

# Configuring MyFaces

{scrollbar}

Thanks to Bart Kummel for providing this text from his book [Apache MyFaces 1.2 Web Application Development](#)

There are two important configuration files for a JSF application— `web.xml` and `faces-config.xml`. We will discuss the basics of both of them.

## The web.xml configuration file

The `web.xml` configuration file resides in the `/WEB-INF/` directory of the (to be) deployed web application. It configures the web server part of the application. The `web.xml` file can be used to define which file types may be requested by users, which directories can be accessed, and so on. With regards to JSF, the most important task of `web.xml` is to tell the web server that there is such a thing as a Faces Servlet, and that URLs containing a certain pattern should be forwarded to that Faces Servlet. A minimal `web.xml` could look like this:

```
xmlweb.xml <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/webapp_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5"> <display-name>MyFaces Test Project</display-name> <servlet> <servlet-name>Faces Servlet</servlet-name> <servlet-class>javax.faces.webapp.FacesServlet</servlet-class> <load-on-startup>1</load-on-startup> </servlet> <servlet-mapping> <servlet-name>Faces Servlet</servlet-name> <url-pattern>/faces/*</url-pattern> </servlet-mapping> </web-app>
```

The part between the `<servlet>` tags tells the application server that it has to instantiate an object of the `javax.faces.webapp.FacesServlet` class as a Servlet and name it `Faces Servlet`. It will be started at the start up of the application. The part between the `<servlet-mapping>` tags tells the web server that any URL starting with `/faces/` immediately after the address of the server and the location of the application will be handled by that Servlet.

MyFaces core behavior can be customized, adding some web config params into your `WEB-INF/web.xml` file for your custom project in this way:

```
<context-param>
<param-name>org.apache.myfaces.SOME_USEFUL_PARAM</param-name>
<param-value>someValue</param-value>
</context-param>
```

On MyFaces site you can find an updated list of all web context parameters available for MyFaces Core. Take a look at these links:

- [Web Context Parameter List from JSF 2.1 sources](#)
- [Web Context Parameter List from JSF 2.0 sources](#)
- [Web Context Parameter List from JSF 1.2 sources](#)

Some additional descriptions about how and when to use these params can be found on:

- [Configuration of special features](#)

## The faces-config.xml configuration file

The `faces-config.xml` file defines the behavior of the Faces Servlet that is at the heart of a JSF application. Whereas a `web.xml` file is generally edited only at the start of a project or when structural changes are made to the application, `faces-config.xml` changes all the time, as the application grows. And while `web.xml` mostly contains general configuration options, a `faces-config.xml` file tends to be more specific to a certain application, as it may contain e.g. navigation details and other application-specific configurations. In JSF 2.0 the presence of a `faces-config.xml` file is no longer required, but in earlier JSF versions it is. A minimalistic `faces-config.xml` for JSF 1.2 may look like this:

```
xmlfaces-config.xml <?xml version="1.0" encoding="UTF-8"?> <faces-config xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd" version="1.2"> <application> <locale-config> <default-locale>en</default-locale> <supported-locale>en</supported-locale> <supported-locale>en_US</supported-locale> </locale-config> <message-bundle>my.company.Messages</message-bundle> </application> </faces-config>
```

## Additional configuration for Glassfish

To use MyFaces Core as the JSF implementation on a Glassfish 2.x application server, we have to make some additional settings in a GlassFish-specific configuration file—`sun-web.xml`. This file has to be in the `WEB-INF` folder of our project, along with most of the other configuration files, such as `web.xml` and `faces-config.xml`. The contents of the file should look like this:

```
xmlsun-web.xml <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE sun-web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Application Server 8.1 Servlet 2.4//EN" "http://www.sun.com/software/appserver/dtds/sun-web-app_2_4-1.dtd"> <sun-web-app> <class-loader delegate="false"/> <property name="useMyFaces" value="true"/> </sun-web-app>
```

This disables the default implementation and forces GlassFish to use MyFaces Core instead. Of course, we have to make sure that the MyFaces Core libraries are added properly to our application and configured correctly, as described in the previous sections.

{scrollbar}