

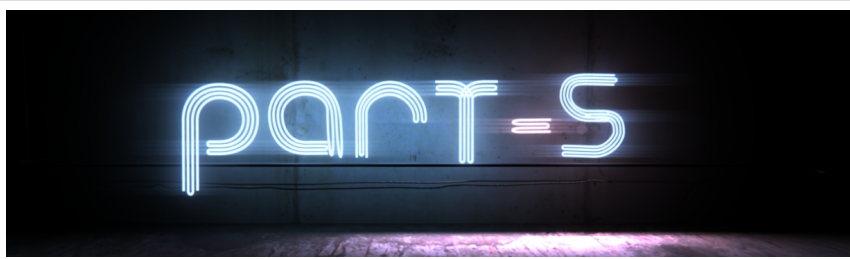
# Diary of a Java Enthusiast

[About Me](#)   [Special Thanks](#)
[Java EE](#)   [Part 5 – Downloading PrimeFaces themes / creating and installing custom themes and overriding PrimeFaces style sheets.](#)
[« Part 4 – Lazy Loading and Pagination with JSF 2.1 / Primefaces 3.4.2 DataTable and Form validation](#)
[Part 6 – Generic Primefaces LazySorter »](#)

## Part 5 – Downloading PrimeFaces themes / creating and installing custom themes and overriding PrimeFaces style sheets. 6

25 Sep 2012 | [Java EE](#)

As you know PrimeFaces enables applying new themes and in this part we are going to apply a different theme to our web application.



Download the theme from repository by adding the dependency to the project's pom.xml file and then run mvn install;

```
1 <dependency>
2   <groupId>org.primefaces.themes</groupId>
3   <artifactId>afterwork</artifactId>
4   <version>1.0.8</version>
5 </dependency>
```

After we add the dependency we will need to add a parameter (between "web-app" tags) to the project's web.xml file as well, and the context-param element will look like the following example.

```
1 <context-param>
2   <param-name>primefaces.THEME</param-name>
3   <param-value>aristo</param-value>
4 </context-param>
```

After all these changes our pom.xml file will look like this

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://
3
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>com.nz</groupId>
6   <artifactId>SimpleCRUD</artifactId>
7   <version>1.0-SNAPSHOT</version>
8   <packaging>war</packaging>
9   <name>SimpleCRUD</name>
10
11   <build>
12     <plugins>
13       <plugin>
14         <groupId>org.apache.maven.plugins</groupId>
15         <artifactId>maven-compiler-plugin</artifactId>
16         <version>${maven-compiler-plugin-version}</version>
17         <configuration>
```

### Blogroll

- > Arquillian testing tool for Java EE
- > PrimeFaces JSF Component Suite

### Recent Posts

- > Glassfish4 and JSF 2.2 ViewScope support (JSF 2.2, CDI, JPA 2 (ORM) / EclipseLink, JAAS, MySQL, example CRUD web application)
- > My Book Review – Instant PrimeFaces Starter- Packt Publishing
- > Reviewing Instant PrimeFaces Starter- Packt Publishing
- > My Book Review – PrimeFaces Cookbook by Oleg Varaksin and Mert Çalışkan
- > Reviewing PrimeFaces Cookbook – Packt Publishing – Oleg Varaksin, Mert Çalışkan

### Recent Comments

- > simtay on Part 4 – Lazy Loading and Pagination with JSF 2.1 / Primefaces 3.4.2 DataTable and Form validation
- > simtay on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA
- > Mohamed on Part 4 – Lazy Loading and Pagination with JSF 2.1 / Primefaces 3.4.2 DataTable and Form validation
- > dukgo on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA
- > simtay on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA

### Archives

- > November 2013   > August 2013
- > April 2013   > March 2013
- > January 2013   > September 2012

### Categories

