Scribd
Carregar um documento
Search books, presentations, Pesquisar Documentos
Explorar

Documentos

- Livros - Ficção
- Livros - Não ficção
- Saúde e medicina
- Catálogos
- Documentos governamentais
- Guias/Manuais Como fazer
- Revistas e jornais
- Receitas/Menus
- Trabalho escolar
- + todas as categorias
-
- Apresentou
- Recente

Pessoas

- Autores
- Estudantes
- Pesquisadores
- Editores
- Governamental e sem fins lucrativos
- Negócios
- Músicos
- Artistas e designers
- Professores
- + todas as categorias
-
- Os mais seguidos
- Popular

- Registre-se
- |
- Logon

```
	protected Session getSession() {
		return sessionFactory.getCurrentSession();
	}
}
```

Now we have everything in core side to create a User. So moving to web project.

Web project

- **pom.xml**

```
<?xml version="1.0"?>
<project>
        <parent>

                <artifactId>exampleproject</artifactId>
                <groupId>blogspot.sezera.exampleproject</groupId>
                <version>1.0-SNAPSHOT</version>
        </parent>
        <modelVersion>4.0.0</modelVersion>
        <groupId>blogspot.sezera.exampleproject.web</groupId>
        <artifactId>exampleproject.web</artifactId>
        <packaging>war</packaging>
        <name>exampleproject.web Maven Webapp</name>
        <version>1.0-SNAPSHOT</version>
        <url>http://maven.apache.org</url>
        <build>
                <finalName>exampleproject.web</finalName>
                <!--MAVEN JETTY PLUGIN-->
                <plugins>
                        <plugin>
                                <groupId>org.mortbay.jetty</groupId>
                                <artifactId>maven-jetty-plugin</artifactId>
                                <version>6.1.10</version>
                                <configuration>
                                        <scanIntervalSeconds>3</scanIntervalSeconds>
                                </configuration>


                        </plugin>
                </plugins>


        </build>
```

```
        <dependencies>
                <dependency>
                        <groupId>blogspot.sezera.exampleproject.core</groupId>
                        <artifactId>exampleproject.core</artifactId>
                        <version>1.0-SNAPSHOT</version>
                </dependency>
                <dependency>
                        <groupId>junit</groupId>
                        <artifactId>junit</artifactId>
                        <version>3.8.1</version>
                        <scope>test</scope>
        </dependency>
        <dependency>

            <groupId>javax.faces</groupId>
            <artifactId>jsf-api</artifactId>

            <version>1.2_02</version>
        </dependency>
        <dependency>
            <groupId>com.sun.facelets</groupId>
            <artifactId>jsf-facelets</artifactId>
            <version>1.1.11</version>
        </dependency>
        <dependency>
            <groupId>javax.faces</groupId>
            <artifactId>jsf-impl</artifactId>
            <version>1.2-b19</version>
        </dependency>
        <dependency>
            <groupId>taglibs</groupId>
            <artifactId>standard</artifactId>
            <version>1.1.2</version>
        </dependency>
        <dependency>
            <groupId>jstl</groupId>
            <artifactId>jstl</artifactId>
            <version>1.1.2</version>

        </dependency>
        <dependency>
            <groupId>org.richfaces.ui</groupId>
            <artifactId>richfaces-ui</artifactId>
            <version>3.1.0</version>
        </dependency>


    </dependencies>
    <repositories>
        <repository>
            <id>repository.jboss.com</id>
            <name>Jboss Repository for Maven</name>
            <url>http://repository.jboss.com/maven2/</url>
            <layout>default</layout>
        </repository>
    </repositories>
</project>
```

- **UserController.java**

```
package blogspot.sezera.exampleproject.controller;

import blogspot.sezera.exampleproject.service.UserService;

public class UserController {

        private UserService service;
        private String username;
        private String password;

        public UserController(){

        }
        public String getUsername() {
                return username;
        }

        public void setUsername(String username) {
                this.username = username;
        }

        public String getPassword() {
                return password;
        }

        public void setPassword(String password) {
                this.password = password;
        }

        public void setService(UserService service) {
                this.service = service;
        }

        public void createUser(){
                service.createUser(username, password);
        }
        public UserService getService() {
                return service;
        }
}
```

