

Table of Contents

JSF Tags.....	1
JSF h:graphicImage.....	1
Resource Bundles and Images JSF.....	2
JSF h:outputStylesheet.....	2
USING UI PARAM: DEFINED IN ANOTHER XHTML PAGE.....	2
JSF REDISPLAY Attribute.....	2
JSF Rendered Attribute.....	3
Grundlegende Arbeiten:.....	3
Internationalisierung:.....	4
f:view tag hinzufügen.....	5
JSF Standard Tag Reference.....	6
Permisssion Setting Linux.....	7
Starting Glassfish Server from CMD.....	7
Glassfish installation.....	7
Setup GlassFish 4.1 with JAVA 8 in Ubuntu 15.04.....	7
Starting HSQLDB Linux.....	7
HSQLDB SERVER Property Files Setting.....	7
Create Maven project in eclipse.....	8
How to insert data into hsqldb.....	9
How to know which framework is used? JPA or Hibernate.....	9
UNDERSTANDING JSF VIEW PARAMETERS.....	9
Creating-JSF-CDI-maven-project-on-eclipse.....	9
Create a Maven Project with CommandLine.....	9

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

JSF Tags

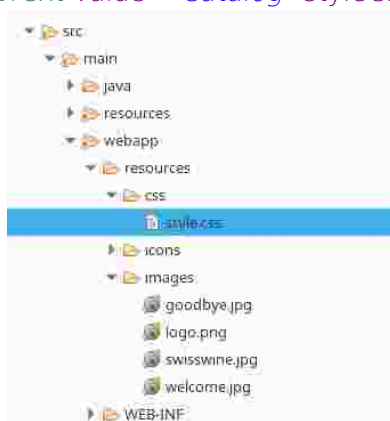
```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:f="http://xmlns.jcp.org/jsf/core"
xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
xmlns:p="http://primefaces.org/ui">
<h:head>
```

JSF h:graphicImage

```
<h:graphicImage library="images" name="logo.png" styleClass="hftmlogo"/>
</h:head>
<h:body>
<h:outputStylesheet name="css/style.css" />
```

```
<h:outputText value="myWinery" styleClass="title"/>
```

```
<h:outputText value="Catalog" styleClass="heading"/>
```



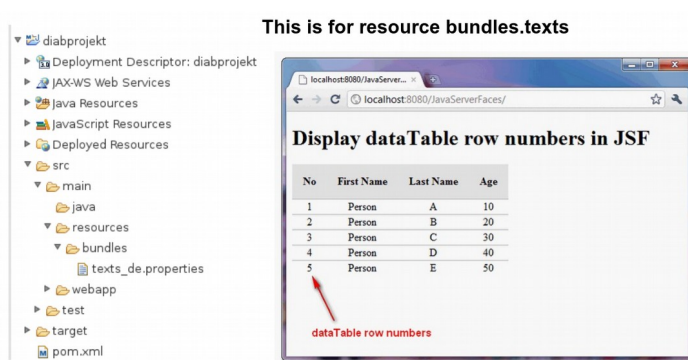
```
.info {
    color: black;
}
.error {
    background-color: yellow;
    font-weight: bold;
}
.swisswine{
    width: 100px;
    height: 100px;
    position: relative;
    float: right;
```

1. Boolean flag indicating whether or not a previously entered password should be rendered in form. Default is false.

Now add this attribute(redisplay="true") in your component.

2. String name = p.getName();

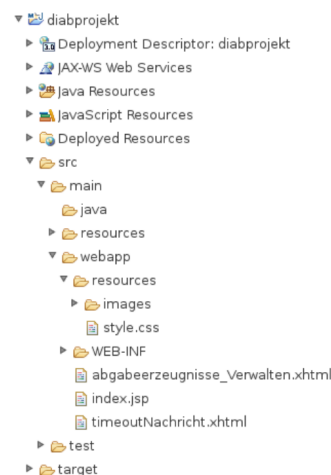
Resource Bundles and Images JSF.



Specifying resource directories

By default, Maven will look for your project's resources under **src/main/resources**.

This is for images



JSF h:outputStylesheet

NOTE: This code must be in between the opening and closing tags of the head section.
<h:outputStylesheet name="style.css" />

USING UI PARAM: DEFINED IN ANOTHER XHTML PAGE

```
<h:outputText value="#{pageTitle}" styleClass="title" />
<hr />
```

JSF REDISPLAY Attribute.

