

CREATING A NEW ROUTE

This tutorial walks you through the process of creating a new Fuse project, adding a route to it, and adding two endpoints to the route. It assumes that you have already set up your workspace .

Goals

In this tutorial you will:

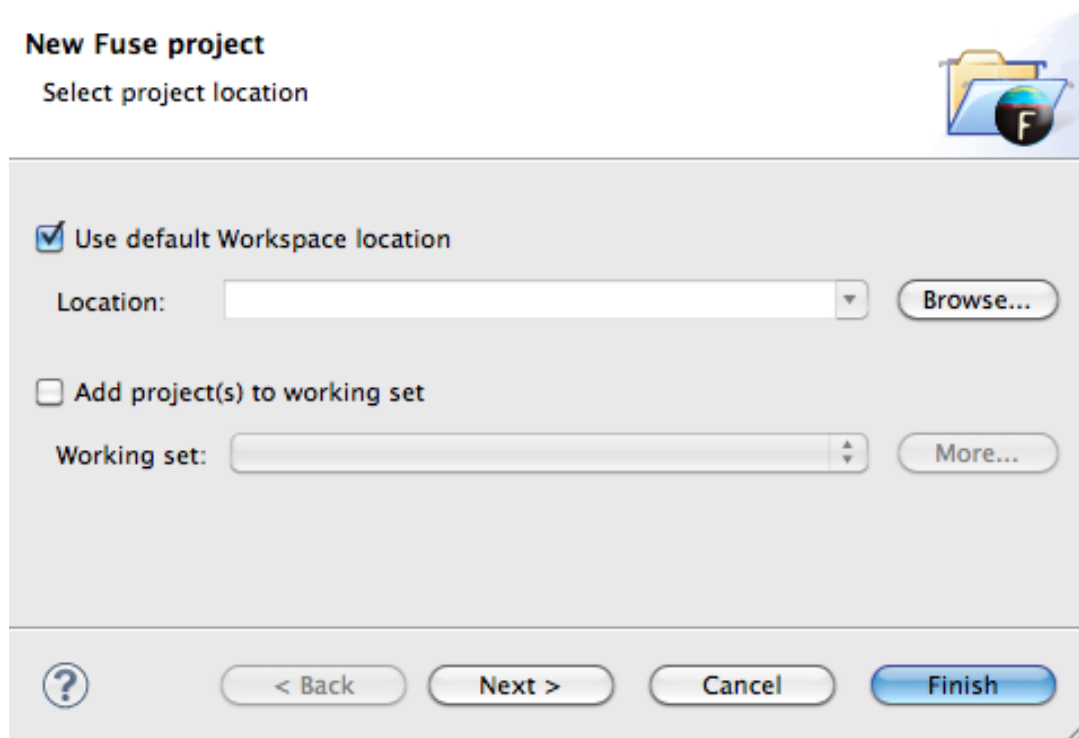
- create a routing project
- add endpoints to a route
- connect two endpoints
- configure the endpoints

Creating the Fuse project

To create a Fuse project:

1. Open the Java perspective.
2. On the Toolbar, select **File|New|Fuse Project** to open the **New Fuse Project** wizard, as shown in [Figure 1](#).

Figure 1. New Fuse Project project location page



3. Click **Next>** to open the **New Fuse Project** details page, as shown in [Figure 2](#).

Figure 2. New Fuse Project details page

New Fuse project

Select a project archetype to create and specify its details



Filter:

Group ID	Artifact ID	Version
org.apache.camel.archetypes	camel-archetype-activemq	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-blueprint	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-component	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-dataformat	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-java	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-spring	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-spring-dm	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-web	2.10.0.redhat-60022
org.apache.camel.archetypes	camel-archetype-webconsole	2.10.0.redhat-60022
org.apache.cxf.archetype	cxf-jaxrs-service	2.6.0.redhat-60022
org.apache.cxf.archetype	cxf-jaxws-javafirst	2.6.0.redhat-60022
org.fusesource.fabric	camel-cxf-code-first-archetype	7.2.0.redhat-022
org.fusesource.fabric	camel-cxf-contract-first-archetype	7.2.0.redhat-022
org.fusesource.fabric	camel-drools-archetype	7.2.0.redhat-022

Creates a new Camel project with added Spring DSL support.

