CXF Proxy Example

CXF Proxy Example

This example is located in the `examples/camel-example-cxf-proxy` directory of the Camel distribution.
There is a `README.txt` file with instructions how to run it.

If you use Maven then you can easily run it from the command line using:

mvn camel:run

About

This example demonstrates how CXF can be used to proxy a real web service. For this example we want Camel to validate and enrich the input message before it's sent to the actual web service. We do this to ensure the data is correct and to have Camel automatically add in any missing information. In real life you may wish to use a proxy in cases where clients are sending bad/faulty data to a web service, and it's costly/not possible to update and fix those clients.

Pure HTTP proxy
If you want a pure HTTP-based proxy then you can use Jetty or Servlet as the front end in Camel. Then delegate those incoming HTTP requests to the real web service. This is for cases where you need a simple straight through proxy where the message should not be altered by Camel. By using the CXF component you get access to all the web service features Apache CXF provides, whereas Jetty will provide just a pure HTTP proxy.

Implementation

For this simple example both the Camel application and the real web service are in the same JVM. In production work the real web service may often be hosted on another server, etc.

Spring XML

In the Spring XML file we have defined the CXF proxy endpoint using the `<cxf:cxfEndpoint>` tag. The real web service is the Spring bean with the id `realWebService`.

As you can see in the Camel route we use a CXF consumer to proxy the web service. Then we route the message to the EnrichBean which validates and adds the missing information. Then we just use the HTTP component to send the web service request to the real web service.
The reply from the real web service is then logged and used as a reply for the proxied web service as well.

{snippet:id=e1|lang=xml|url=camel/trunk/examples/camel-example-cxf-proxy/src/main/resources/META-INF/spring/camel-config.xml}

Enrich bean

The enrich bean is Java code which in our simple example will just set the incidentId parameter to the fixed value of 456. In your implementation you can of course do a lot more.

{snippet:id=e1|lang=java|url=camel/trunk/examples/camel-example-cxf-proxy/src/main/java/org/apache/camel/example/cxf/proxy/EnrichBean.java}

Running the example

You start the example from the command line using the Maven goal `mvn camel:run`.
You can also start the Camel application from your IDE, by running the `org.apache.camel.example.cxf.proxy.MyMain` main class.

You can then use SoapUI or another web service client and send a request to the `http://localhost:9080/camel-example-cxf-proxy/webservices/incident` url. The wsdl is located at: `http://localhost:9080/camel-example-cxf-proxy/webservices/incident?wsdl`.

After making a SOAP request, check the console to see the SOAP request and response:

Sample output

Here is a sample output from the console:

2010-09-26 12:20:46,974 [main] INFO DefaultCamelContext - Apache Camel 2.5-SNAPSHOT (CamelContext: camel-1) started in 0.858 seconds 2010-09-26 12:20:55,685 [tp-1790017034-1] INFO input - Exchange[ExchangePattern:InOut, BodyType:null, Body: <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:rep="http://reportincident.example.camel.apache.org"> <soapenv:Header/> <soapenv:Body> <rep:inputReportIncident> <incidentId>63</incidentId> <incidentDate>2010-09-26</incidentDate> <givenName>Claus</givenName> <familyName>Ibsen</familyName> <summary>Bla bla</summary> <details>More bla</details> <email>davsclaus@apache.org</email> <phone>12345678</phone> </rep:inputReportIncident> </soapenv:Body> </soapenv:Envelope>] Incident was 63, changed to 456 Invoked real web service: id=456 by Claus Ibsen 2010-09-26 12:20:55,997 [tp-1790017034-1] INFO output - Exchange[ExchangePattern:InOut, BodyType:org.apache.camel.converter.stream.CachedOutputStream.WrappedInputStream, Body: <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns2:outputReportIncident xmlns:ns2="http://reportincident.example.camel.apache.org"> <code>OK;456</code> </ns2:outputReportIncident> </soap:Body> </soap:Envelope>]

See Also

- Examples

- CXF
- HTTP
- Jetty
- Servlet
- SOAP