Is a boolean flag indicating whether or not a **previously** entered **password** should be rendered (**redisplayed**) in form component when the page is reloaded. (Default is false).

```
<h:outputLabel for="pass" value="Password: "/>
<h:inputText id="pass" redisplay="false" value="#{courseBean.password}"/>
```

Adding the attribute (**redisplay="true"**) into the component code, means the password will be redisplay again in the input component when the page is reloaded.

JSF Rendered Attribute.

NOTE: The Rendered attribute works on [Collection](#) and [Map](#) only

Rendered means to **Display** a jsf component.

Grundlegende Arbeiten:

1. Pom.xml anpassen!
2. CSS hinzufügen im xhtml im head.
 - `<h:outputStylesheet name="css/bootstrap.css" />`Achtung: Pfad ist abhängig vom Ablageort!

Template:

1. Im WEB-INF Ordner ein neues xhtml File erstellen -> template.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:f="http://xmlns.jcp.org/jsf/core"
xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
```

Namespace hinzufügen (xmlns:f dient später für die Internationalisierung)

2. Die anderen xhtml views inkludieren

```
<h:body>
  <div class="container">

    <div class="header">
      <ui:insert name="catalog" id="catalog">
        <ui:include src="catalog.xhtml" />
      </ui:insert>
    </div>

    <div class="header">
      <ui:insert name="header" id="header">
        <ui:include src="header.xhtml" />
      </ui:insert>
    </div>

    <div class="content">
      <ui:insert name="content" id="content" />
    </div>

  </div>
</h:body>
```

3. In den anderen xhtml Files die `ui:composition` Tags hinzufügen und das `Template` inkludieren (nur die Seiten, die als `<ui:Define name = "content">` ausgewiesen sind).

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE html>
```

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://xmlns.jcp.org/jsf/html"
  xmlns:f="http://xmlns.jcp.org/jsf/core"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
  template="/template.xhtml">
```

4. Mit `ui:param` den Page Titel angeben
Mit `ui:define` den Inhalt als Content deklarieren

```
<ui:param name="pageTitle" value="#{texts.catalog}" />

<ui:define name="content">

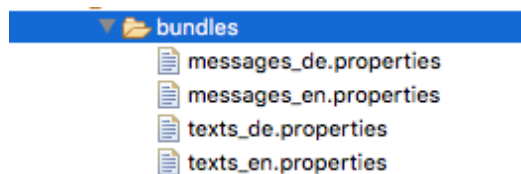
</ui:define>
```

```
</ui:composition>
```

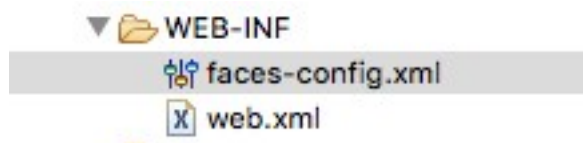
5. Evtl. Welcome File im web.xml anpassen (Startseite)
- ```
<!-- File appended to a request for a URL that is not mapped to a web component -->
<welcome-file-list>
 <welcome-file>catalog.xhtml</welcome-file>
</welcome-file-list>
```

## Internationalisierung:

Im package `src/main/resource` den ordner bundles erstellen mit folgenden files



6. Im Ordner WEB-INF die faces-config.xml erstellen



Im faces-config.xml die bundles hinzufügen

```
<faces-config version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
 http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">

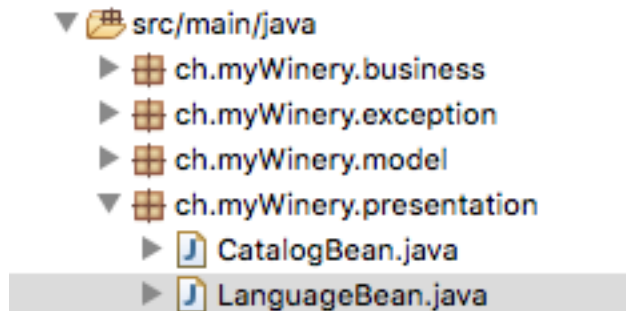
<application>
 <locale-config>
 <default-locale>de</default-locale>
 <supported-locale>en</supported-locale>
 </locale-config>
 <resource-bundle>
 <base-name>bundles.texts</base-name>
```

```

 <var>texts</var>
 </resource-bundle>
 <message-bundle>bundles.messages</message-bundle>
</application>
</faces-config>

```

7. Im Package ch.myWinery.presentation das LanguageBean erstellen



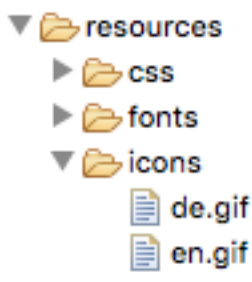
Flagge hinzufügen um auf die Sprache ändern zu können.

