More Component Types

In This Topic

- Presentation Component
- Service Component
- Form Component
- Servlet Component

In Adding an Input Field to a Component, the main XperienCentral component types element, panel and media item are described. This topic describes the component types presentation, service, form, and servlet.

Presentation Component

A presentation component contains all the files that make up the layout of the website environment. By default XperienCentral comes with one presentation plugin named “XperienCentral Corporate". In the context of this document, a style refers to the layout objects that are contained by a presentation component (including JSPs, images, etc.).

The presentation component in fact only copies JSP and static files to particular directories and as a result adjusts the style of the website environment.

- JSPs are copied from `/src/main/resources/editpresentation` to the `<backend web root>/WEB-INF/project/<Component bundle id>` directory
- Static files are copied from `/src/main/resources/static` to `<static web root>/static`

where:

- `<backend web root>` indicates the root folder of where the backend web application is located, for example `/Webmanager-webapps /Webmanager-backend-webapp/target/Webmanager-backend-webapp-x.x.x-SNAPSHOT`
- `<static web root>` indicates the root folder of where the static web application is located, for example `/Webmanager-webapps /Webmanager-static-webapp/target/Webmanager-static-webapp-x.x.x-SNAPSHOT`

When creating presentation plugins, be sure that they conform to the Plugin Development Guidelines (G083-G098, G138, G139, G140, G142, G143 and G144 in particular).

Using Multiple Styles at the Same Time

It is possible to have two (or more) styles installed on the same XperienCentral installation at the same time without them conflicting with each other. In order to do so the following two rules must be followed:

- The name property in the JSP descriptors must be unique across all installed styles. A good way to ensure its uniqueness is to prefix the name with the plugin ID.
- All static files must be located in a unique subfolder within `/src/main/resources/static`. A good way to ensure its uniqueness is to use the plugin ID as folder name.

In this case the different styles will be available in XperienCentral and can be assigned to pages, elements, etc., individually. When a style is removed by uninstalling the presentation component which publishes the style, only those elements that were specifically assigned to this style are affected. Henceforth, these objects will use the default style instead.

Using One Runtime Style

The opposite approach is to have multiple presentation components each of which overwrites the other. Only one presentation component will be active at a time. In this case the following rules must be followed:

- The name property in the JSP descriptors must be the same for each target presentation type across all styles. For example, the name of the descriptor of the page presentation is the same for all styles, likewise for the name of the descriptor of the text element presentation, etc.
- All static files must be located in the same subfolder and have the same filename.

In this case the complete style of a website can be changed by uploading a new style. However, you must perform the following tasks in the order given:

1. Uninstall the old style by uninstalling the old presentation component. It doesn’t matter whether you purge the content.
2. Install the new style by installing the new presentation component.

Be aware that between the first and second step the website will have no style and so is practically "offline". This only lasts for a short time if you install the new style immediately.

Back to Top

Service Component

A service component is a component that provides a (usually headless) service that can be used by other plugins. Typical services are, for example, an authorization service, license service or preferences service. They provide functionality that is available for all plugins, therefore they are implemented as a service.

The service component created by the archetype as described in the Quick Start guide already provides a simple service component. The only thing you have to do is to implement the service and define the interface that it exposes. Other plugins will be able to define a dependency on this service component and use it.

When creating service components, be sure that they conform to the Plugin Development Guidelines (G083-G098, G112 and G151 in particular).

Back to Top

Form Component

A form component can be used to deploy form handlers into the XperienCentral application without a server restart. The form archetype generates a handler with an empty implementation. To implement the handler the `doHandle` method of the handler must be implemented.

When creating form components, be sure that they conform to the Plugin Development Guidelines (G083-G098 in particular).

Back to Top

Servlet Component

A servlet component is a component that registers a servlet which is mapped to a predefined URL. The syntax of this URL is:

`http://<hostname>:<portnumber>/<context>/wcbservlet/<componentid>`

where `componentid` equals the ID of the component definition. The servlet component definition contains properties that you must pay particular attention to with regard to servlets. These property methods are:

Method	Description
`setName`	Sets the name of the component but sets also the servlet name (returned by `javax.servlet.HttpServlet.getServletName()`.
`setServletClassName`	Sets the class name of the servlet to be instantiated by the servlet component.
`setProperties`	Defines the `init` parameters usually defined by adding `init-param` attributes to the servlet definition in the `web.xml`. These properties can be retrieved from `javax.servlet.HttpServlet.getInitParameter()` and `javax.servlet.HttpServlet.getInitParameterNames`

The `ServletContext` of this servlet is the same as the `ServletContext` of `nl.gx.webmanager.servlet.WCBDispatcherServlet` since this is the servlet that dispatches all incoming HTTP requests to the appropriate component servlets.

When creating servlet components, be sure that they conform to the Plugin Development Guidelines (G083-G098 in particular).