Hibernate Validator

1.In the last project’s frontendproject pom.xml,add the hibernate validator dependency.

a.In the pom.xml inside the properties , add:

<hibernate.validator.version>5.1.2.Final</hibernate.validator.version>

b.Inside the dependencies , add”

<!-- Hibernate validator dependency -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>${hibernate.validator.version}</version>

</dependency>

2.Also do the same in the backendproject.

3.Annotate fields in product.cs .

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int productId;

@Size(min=1, max=30,message="product name should be between 1 to 30 characters long")

private String productName;

@NotBlank(message="product category should not be blank")

private String productCategory;

@Size(min=20, max=300,message="product description should be between 20-300 characters long")

private String productDescription;

@NotBlank(message="product keyword should not be blank")

private String productKeyword;

@NotBlank(message="product image should not be blank")

private String productImage;

@Range(min=1,max=1000000,message="product price should be between 1-1000000")

private int productPrice;

@Range(min=0,max=100,message="product quantity should be between 1-100")

private int productQuantity;

4.Has-error to be updated in all input fields ,in productCRUD.js.

<div class="container">

<div class="row">

<div class="col-md-3"></div>

<div class="col-md-6">

<div class="form-container">

<h3>Product Form</h3>

<form:form action="productData.do" method="POST"

commandName="product" modelAttribute="product">

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="productName">Product

Name</label>

<form:input path="productName" class="form-control input-sm" />

<div class="has-error">

<form:errors style="color:red" path="productName" class="help-inline" />

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="productCategory">Product

Category</label>

<form:input path="productCategory" class="form-control input-sm" />

<div class="has-error">

<form:errors style="color:red" path="productCategory" class="help-inline" />

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="productDescription">Product

Description</label>

<form:input path="productDescription"

class="form-control input-sm" />

<div class="has-error">

<form:errors style="color:red" path="productDescription" class="help-inline" />

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="productKeyword">Product

Keyword</label>

<form:input path="productKeyword" class="form-control input-sm" />

<div class="has-error">

<form:errors style="color:red" path="productKeyword" class="help-inline" />

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="productImage">Product

Image</label>

<form:input path="productImage" class="form-control input-sm" />

<div class="has-error">

<form:errors style="color:red" path="productImage" class="help-inline" />

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="productPrice">Product

Price</label>

<form:input path="productPrice" class="form-control input-sm" />

<div class="has-error">

<form:errors style="color:red" path="productPrice" class="help-inline" />

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="productQuantity">Product

Quantity</label>

<form:input path="productQuantity" class="form-control input-sm" />

<div class="has-error">

<form:errors style="color:red" path="productQuantity" class="help-inline" />

</div>

</div>

</div>

<div class="row">

<div class="form-actions floatRight">

<input type="submit" name="action" class="btn btn-primary btn-sm" value="Add" /></th>

</div>

<div class="col-md-3"></div>

</div>

</div>

</div>

<br>

<br />

<div class="row">

<div class="table-responsive col-md-12">

<table id="myDataTable1" class="table table-striped">

<thead>

<tr>

<th>Product ID</th>

<th>Product Image</th>

<th>Product Name</th>

<th>Product Category</th>

<th>Keywords</th>

<th>Price</th>

<th>Quantity</th>

<th>Edit Product</th>

<th>Delete Product</th>

</tr>

</thead>

</table>

</div>

</form:form>

</div>

</div>

5.In the Frontendproductcontroller ,update the doAction method:

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

public ModelAndView doActions(@ModelAttribute("product") @Valid Product product,BindingResult result , @RequestParam String action)

{

Product productResult=new Product();

if(result.hasErrors())

{

ModelAndView model1=new ModelAndView("page");

model1.addObject("title","Product Management");

model1.addObject("userClickProductCRUD", true);

return model1;

}

switch(action.toLowerCase())

{

case "add":

productDAO.addProducts(product);

productResult=product;

break;

case "edit":

productDAO.updateProduct(product);

productResult=product;

break;

case "delete":

productDAO.deleteProduct(product.getProductId());

productResult=product;

break;

case "search":

Product searchedProduct= productDAO.getProduct(product.getProductId());

productResult=searchedProduct!=null?searchedProduct:new Product();

break;

}

ModelAndView model1=new ModelAndView("page");

model1.addObject("product",productResult);

model1.addObject("productList",productDAO.getProducts());

model1.addObject("title","Product Management");

model1.addObject("userClickProductCRUD", true);

return model1;

}