Configuring Camel

Unable to render {include} The included page could not be found.
Unable to render {include} The included page could not be found.

How do I add a component

You might first want to read Writing Components for a background in how to implement a new component.
Typically it means you write an implementation of the Component interface, usually deriving from DefaultComponent.

You can then register your component explicitly via

CamelContext context = new DefaultCamelContext(); context.addComponent("foo", new FooComponent(context));

However you can use the auto-discovery feature of Camel where by Camel will automatically add a Component when an endpoint URI is used. To do this you would create a file called

/META-INF/services/org/apache/camel/component/foo

with contents

class=org.acme.FooComponent

(you can add other property configurations in there too if you like)

Then if you refer to an endpoint as **`foo://somethingOrOther`** Camel will auto-discover your component and register it.

The **`FooComponent`** can then be auto-injected with resources using the Injector, such as to support Spring based auto-wiring, or to support **`@Resource`** (EJB3 style) injection or Guice style **`@Inject`** injection.

Working with Spring XML

You can configure a component via Spring using the following mechanism...{snippet:id=example|lang=xml|url=camel/trunk/components/camel-jms/src/test/resources/org/apache/camel/component/jms/jmsRouteUsingSpring.xml}Which allows you to configure a component using some name (activemq in the above example), then you can refer to the component using **`activemq:[queue:|topic:]destinationName`**.

If you want to add explicit Spring 2.x XML objects to your XML then you could use the **`xbean-spring`** which tries to automate most of the XML binding work for you; or you could look in camel-spring at **`CamelNamespaceHandler`** you'll see how we handle the Spring XML stuff (warning its kinda hairy code to look at 🙂. If you wanted **`<fooComponent>`** to be a standard part of the core Camel schema then you'd hack that file to add your component & conftribute a patch to the camel XSD. Otherwise you could write your own namespace & schema if you prefer.

See Also

- Writing Components
- How Do I Configure Endpoints?