```

18         <source>${maven-source}</source>
19         <target>${maven-target}</target>
20         <encoding>${project.build.sourceEncoding}</enco
21     </configuration>
22 </plugin>
23 <plugin>
24     <groupId>org.apache.maven.plugins</groupId>
25     <artifactId>maven-resources-plugin</artifactId>
26     <version>${maven-resources-plugin-version}</version>
27     <configuration>
28         <encoding>${project.build.sourceEncoding}</enco
29     </configuration>
30 </plugin>
31 </plugins>
32 </build>
33
34 <repositories>
35     <repository>
36         <id>prime-repo</id>
37         <name>PrimeFaces Maven Repository</name>
38         <url>http://repository.primefaces.org</url>
39         <layout>default</layout>
40     </repository>
41     <repository>
42         <url>http://download.eclipse.org/rt/eclipselink/maven.r
43         <id>eclipselink</id>
44         <layout>default</layout>
45         <name>Repository for library Library[eclipselink]</name
46     </repository>
47 </repositories>
48
49 <dependencies>
50     <dependency>
51         <groupId>org.eclipse.persistence</groupId>
52         <artifactId>eclipselink</artifactId>
53         <version>${eclipselink-persistence-version}</version>
54         <scope>provided</scope>
55     </dependency>
56     <dependency>
57         <groupId>org.eclipse.persistence</groupId>
58         <artifactId>org.eclipse.persistence.jpa.modelgen.proces
59         <version>${eclipselink-jpa-version}</version>
60         <scope>provided</scope>
61     </dependency>
62     <dependency>
63         <groupId>org.eclipse.persistence</groupId>
64         <artifactId>javax.persistence</artifactId>
65         <version>${javax-persistence-version}</version>
66         <scope>provided</scope>
67     </dependency>
68     <dependency>
69         <groupId>javax</groupId>
70         <artifactId>javaee-web-api</artifactId>
71         <version>${javaee-web-api-version}</version>
72         <scope>provided</scope>
73     </dependency>
74     <dependency>
75         <groupId>javax.ejb</groupId>
76         <artifactId>ejb-api</artifactId>
77         <version>${ejb-api-version}</version>
78     </dependency>
79     <dependency>
80         <groupId>org.primefaces</groupId>
81         <artifactId>primefaces</artifactId>
82         <version>${primefaces-version}</version>
83     </dependency>
84     <dependency>
85         <groupId>org.primefaces.themes</groupId>
86         <artifactId>${primefaces-themes-artifact}</artifactId>
87         <version>${primefaces-themes-version}</version>
88     </dependency>
89     <dependency>
90         <groupId>javax.faces</groupId>
91         <artifactId>javax.faces-api</artifactId>
92         <version>${javax-faces-api-version}</version>
93         <scope>provided</scope>
94     </dependency>
95     <dependency>
96         <groupId>org.omnifaces</groupId>
97         <artifactId>omnifaces</artifactId>
98         <version>${omnifaces-version}</version>
99     </dependency>
100 </dependencies>
101
102 <properties>
103     <project.build.sourceEncoding>UTF-8</project.build.sourceEn
104
105     <maven-source>1.6</maven-source>
106     <maven-target>1.6</maven-target>

```

› Java EE

## Meta

› Log in

› Entries RSS

› Comments RSS

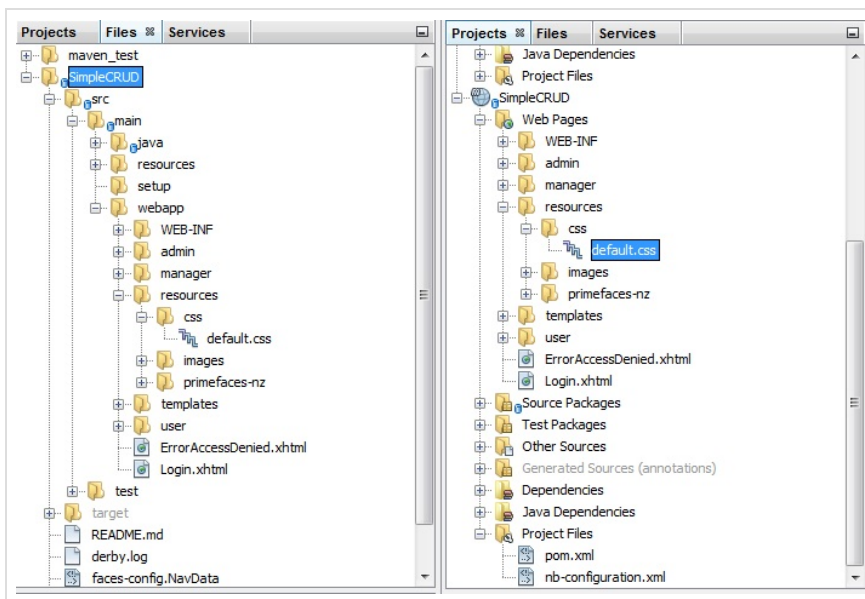
› WordPress.org

```

107 <maven-ejb-plugin-version>2.3</maven-ejb-plugin-version>
108 <maven-war-plugin-version>2.1.1</maven-war-plugin-version>
109 <maven-dependency-plugin-version>2.1</maven-dependency-plug
110 <maven-resources-plugin-version>2.4.3</maven-resources-plug
111 <maven-compiler-plugin-version>2.3.2</maven-compiler-plugin
112
113 <eclipselink-persistence-version>2.4.0</eclipselink-persist
114 <eclipselink-jpa-version>2.3.2</eclipselink-jpa-version>
115 <javax-persistence-version>2.0.3</javax-persistence-version>
116 <javaee-web-api-version>6.0</javaee-web-api-version>
117 <ejb-api-version>3.0</ejb-api-version>
118 <javax-faces-api-version>2.1</javax-faces-api-version>
119 <omnifaces-version>1.4.1</omnifaces-version>
120 <primefaces-version>3.4.2</primefaces-version>
121 <primefaces-themes-artifact>afterwork</primefaces-themes-ar
122 <primefaces-themes-version>1.0.8</primefaces-themes-version>
123
124 </properties>
125 </project>

```

If we want to override the theme css we need to create "resources and css" folders under webapp and name our css file default.css and add the corresponding css code to that file.



Let's say we want to change font size and all we need to do add the following css code to default.css file.

