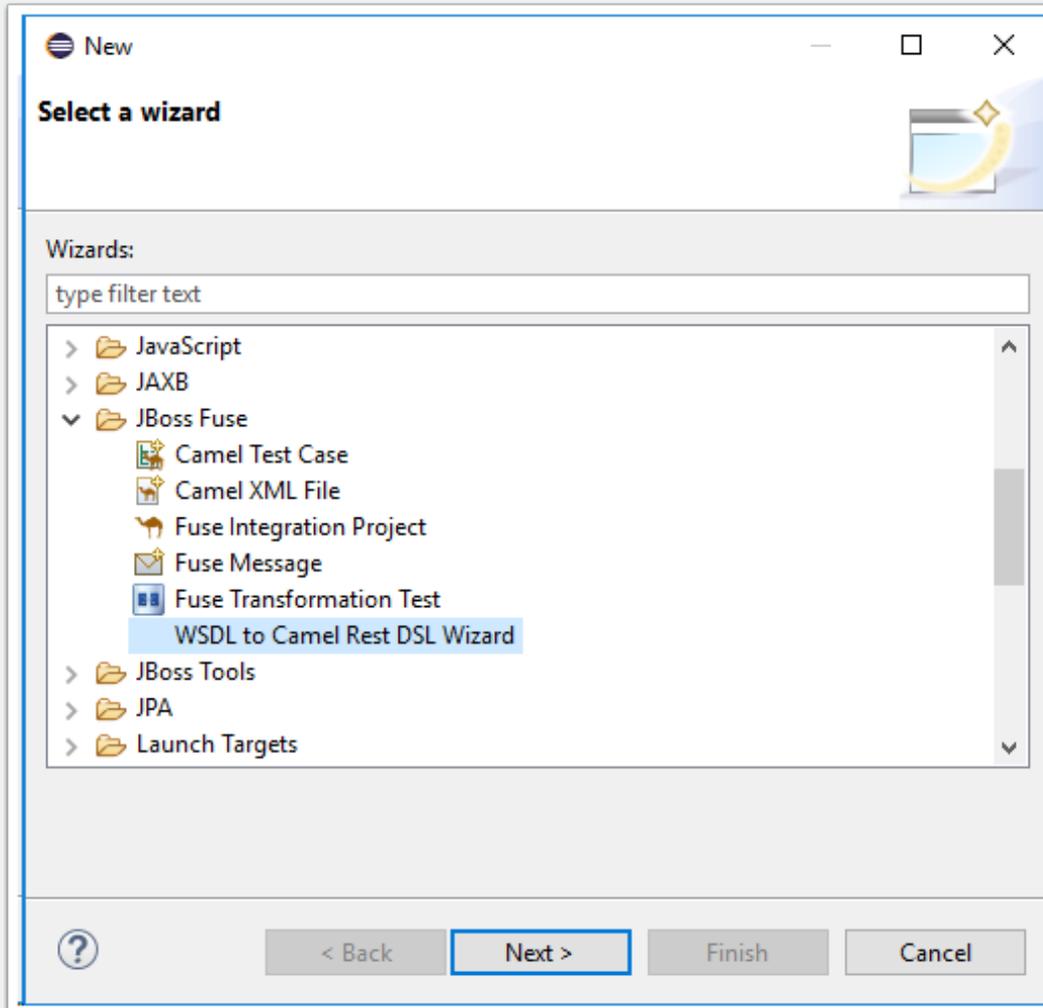


# New wsdl2rest Wizard Mock-up

This document details a potential flow, with mockups, of the UI for using the wsdl2rest tool for adapting a SOAP project into a Camel-based REST project.

## Starting the process

In Eclipse, the user right-clicks on an existing Fuse Integration Project and selects New->Other..., then chooses the JBoss Fuse->WSDL to Camel Rest DSL Wizard.



# New wsdl2rest Wizard Mock-up

## Prototype Wizard

The current prototype wizard offers four basic fields:

- WSDL File URL - can be either a file:// URL (the "..." browse button opens a file dialog) or a web-based URL
- Output Path - specifies where generated artifacts should go. Right now it defaults to the selected fuse project and puts them in a subfolder.
- Target Address - specifies the address for the generated Camel endpoint
- Bean Class - Class name for the bean Camel delegates to.