Group Id:

Artifact Id:

Version:

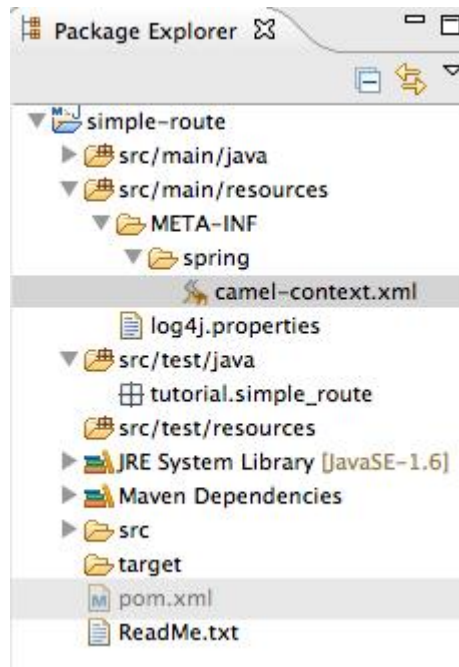
Package:

4. Select `camel-archetype-spring`.
5. Enter `tutorial` in the **Group Id:** field.
6. Enter `simple-route` in the **Artifact Id:** field.
7. The **Version:** field defaults to `1.0.0-SNAPSHOT`. To change it, enter a different version identifier.
8. The **Package:** field defaults to `tutorial.simple.route`, the name of the package that contains `camel-archetype-spring`. To include the route in a different package, enter the name of that package.
9. Click **Finish**.

This procedure creates a Fuse project, `simple-route`, in **Package Explorer** that contains everything needed to create and run routes. As shown in [Figure 3](#), the files generated for `simple-route` include:

- `simple-route/pom.xml` (Maven project file)
- `simple-route/src/main/resources/META-INF/spring/camel-context.xml` (Spring XML file containing the routing rules)

Figure 3. Generated project files



Creating the route

To create the new route:

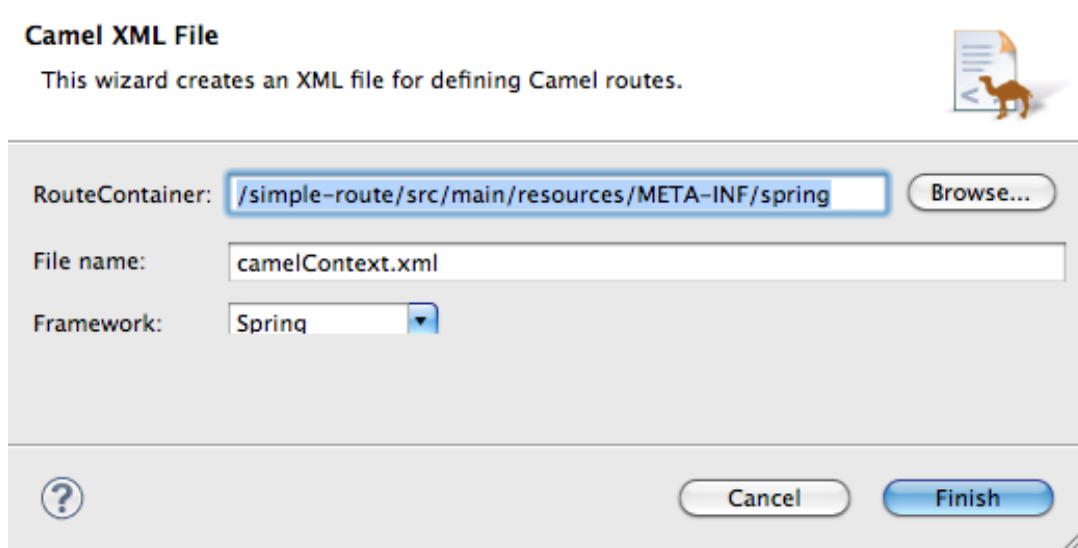
1. In **Package Explorer**, locate `simple-route/src/main/resources/META-INF/spring/camel-context.xml`.
2. Right-click it to open the context menu, then select **Delete**.

You're going to replace the old `camel-context.xml` file with your own to create a new route.

3. In the **Confirm Delete** dialog, click **OK**.
4. In **Package Explorer**, select `simple-route/src/main/resources/META-INF/spring`.
5. Right-click it to open the context menu.

6. Select **New | Camel XML File** to open the **Camel XML File** wizard, as shown in [Figure 4](#).

Figure 4. Camel XML file wizard



7. Check that `/simple-route/src/main/resources/META-INF/spring` appears in the **Container:** field. Otherwise enter it manually, or select it using the **Browse...** button.

Note

The **Browse...** button opens a dialog that displays the folders of all active projects, which you can browse to find and select the files you need.