```

1 .ui-widget, .ui-widget .ui-widget {
2     font-size: 90% !important;
3 }

```

If we like to have a custom PrimeFaces theme we can easily create with JQuery ThemeRoller. Once we are done with customising the theme, we can download the theme. Unzip jquery-ui-[version].custom.zip and copy the custom-theme folder and we rename this folder to primefaces-nz or anything we like

After that, we also need to rename the css the file's name has to be theme.css and change image path to JSF resource path.

i.e.  
url(images/ui-icons\_222222\_256x240.png);  
has to be replaced with  
url("#{resource['primefaces-nz:images/ui-icons\_222222\_256x240.png']}");

after that we will need to change or add the parameter that is in the project's web.xml file.

```

1 <context-param>
2     <param-name>primefaces.THEME</param-name>
3     <param-value>primefaces-nz</param-value>
4 </context-param>

```

I have used PrimeFaces layout in our project it is very simple to use. The project layout page looks like the following example;

```

1 <!DOCTYPE html>
2 <html lang="en"
3     xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:f="http://java.sun.com/jsf/core"
5     xmlns:h="http://java.sun.com/jsf/html"

```

```

6      xmlns:ui="http://java.sun.com/jsf/facelets"
7      xmlns:p="http://primefaces.org/ui"
8  >
9      <h:head>
10         <title>Title</title>
11         <h:outputStylesheet library="css" name="default.css" />
12     </h:head>
13     <h:body>
14         <p:layout fullPage="true">
15             <p:layoutUnit position="north" size="75" header="Project" r
16                 <h:outputText value="" />
17             </p:layoutUnit>
18
19             <p:layoutUnit position="west" size="200" header="Menu" r
20                 <ui:include src="../templates/tiles/LeftMenu.xhtml" />
21             </p:layoutUnit>
22
23             <p:layoutUnit position="east" size="200" header="Help" r
24                 <h:outputText value="Right unit content." />
25             </p:layoutUnit>
26
27             <p:layoutUnit position="center">
28                 <ui:insert name="content">Put default content here,
29             </p:layoutUnit>
30         </p:layout>
31     </h:body>
32 </html>

```

I inserted jsf contents to the center layoutUnit

i.e.

```

1  <ui:composition template="/templates/layout.xhtml"
2      xmlns="http://www.w3.org/1999/xhtml"
3      xmlns:f="http://java.sun.com/jsf/core"
4      xmlns:h="http://java.sun.com/jsf/html"
5      xmlns:ui="http://java.sun.com/jsf/facelets"
6      xmlns:p="http://primefaces.org/ui"
7  >
8      <ui:define name="content">
9          <h:form styleClass="loginPanelStyle">
10              <p:growl id="msgs" showDetail="true" sticky="false"
11                  <p:panelGrid columns="2">
12                      <f:facet name="header">
13                          Login Panel
14                      </f:facet>
15                      <h:outputText value="Username : "></h:outputText>
16                      <p:inputText id="username" value="#{loginController.
17                          <f:validateLength minimum="1" />
18                      </p:inputText>
19                      <h:outputText value="Password : "></h:outputText>
20                      <p:password id="password" value="#{loginController.p
21                          <f:validateLength minimum="1" />
22                      </p:password>
23                      <f:facet name="footer">
24                          <p:commandButton value="Submit" update="msgs" action
25                      </f:facet>
26                  </p:panelGrid>
27              </h:form>
28          </ui:define>
29 </ui:composition>

```

For more information about layout please visit PrimeFaces [showcase](#) and check out the demo.

Download the source code [here](#)

**Please try live demo** (Username: Admin, Password:1234)

I hope you enjoyed it.

Looking forward to posting my next project.

Thank you

---

6 thoughts on “Part 5 – Downloading PrimeFaces themes / creating and installing custom themes and overriding PrimeFaces style sheets.”



Reply

**ESP**

Oct 18, 2012 10:18 pm

Can you make tutorial about upload and reading a text file then save the contents to the database using jsf, ejb, jpa ...



Reply

**simtay**

Nov 1, 2012 12:10 pm

Sure, I will include this feature in my next project.  
Thanks



Reply

**Oc**

Dec 19, 2012 12:17 pm

This is beautiful! Fantastic piece of work!!



Reply

**simtay**

Dec 19, 2012 1:53 pm

Thanks Oc



Reply

**Halil Karakose**

Sep 17, 2013 2:47 am

Hi Emre  
Thanks for this brilliant tutorial. Until this section, my IDE couldn't identify LazyDataModel class since related dependency wasn't added to the pom. So, what about providing pom.xml in the beginning of this tutorial? In this way it would be more understandable.



Reply

**simtay**

Sep 17, 2013 12:27 pm

LazyDataModel comes with Primefaces dependency added in Part4! You shouldn't be having any problem on Part5, so it was already provided. Cheers

## Leave a Reply

**Author** (required)**Email** (will not be published)(required)**Website**

Just to prove you are a human, please answer the following math challenge either out loud or typing into the text box

 × one = three[b](#)[i](#)[link](#)[b-quote](#)[code](#)[close tags](#)

