

## JAVASERVER FACES COURSE

# INTERNATIONALIZATION WITH JSF



By the expert: Eng. Ubaldo Acosta



**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson.

We are going to study the topic of Internationalization with JSF.

Are you ready? Let's go!

## INTERNATIONALIZATION WITH JSF

- JSF has full support for the I18n Java specification, for the Language specification in the Web application

Language Code / Subregion	Description
en	English
en_GB	English / British
en_US	English / United States
es	Spanish
es_MX	Spanish / Mexico

**JAVASERVER FACES COURSE**

www.globalmentoring.com.mx

A Web application may have the need to display its pages in several languages. JSF resolves this requirement through property files that support the I18n Java standard.

These files allow us to specify our Web page labels in several languages.

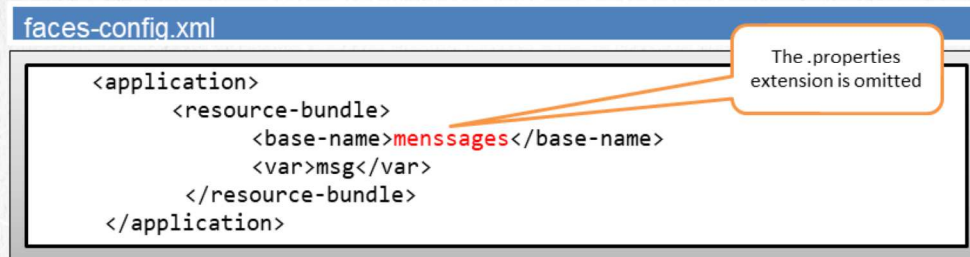
We can see in the figure that we have a message file with the default language, in this case Spanish. And we also have 2 languages, Spanish from Mexico and English from the United States.

The default language of the application can be specified either in the faces-config.xml file or programmatically in some action method of a Managed Bean. The necessary code is:

```
FacesContext.getCurrentInstance (). GetViewRoot (). SetLocale ("en_US");
```

## RESOURCE BUNDLE IN JSF

JSF can load tags from a property file (Resource Bundle)  
In the file faces-config.xml, the file to be used is configured.



We can use the properties file on the JSF page as follows :

```
<h:outputText value="#{msg['form.user']}" />
```

```
<h:commandButton value="#{msg.send}" type="submit" action="login" />
```

### JAVASERVER FACES COURSE

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

The resource bundle files will help us:

- Centralize the labels (labels) of our forms.
- Configure the application language (internationalization).

The file to be used can be specified in the faces-config.xml file or optionally, you can specify in the JSF page to use:

```
<f:loadBundle basename="menssages" var="msg"/>
```

Once we have configured the properties file, we can replace the labels of our JSF pages with syntax such as:

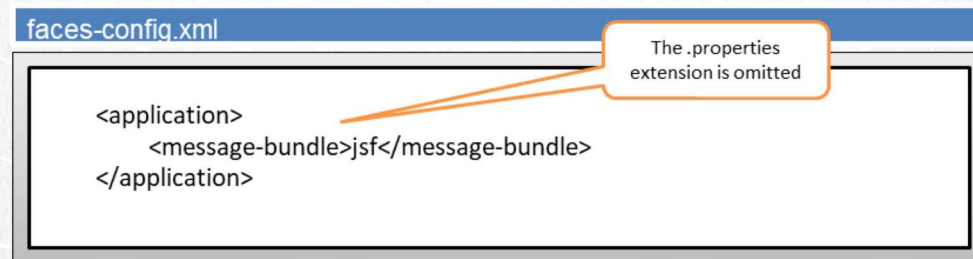
```
<h:outputText value="#{msg['form.user']}" />
```

```
<h:commandButton value="#{msg.send}" type="submit" action="login" />
```



## JSF MESSAGE OVERWRITING

- ✓ You must create a file (name does not matter) of properties  
`jsf.properties`
- ✓ The desired system messages are overwritten  
`javax.faces.component.UIInput.REQUIRED=Required Value`
- ✓ It is configured in the `faces-config.xml` file



**JAVASERVER FACES COURSE**  
www.globalmentoring.com.mx

JSF handles the location of errors and the information that occurs in events such as conversions, validations and other actions resulting from the JSF life cycle.

We can overwrite these messages, creating a new resource bundle file and configuring the entries that we want to modify.

In the example code we can see that we are overwriting the default value for the validation of some required field.

Some examples of constants to modify can be found in this resource:

<http://icursos.net/cursos/JSF/referencias/Messages.properties.txt>

**ONLINE COURSE**

# **JAVASERVER FACES (JSF)**

By: Ing. Ubaldo Acosta



**JAVASERVER FACES COURSE**  
[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)