @Override
△21
      public String toString() {
         return "TestBean [person=" + person + "]";
22
 23
25 }
```

now try again, you will see that got some error as expected.

