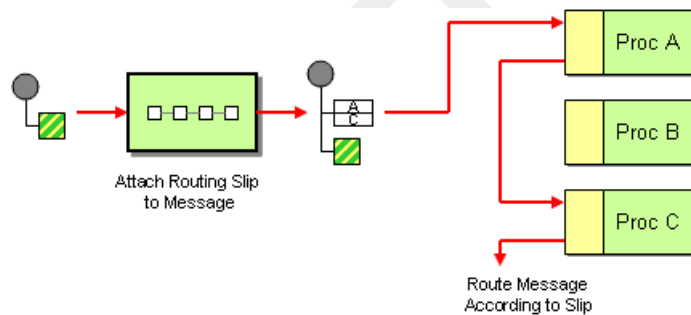


Routing Slip

Overview

The *routing slip* pattern enables you to route a message consecutively through a series of processing steps, where the sequence of steps is not known at design time and can vary for each message. The list of endpoints through which the message should pass is stored in a header field (the *slip*), which Mediation Router reads at run time in order to construct a pipeline on the fly.

Figure 21. Routing Slip Pattern



The slip header

By default the routing slip appears in a header named, `routingSlipHeader`, where the header value is a comma-separated list of endpoint URIs. For example, a routing slip that specifies a sequence of security tasks—decrypting, authenticating, and de-duplicating a message—might look like the following:

```
cxfr:bean:decrypt,cxfr:bean:authenticate,cxfr:bean:dedup
```

Java DSL example

The following route takes messages from the `direct:a` endpoint and passes them into the routing slip pattern:

```
from("direct:a").routingSlip();
```

You can customize the name of the routing slip header by passing a string argument to the `routingSlip()` command, as follows:

```
from("direct:b").routingSlip("aRoutingSlipHeader");
```

You can also customize the URI delimiter using the two-argument form of `routingSlip()`. For example, to customize the routing slip header to be `aRoutingSlipHeader` and to specify `#` as the URI delimiter, define the route as follows:

```
from("direct:c").routingSlip("aRoutingSlipHeader", "#");
```

XML configuration example

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

```
<camelContext id="buildRoutingSlip" xmlns="http://activemq.apache.org/camel/schema/spring">
  <route>
    <from uri="direct:c"/>
    <routingSlip headerName="aRoutingSlipHeader" uriDelimiter="#" />
  </route>
</camelContext>
```