- **EmailValidator.java**

```
package blogspot.sezera.exampleproject.validator;

import java.util.ResourceBundle;

import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;

public class EmailValidator implements Validator{

        public void validate(FacesContext arg0, UIComponent arg1, Object arg2)
                        throws ValidatorException {

                String email = arg2.toString();
                if(email.contains("@")==false){
                        //error message
                 ResourceBundle bundle =
ResourceBundle.getBundle("messages",arg0.getCurrentInstance().getViewRoot().getLocale());
                 FacesMessage msg = new FacesMessage(bundle.getString("emailNotValid"));
                 throw new ValidatorException(msg);
                }

        }
}
```

To validate user email address when creating user I have created a basic EmailValidator. It validates if user email address contains "@". If not a ValidatorException is thrown.

- **messages_en_US.properties**

```

emailNotValid=E-mail is not valid
```

```
emailNotValid=E-mail is not valid
```

- **newUser.xhtml**

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:ui="http://java.sun.com/jsf/facelets"
        xmlns:f="http://java.sun.com/jsf/core"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:a4j="http://richfaces.org/a4j"
        xmlns:rich="http://richfaces.org/rich">
<head>
<title>New User</title>
</head>
<body>
<f:view>

        <h:form>
          <table>
                <tr>
                <td><h:outputLabel value="UserName "></h:outputLabel></td>
                <td><h:inputText id="email" value="#{userController.username}">
                        <f:validator validatorId="emailValidator" />
                        <a4j:support event="onkeyup" requestDelay="300"
reRender="output"/>
                </h:inputText>
                </td>
                <td>
                 <h:outputLabel id="output">
<h:message for="email"/>
</h:outputLabel>
                </td>
                </tr>
                <tr>
                <td><h:outputLabel value="Password"></h:outputLabel></td>
                <td><h:inputSecret
value="#{userController.password}"></h:inputSecret><br/></td>
                </tr>
                </table>
                <h:commandButton value="Create User"
action="#{userController.createUser}"></h:commandButton>
        </h:form>
```

```
            </h:form>
</f:view>
</body>
</html>
```

When user started to write his email address, page send an AJAX request to server in every 300 second and EmailValidator executes. If there is an error in user email, page renders an error message right of user email input text area.

- **web.xml**

```
<?xml version="1.0"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee  http://java.sun.com/xml/ns/j2ee/web-
app_2_4.xsd">
            <display-name>ExampleProject Web</display-name>

            <context-param>
                        <param-name>contextConfigLocation</param-name>
                        <param-value>classpath*:/main-Context.xml</param-value>
            </context-param>
            <context-param>
                        <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
                        <param-value>.xhtml</param-value>
            </context-param>
            <context-param>
                        <param-name>org.richfaces.SKIN</param-name>
                        <param-value>classic</param-value>
```

Leave a Comment

You must be logged in to leave a comment.
Enviar
Caracteres: 400

You must be logged in to leave a comment.
Enviar
Caracteres: ...

A Tutorial for Spring Hibernate JSF Richfaces

Download or Print
4,844 Reads
Informações e classificação

Categoria:	*Não classificado.*
Rating:	
Upload Date:	09/14/2010
Copyright:	Atribuição não comercial
Tags:	*This document has no tags.*

Flag document for inapproriate content

Uploaded by
Chandru Jangin

Seguir

Search This Document

Pesquisar

TIP Press Ctrl-F⌘F to quickly search anywhere in the document.

More from This User

Documentos relacionados

More From This User

39 p.
7264537 Struts Framework

39 p.
7264537 Struts Framework

19 p.
A Tutorial for Spring Hibernate JSF Richfaces

Next

Carregar um documento

Search books, presentations, Pesquisar Documentos

- Siga-nos!
- scribd.com/scribd
- twitter.com/scribd
- facebook.com/scribd

- Sobre
- Imprensa
- Blog
- Parceiros
- Scribd 101
- Material da web
- Suporte
- Perguntas frequentes
- Desenvolvedores/API
- Vagas
- Termos
- Copyright
- Privacidade

Idioma:
Português