

Creating and Editing Faces Configuration

When you create different application elements you have to register them in the Faces configuration file `faces-config.xml`. The navigation rules (defining the navigation between the application pages) must be registered in this file as well.

To make this work more efficient, IntelliJ IDEA provides the corresponding graphical interface. This interface is available on the **Structure** tab, right in the editor. So you don't need to remember the correct order of tags. Just fill in the fields to [register the elements](#).

If necessary, you can use more than one Faces configuration file in the module or project. To create a new Faces configuration file, right-click the destination node in the **Project** tree, and select **New | XML Configuration File | Faces Config** from the context menu. When a new Faces configuration file is created, IntelliJ IDEA adds its definition to `web.xml`.

Here is a list of JSF elements that can be registered in the `faces-config.xml` file:

Component

The **Component** element represents a concrete `UIComponent` implementation class, and defines the state and behavior of a UI component. This behavior includes converting the value of a component to the appropriate markup, queuing events on components, performing validation, and other functionality.

Converter

The **Converter** element is an implementation of the `Converter` interface. You can create a custom converter to perform some specialized conversion, if the standard converters included with JavaServer Faces technology don't perform the data conversion that you need.

Managed Bean

The **Managed Bean** element represents an instance of a bean class. At runtime, the JavaServer Faces implementation processes the managed-bean element.

Referenced Bean

Declares a `JavaBean` that is available in a JSF scope. Unlike a managed bean, the application itself must instantiate this bean and ensure it is accessible in the desired scopes. The element is meant to be used as metadata for development tools.

Render Kit

The **Render Kit** element represents a concrete `RenderKit` implementation. You can create a custom render kit in order to render components to non-HTML clients, for example.

Validator

If the standard validators don't perform the validation checking you need, you can create a custom validator to validate user input. There are two ways to implement the validation code:

- Implement a backing bean method that performs the validation.
- Provide an implementation of the `Validator` interface to perform the validation.

Registering an application element in the Faces configuration file

1. Open the `faces-config.xml` file in the [editor](#).
2. Switch to the **Structure** tab.
3. Select the element you want to register in the tree view (for example, **Managed Bean**).

If you want to add a navigation rule, you can start by drawing the navigation diagram. For more information, see ["Defining Navigation Rules"](#).

4. On the main toolbar click **+**.
5. Specify the element's options. For more information, see ["Faces Config Structure"](#).

See Also

Concepts:

- [JSF](#)

Reference:

- [Faces Config Structure](#)

External Links:

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

Web Resources:

- [Developer Community](#)