These correspond to the options available in the wsdl2rest command-line tool

(<https://github.com/jboss-fuse/wsdl2rest>)

# New wsdl2rest Wizard Mock-up

The image shows a mock-up of a wizard window titled "Wsdl-to-Camel Rest DSL Wizard". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is titled "Select Incoming WSDL and Location for Generated Output" and includes the instruction "Provide a URL to your WSDL and select the project and folder for the generated artifacts." Below this, there are four input fields, each with a "..." button to its right: "WSDL File URL" with the value "http://localhost:9292/cxf/order?wsdl", "Output Path" with "c:\jee-oxygen\runtime\fuseproject\path", "Target Address" with "http://localhost:8080/mycontext", and "Bean Class" with "com.demo.order.DemoOrderService". At the bottom of the window, there are four buttons: "< Back", "Next >", "Finish", and "Cancel".

## Revised Wizard, First Page

I propose, based on the meeting we had on 29-NOV-2017, that we rearrange the wizard slightly to simplify it for the user, breaking it into two wizard pages. The first gets the WSDL info and where they want the generated artifacts. The second allows them to customize a bit.

# New wsdl2rest Wizard Mock-up

Essentially on this first page all the user should have to do is provide the WSDL for the existing SOAP service and a destination project for the generated artifacts, so that boils the page down to two fields.

- To specify the WSDL file, the user can type the URL to a web-based service or use the "..." (Browse) button to select a WSDL from the Workbench. This file does not need to reside in the same project as the generated Java classes and Camel REST DSL XML file.
- If the user pre-selected a Fuse Integration Project in the Project Explorer, it should come up automatically specified. But they should have the ability to change the selected Fuse Integration Project.

We can infer the type of project (i.e. Spring/Blueprint) from the project the user specifies as the destination, which then informs the paths we specify by default for the Java code and Camel XML config.

# New wsdl2rest Wizard Mock-up

The image shows a window titled "WsdL-to-Camel Rest DSL Wizard". The window contains the following elements:

- Title Bar:** "WsdL-to-Camel Rest DSL Wizard" with standard window control buttons (minimize, maximize, close).
- Section Header:** "Select Incoming WSDL and Project for Generated Output".
- Instruction:** "Provide a URL to your WSDL and select the project for the generated artifacts."
- WSDL File:** A text input field with a browse button ("...") and a "Validate" button.
- Destination Project:** A text input field containing the text "fuseproject" and a browse button ("...").
- Navigation Buttons:** A row of four buttons at the bottom: "< Back", "Next >", "Finish", and "Cancel".

## Revised Wizard, Second Page

The second page would then allow the user to customize any path or service address details for the generated code and Camel configuration.

