

How do I retry processing a message from a certain point back or an entire route

How do I retry processing a message from a certain point back or an entire route

By default Apache Camel will perform any redelivery (retry) attempts from the point of failure. So if you want to retry from a point before this, you would need to split up your route .

In the example above we have 2 routes (direct:start, direct:sub). In case of a failure anywhere in the direct:sub route, then the entire route is retried. This happens because we have instructed the direct:sub route to not use any error handler (eg the no error handler). Then we link the routes using the [Direct](#) component by calling the sub route from the 1st route.

```
Error formatting macro: snippet: java.lang.IndexOutOfBoundsException: Index: 20, Size: 20
```

The code above is based on an unit test, and as you can see the processor below is configured to fail the first 2 attempts. So that means the entire direct:sub route is redelivered, meaning that the "mock:b" endpoint receives the incoming message again.

```
Error formatting macro: snippet: java.lang.IndexOutOfBoundsException: Index: 20, Size: 20
```

The same example is shown below using the XML DSL:

Error rendering macro 'code': Invalid value specified for parameter 'java.lang.NullPointerException'

```
<!-- this is the processor that will fail the first 2 attempts -->
<bean id="myProcessor" class="org.apache.camel.processor.RedeliverToSubRouteTest.MyProcessor"/>

<camelContext xmlns="http://camel.apache.org/schema/spring">

  <!-- setup no error handler with an id, we refer to from the 2nd route -->
  <errorHandler id="noErrorHandler" type="NoErrorHandler"/>

  <!-- configure on exception to redelivery at most 2 times when an IOException was thrown
  do not use redelivery delay to run unit test faster -->
  <onException>
    <exception>java.io.IOException</exception>
    <redeliveryPolicy maximumRedeliveries="2" redeliveryDelay="0"/>
  </onException>

  <!-- 1st route, no need to setup error handler, as it will use the default error handler -->
  <route>
    <from uri="direct:start"/>
    <to uri="mock:a"/>
    <to uri="direct:sub"/>
    <to uri="mock:c"/>
  </route>

  <!-- disable error handler on this route, so the entire route can be redelivered
  when called from the 1st route -->
  <route errorHandlerRef="noErrorHandler">
    <from uri="direct:sub"/>
    <to uri="mock:b"/>
    <process ref="myProcessor"/>
  </route>
</camelContext>
```