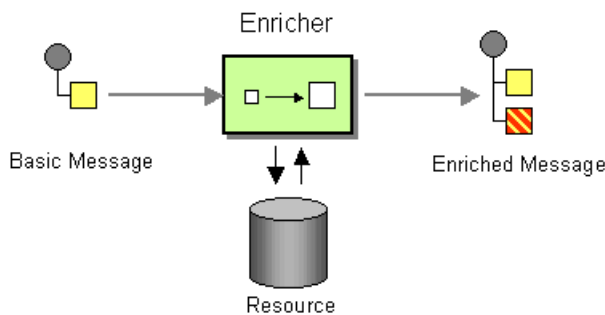# Content Enricher

Overview

The *content enricher* pattern describes a scenario where the message destination requires more data than is present in the original message. In this case, you would use a content enricher to pull in the extra data from an external resource. *REVISIT - NB: It wouldn't hurt to add more examples for different scenarios, as we come across them!*

Figure 23. Content Enricher Pattern

Implementing a content enricher

You can use one of the following approaches to implement a content enricher:

• *Templating*—is a scripting technique that involves extracting portions of a message and inserting them into a given template. FUSE Mediation Router supports templating with several different scripting languages and components. See Templating [http://activemq.apache.org/camel/templating.html] for details.

• *Bean integration*—enables you to call any method on a registered bean. The bean method can modify the message to enrich the content. See Bean integration [http://activemq.apache.org/camel/bean-integration.html] for details.

Java DSL example

You can use templating to consume a message from one destination, transform it with a scripting language, like Velocity or XQuery, and then send it on to another destination. For example, using the *InOnly* exchange pattern (one way messaging):

```
from("activemq:My.Queue").
  to("velocity:com/acme/MyResponse.vm").
  to("activemq:Another.Queue");
```

If you want to use *InOut* (request-reply) semantics to process requests on the My.Queue queue in ActiveMQ with a template-generated response and then send responses back to the JMSReplyTo destination, you could define a route like the following:

```
from("activemq:My.Queue").
  to("velocity:com/acme/MyResponse.vm");
```

For more details about the Velocity component, see Velocity in the *Component Reference*.

XML configuration example

The following example shows how to configure the same route in XML:

```
<camelContext xmlns="http://act
ivemq.apache.org/camel/schema/spring">
  <route>
    <from uri="activemq:My.Queue"/>
    <to uri="velocity:com/acme/MyResponse.vm"/>
    <to uri="activemq:Another.Queue"/>
  </route>
</camelContext>
```