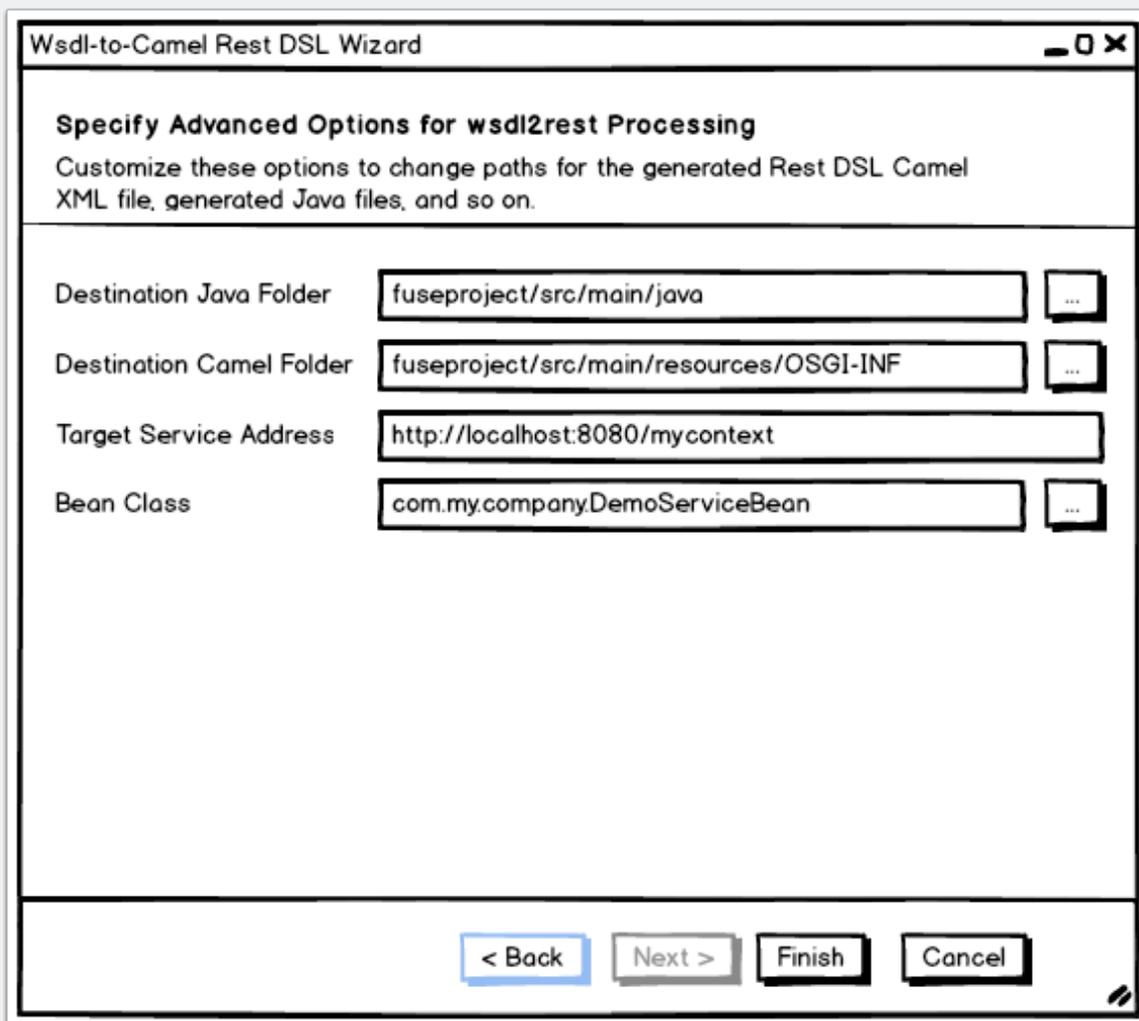
- The folder paths are inferred by the project selected on the first page and the type of project.

# New wsdl2rest Wizard Mock-up

- The Target Service Address and Bean Class should be optional, but can help customize the generated artifacts.

Going back to the first page from here and changing the selected project will change the destination folders. Changing the WSDL may change the Target Service Address.

Note: Currently only one destination path can be specified for the tool to generate the artifacts. It would be good to provide a separate Java vs. Camel config destination to make this play nicer with Fuse projects in the tooling. I have started a conversation with Thomas about this.



The image shows a mock-up of a wizard dialog box titled "Wsdl-to-Camel Rest DSL Wizard". The dialog has a title bar with standard window controls. The main content area is titled "Specify Advanced Options for wsdl2rest Processing" and includes a subtitle: "Customize these options to change paths for the generated Rest DSL Camel XML file, generated Java files, and so on." Below this, there are four labeled text input fields, each with a browse button (three dots) to its right:

- Destination Java Folder:** fuseproject/src/main/java
- Destination Camel Folder:** fuseproject/src/main/resources/OSGI-INF
- Target Service Address:** http://localhost:8080/mycontext
- Bean Class:** com.my.companyDemoServiceBean

At the bottom of the dialog, there are four buttons: "< Back" (highlighted in blue), "Next >", "Finish", and "Cancel".

# New wsdl2rest Wizard Mock-up

## Generated Artifacts

The end result when the user clicks Finish will be a deployable REST DSL XML file (Spring or Blueprint).