```

<h:panelGrid columns="3">
 <h:commandLink action="#{languageBean.english}">
 <h:graphicImage library="icons" name="en.gif"
 styleClass="languageIcon" />
 </h:commandLink>
 <h:commandLink action="#{languageBean.german}">
 <h:graphicImage library="icons" name="de.gif"
 styleClass="languageIcon" />
 </h:commandLink>
</h:panelGrid>

```

Die entsprechenden Icons müssen in das resource Folder



f:view tag hinzufügen

Im Template das f:view tag hinzufügen, sonst bleibt die Umgestellte Sprache nur auf der entsprechenden Seite erhalten.

```

<f:view locale="#{languageBean.locale}">
 <h:body>

 <div class="container">
 <div class="menu">
 <ui:insert name="menu" id="menu">
 <ui:include src="menu.xhtml" />
 </ui:insert>
 </div>

 <div class="header">
 <ui:insert name="header" id="header">
 <ui:include src="header.xhtml" />
 </ui:insert>
 </div>
 </div>
 </h:body>
</f:view>

```

```

 </ui:insert>
 </div>

 <div class="content">
 <ui:insert name="content" id="content" />
 </div>

</div>
</h:body>
</f:view>

```

## JSF Standard Tag Reference

<http://www.developersbook.com/jsf/jsf-tags/jsf-standard-components.php>

Core	HTML
<a href="#">f:actionListener</a>	<a href="#">h:column</a>
<a href="#">f:attribute</a>	<a href="#">h:commandButton</a>
<a href="#">f:converter</a>	<a href="#">h:commandLink</a>
<a href="#">f:convertDateTime</a>	<a href="#">h:dataTable</a>
<a href="#">f:convertNumber</a>	<a href="#">h:form</a>
<a href="#">f:facet</a>	<a href="#">h:graphicImage</a>
<a href="#">f:loadBundle</a>	<a href="#">h:inputHidden</a>
<a href="#">f:param</a>	<a href="#">h:inputSecret</a>
<a href="#">f:selectitem</a>	<a href="#">h:inputText</a>
<a href="#">f:selectitems</a>	<a href="#">h:inputTextarea</a>
<a href="#">f:subview</a>	<a href="#">h:message</a>
<a href="#">f.validateDoubleRange</a>	<a href="#">h:messages</a>
<a href="#">f.validateLength</a>	<a href="#">h:outputFormat</a>
