

Preparing for JSF Application Development

To prepare for JSF application development, you should:

- Make sure that the Java EE: Java Server Faces [plugin](#) is enabled. This plugin is bundled with the IDE and enabled by default. However, you may have disabled it for some reason by now.
- Create a new [module](#) with JSF support enabled, or enable JSF support in an existing Java module.

On this page:

- [Making sure that the Java Server Faces plugin is enabled](#)
- [Enabling JSF support when creating a module](#)
- [Enabling JSF support for an existing module](#)

Making sure that the Java Server Faces plugin is enabled

1. [Open the Settings dialog](#) (e.g. Ctrl+Alt+S).
2. In the left-hand part of the dialog, under **IDE Settings**, select **Plugins**.
3. In the right-hand part of the dialog, on the [Plugins page](#), type `faces` in the search box. As a result, only the plugins whose names and descriptions contain `faces` are shown in the list of plugins.
4. If the check box to the left of **Java EE: Java Server Faces** is not selected, select it.
5. Click **OK** in the **Settings** dialog.
6. If suggested, restart IntelliJ IDEA.

Enabling JSF support when creating a module

Depending on the situation, you can choose to create a new project with a Java module or to add a Java module to an existing project.

1. Do one of the following:
 - If you are going to create a new project: click **Create New Project** on the [Welcome screen](#) or select **File | New Project**.
As a result, the [New Project wizard](#) opens.
 - If you are going to add a module to an existing project: [open the project](#) you want to add a module to, and select **File | New Module**.
As a result, the [New Module wizard](#) opens.
2. On the first page of the wizard, in the left-hand pane, select **Java Enterprise**. In the right-hand part of the page, specify the [JDK](#) and select the Java EE version to be supported.
3. If, at this step, you are ready to specify the application server you are going to deploy your application to (e.g. to test the application behavior), do so. This will result in the corresponding server-specific [run/debug configuration](#) for your module generated automatically. Otherwise, to be able to run your application, you will have to create the run/debug configuration yourself.

Select the server from the list or click **New** and select the server of interest. Then, specify the server settings:

- For a server installed locally, specify the path to the server installation directory. (Click  to select the directory in the [corresponding dialog](#).)
- For a hosted server (Cloud Foundry or CloudBees), specify your user account details.

4. Select the **Web Application** check box.

Select the version of the Servlet specification to be supported from the **Versions** list.

If you want the deployment descriptor `web.xml` file to be created, select the **Create web.xml** check box.

5. Select the **JSF** check box.

If you want the configuration file `faces-config.xml` to be created, select the **Create faces-config.xml** check box.

Select the required library option and, if necessary, specify the associated settings. You can choose to:

- Download and use a JSF implementation library (Mojarra).

To do that, under **Libraries**, select **Download**.

Now, to view or modify the associated options, click **Configure**, and in the **Downloading Options** dialog that opens:

- Select the library version.
- Specify the library name.
- Select the **library level** (global, project, or module).
- Under **Files to download**, select which of the files you want to download.
- Under **Copy downloaded files to**, specify the path to the destination folder. If you want to change the default path, click  and specify the folder location in the [dialog that opens](#).

- Use a JSF library IntelliJ IDEA is already aware of.

To do that, click **Use library** and select the required library from the list.

If necessary, configure the library settings (for example, change its name). This is done in the **Edit Library** dialog which you can open by clicking **Configure**.

- Create a new library using the appropriate JAR files available on your computer.

To do that, click **Use library** and then click **Create**. Select the required JAR files in the [dialog that opens](#). (For multiple selection, keep the Ctrl key pressed.)

If necessary, configure the new library (for example, change its name or **level**). To do that, click **Configure** and specify the required settings in the **Create Library** dialog.

- Postpone setting up the library until a later time. In this case, select **Set up library later**.

