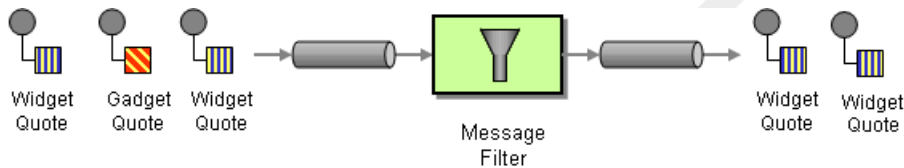


Message Filter

Overview

A *message filter* is a processor that eliminates undesired messages based on specific criteria. In Mediation Router, the message filter pattern is implemented by the `filter()` Java DSL command. The `filter()` command takes a single predicate argument, which controls the filter as follows: when the predicate is `true`, the incoming message is allowed to proceed, and when the predicate is `false`, the incoming message is blocked.

Figure 16. Message Filter Pattern



Java DSL example

The following example shows how to create a route from endpoint, `seda:a`, to endpoint, `seda:b`, that blocks all messages except for those messages whose `foo` header have the value, `bar`,

```
RouteBuilder builder = new RouteBuilder() {
    public void configure() {
        from("seda:a").filter(header("foo").isEqualTo("bar")).to("seda:b");
    }
};
```

To evaluate more complex filter predicates, you can invoke one of the supported scripting languages, such as XPath, XQuery, or SQL (see [Languages for Expressions and Predicates](#) in the *Defining Routes*). For example, to define a route that blocks all messages except for those containing a `person` element whose `name` attribute is equal to `James`:

```
from("direct:start").
    filter().xpath("/person[@name='James']").
    to("mock:result");
```

XML configuration example

The following example shows how to configure the route with a simple predicate in XML:

```
<camelContext id="simpleFilterRoute" xmlns="http://activemq.apache.org/camel/schema/spring">
  <route>
    <from uri="seda:a"/>
    <filter>
      <simple>header.foo = 'bar'</simple>
      <to uri="seda:b"/>
    </filter>
  </route>
</camelContext>
```

The following example shows how to configure the route with an XPath predicate in XML (see [Languages for Expressions and Predicates](#) in the *Defining Routes*):

```
<camelContext id="xpathFilterRoute" xmlns="http://activemq.apache.org/camel/schema/spring">
  <route>
    <from uri="direct:start"/>
    <filter>
      <xpath>/person[@name='James']</xpath>
      <to uri="mock:result"/>
    </filter>
  </route>
</camelContext>
```