<a href="#">f.validateLongRange</a>	<a href="#">h:outputLabel</a>
<a href="#">f.validator</a>	<a href="#">h:outputLink</a>
<a href="#">f.valueChangeListener</a>	<a href="#">h:outputText</a>
<a href="#">f:verbatim</a>	<a href="#">h:panelGrid</a>
<a href="#">f:view</a>	<a href="#">h:panelGroup</a>
	<a href="#">h.selectBooleanCheckbox</a>
	<a href="#">h.selectManyCheckbox</a>
	<a href="#">h.selectManyListbox</a>
	<a href="#">h.selectManyMenu</a>
	<a href="#">h.selectOneListbox</a>
	<a href="#">h.selectOneMenu</a>
	<a href="#">h.selectOneRadio</a>

---

## Permission Setting Linux

sudo chmod -R 777 --> then the directory path /opt/Home\_Documents/hsqldb-2.3.4/hsqldb-2.3.4/db/

## Starting Glassfish Server from CMD

/opt/glassfish4/glassfish/bin/ ---> Enter ---> ./asadmin start-database

## Glassfish installation.

<http://blog.topspeedsnail.com/archives/6978>

## Setup GlassFish 4.1 with JAVA 8 in Ubuntu 15.04

<https://linuxide.com/ubuntu-how-to/setup-glassfish-4-1-java-8-ubuntu-15-04/>

## Starting HSQLDB Linux

Erste command fenster

=====

/opt/Home\_Documents/hsqldb-2.3.4/hsqldb-2.3.4/  
directory

===== Ordner

/opt/Home\_Documents/hsqldb-2.3.4/hsqldb-2.3.4/hsqldb/

java -cp ./lib/hsqldb.jar org.hsqldb.server.Server

Zweite commando fenster

=====

/opt/Home\_Documents/hsqldb-2.3.4/hsqldb-2.3.4/  
directory

===== Ordner

java -cp ./lib/hsqldb.jar org.hsqldb.util.DatabaseManagerSwing

## HSQLDB SERVER Property Files Setting.

server.database.0=db/wtmManager/wtmManager

server.dbname.0=wtmManager

server.database.1=db/wtm/wtm

server.dbname.1=wtm

server.database.2=db/myWinery/myWinery

server.dbname.2=myWinery

server.database.3=db/bmi/bmi

server.dbname.3=bmi

server.database.4=db/einzelprojekt/einzelprojekt

server.dbname.4=einzelprojekt

server.database.5=db/catalog-0.6/catalog-0.6

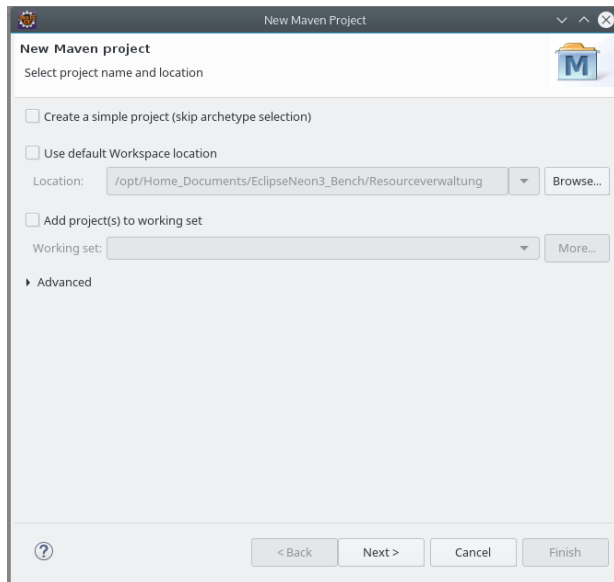
server.dbname.5=catalog-0.6

server.database.6=db/shipper/shipper  