6. If you are going to use a JSF component library or libraries (e.g. PrimeFaces, RichFaces, etc.), select the corresponding check box or check boxes and specify the associated options. The procedure is similar to that for the JSF implementation library.

Click **Next**.

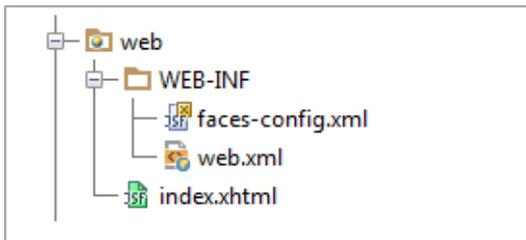
7. Specify the name and location settings. For more information, see [Project Name and Location](#) or [Module Name and Location](#).

Click **Finish**.

As a result, your new module will contain:

- The `web` and `WEB-INF` directories.
- The file `index.xhtml` in the `web` directory. With minor modifications, you can use this file as a starting page of your application.

- In the WEB-INF directory, if specified:
 - web.xml, the Web application deployment descriptor.
 - faces-config.xml, the JSF configuration file.



- If specified, the JSF library or libraries included in the [module dependencies](#).
- An [artifact](#) specification for you module.
- If you have specified the server, a [run/debug configuration](#) for running your application in the context of that server.

Enabling JSF support for an existing module

1. Open the [Project tool window](#) (e.g. **View | Tool Windows | Project**).
2. Right-click the module and select **Add Framework Support**.
3. In the left-hand pane of the **Add Frameworks Support** dialog, select the **Web Application** check box.

In the right-hand part of the dialog, select the version of the Servlet specification to be supported from the **Versions** list.

If you want the deployment descriptor `web.xml` file to be created, select the **Create web.xml** check box.

4. Select the **JSF** check box.

If you want the configuration file `faces-config.xml` to be created, select the **Create faces-config.xml** check box.

Select the required library option and, if necessary, specify the associated settings. You can choose to:

- Download and use a JSF implementation library (Mojarra).

To do that, under **Libraries**, select **Download**.

Now, to view or modify the associated options, click **Configure**, and in the **Downloading Options** dialog that opens:

- Select the library version.
- Specify the library name.
- Select the [library level](#) (global, project, or module).
- Under **Files to download**, select which of the files you want to download.
- Under **Copy downloaded files to**, specify the path to the destination folder. If you want to change the default path, click  and specify the folder location in the [dialog that opens](#).

- Use a JSF library IntelliJ IDEA is already aware of.

To do that, click **Use library** and select the required library from the list.

If necessary, configure the library settings (for example, change its name). This is done in the **Edit Library** dialog which you can open by clicking **Configure**.

- Create a new library using the appropriate JAR files available on your computer.

To do that, click **Use library** and then click **Create**. Select the required JAR files in the [dialog that opens](#). (For multiple selection, keep the **Ctrl** key pressed.)

If necessary, configure the new library (for example, change its name or [level](#)). To do that, click **Configure** and specify the required settings in the **Create Library** dialog.

- Postpone setting up the library until a later time. In this case, select **Set up library later**.

5. If you are going to use a JSF component library or libraries (e.g. PrimeFaces, RichFaces, etc.), select the corresponding check box or check boxes and specify the associated options. The procedure is similar to that for the JSF implementation library.

6. Click **OK** in the **Add Frameworks Support** dialog.

See Also

Concepts:

- [Facet](#)

- [Run/Debug Configuration](#)
- [Artifact](#)
- [JavaServer Faces \(JSF\)](#)
- [Available Facets and Their Dependencies](#)

Procedures:

- [Adding Modules to a Project](#)
- [Creating a Project from Scratch](#)
- [Configuring Project and Global Libraries](#)
- [Enabling Web Application Support](#)

Reference:

- [JSF Facet Page](#)

External Links:

- [Developing JSF applications in IntelliJ IDEA](#) 

Web Resources:

- [Developer Community](#) 