

# Content Based Router

## Content Based Router

The [Content Based Router](#) from the [EIP patterns](#) allows you to route messages to the correct destination based on the contents of the message exchanges.

[blocked URL](#)

The following example shows how to route a request from an input **seda:a** endpoint to either **seda:b**, **seda:c** or **seda:d** depending on the evaluation of various [Predicate](#) expressions

### Using the [Fluent Builders](#)

```
RouteBuilder builder = new RouteBuilder() {
    public void configure() {
        errorHandler(deadLetterChannel("mock:error"));

        from("direct:a")
            .choice()
                .when(header("foo").isEqualTo("bar"))
                    .to("direct:b")
                .when(header("foo").isEqualTo("cheese"))
                    .to("direct:c")
                .otherwise()
                    .to("direct:d");
    }
};
```



See [Why can I not use when or otherwise in a Java Camel route](#) if you have problems with the Java DSL, accepting using when or otherwise.

### Using the [Spring XML Extensions](#)

```
<camelContext errorHandlerRef="errorHandler" xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="direct:a"/>
    <choice>
      <when>
        <xpath>$foo = 'bar'</xpath>
        <to uri="direct:b"/>
      </when>
      <when>
        <xpath>$foo = 'cheese'</xpath>
        <to uri="direct:c"/>
      </when>
      <otherwise>
        <to uri="direct:d"/>
      </otherwise>
    </choice>
  </route>
</camelContext>
```

For further examples of this pattern in use you could look at the [junit test case](#)

## Using This Pattern

If you would like to use this EIP Pattern then please read the [Getting Started](#), you may also find the [Architecture](#) useful particularly the description of [Endpoint](#) and [URIs](#). Then you could try out some of the [Examples](#) first before trying this pattern out.