server.dbname.6=shipper

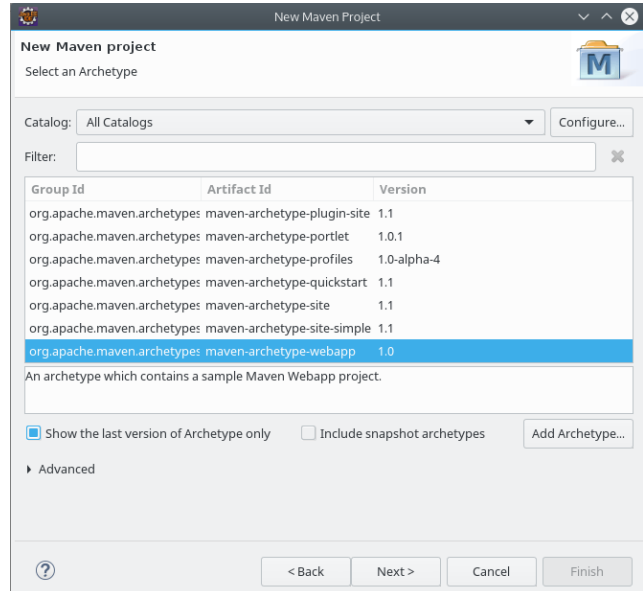
server.database.7=db/customerRegistration/customerRegistration  
server.dbname.7=customerRegistration  
server.port=9001

## Create Maven project in eclipse

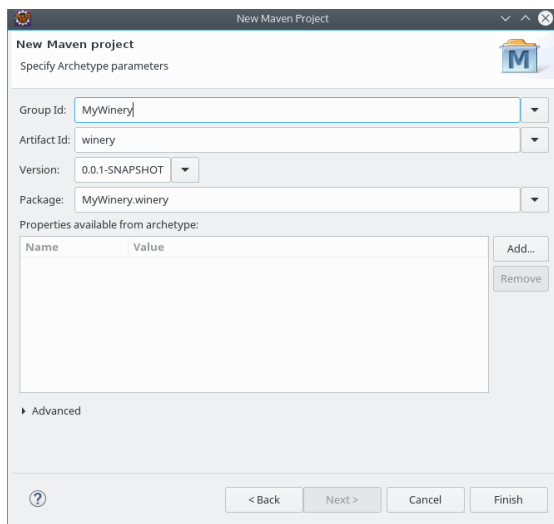
### Step 1



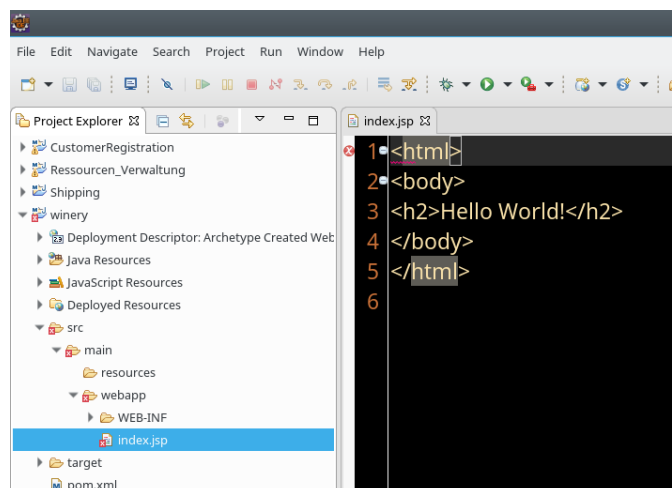
### Step 2



### Step 3



### Step 4



Error 1.

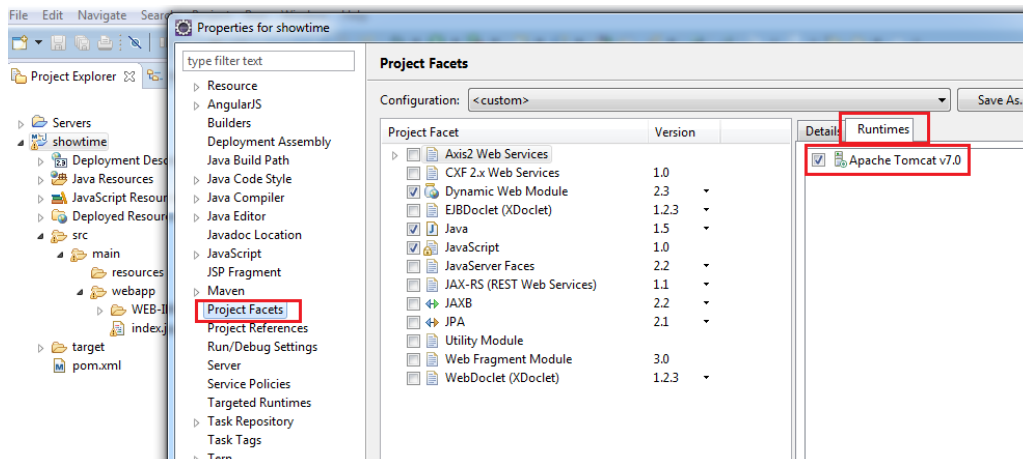
The superclass "javax.servlet.http.HttpServlet" was not

found on the Java Build Path

Solution:

<https://stackoverflow.com/questions/22756153/the-superclass-javax-servlet-http-http-servlet-was-not-found-on-the-java-build>





<http://www.java2blog.com/2015/09/how-to-create-dynamic-web-project-using.html>

Error: JavaServer Faces 2.2 requires Dynamic Web Module 2.5 or newer

Solution: <https://stackoverflow.com/questions/22361005/javaxserver-faces-2-2-requires-dynamic-web-module-2-5-or-newer>

Solution 2: <https://crunchify.com/how-to-fix-cannot-change-version-of-project-facet-dynamic-web-module-to-3-0-error-in-eclipse/>

How to insert data into hsqldb.

Example:

[https://www.tutorialspoint.com/hsqldb/hsqldb\\_insert\\_query.htm](https://www.tutorialspoint.com/hsqldb/hsqldb_insert_query.htm)

```
INSERT INTO customer VALUES(5,'Dorfstrasse 56','John','Bettlach','Robinsonovich');
```

How to know which framework is used? JPA or Hibernate

<https://stackoverflow.com/questions/23055437/how-to-know-which-framework-is-used>

UNDERSTANDING JSF VIEW PARAMETERS

<http://blog.webagesolutions.com/archives/643>

Creating-JSF-CDI-maven-project-on-eclipse

<https://dzone.com/articles/creating-jsfcdi-maven-project-on-eclipse>

**Create a Maven Project with CommandLine**

<http://websystique.com/maven/create-a-maven-project-with-commandline/>

```
mvn archetype:generate -DgroupId=gscs.ch-DartifactId=TruckersForum -
DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```