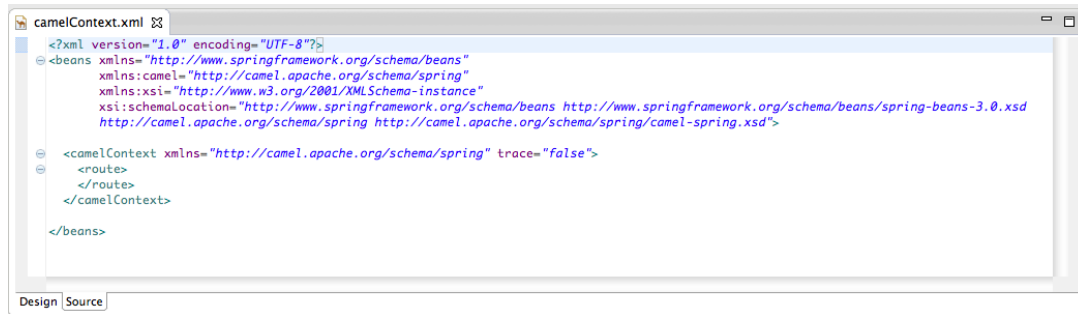
8. Check that `camelContext.xml` appears in the **File Name:** field. Otherwise enter it manually.
9. Check that **Spring** appears in the **Framework** field, or select it from the field's drop-down list.
10. Click **Finish**.


Note

By default, **Outline** view is located in the upper, right corner of the Java perspective. To provide more room for **Design** view to display the graphical representation of your route, drag **Outline** view to the lower, left corner of the workspace, below **Package Explorer**.

11. Click the **Source** tab at the bottom, left of the canvas to open the new `camelContext.xml` file in the route editor's Source view, as shown in Figure 5.

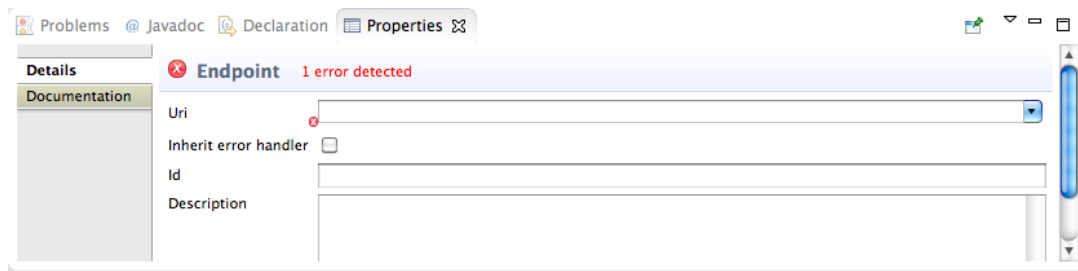
Figure 5. New camelContext file in the route editor's source view



12. Click the **Design** tab at the bottom, left of the canvas to return to the route editor's Design view.
13. Drag an **Endpoint** element () from the **Palette** to the canvas.
14. Drag a second **Endpoint** element from the **Palette** to the canvas.
15. Select the first endpoint you dragged onto the canvas.

The **Properties** editor, located below the canvas, displays the endpoint's property fields for editing, as shown in Figure 6.

Figure 6. Endpoint property editor



16. Enter `file:src/data?noop=true` in the **Uri** field. Leave the other fields blank.
17. Select the second endpoint you dragged onto the canvas.
18. Enter `file:target/messages/others` in the **Uri** field. Leave the other fields blank.

19. On the canvas, select the first endpoint (`file:src/data?noop=true`), and drag its connector arrow (↗) to the second endpoint (`file:target/messages/others`), then release it.

A segmented line connects the two endpoints, as shown in [Figure 7](#).

Figure 7. Completed route, diagram view



Note

You can drag the line's bendpoint (orange dot) to change the angle of the line's segments. Doing so creates two new bendpoints, one on either side of the original. This behavior enables you to easily adjust your diagram to accommodate increasingly complex routes.

20. To quickly align the connected endpoints, right-click the canvas to open the context menu, and then select **Layout Diagram**.
21. Select **File|Save** to save the route.
22. Click the **Source** tab at bottom, left of the canvas.

Source view displays the XML for the route. The `camelContext` element will look like [Example 1](#).

Example 1. XML for simple route

```
<camelContext xmlns="http://camel.apache.org/schema/spring" trace="false">
  <route>
    <from uri="file:src/data?noop=true"/>
    <to uri="file:target/messages/others"/>
  </route>
</camelContext>
```

Next steps

After you have created and designed your route, you can run it by deploying it into your Apache Camel runtime, as described in [Running a Route](#).