CXF Example OSGi

CXF Example OSGi

Available as of Camel 2.8

> **Spring-DM vs. OSGi Blueprint**
>
> This example uses Spring DM for OSGi. There is another CXF Example OSGi Blueprint that uses Blueprint.

A simple example which receives web service calls (via a CXF consumer, using bean binding) and writes these requests to the file system. It's not a very useful use case, but the goal of this example is to show you how you can use the CXF consumer component in an OSGI environment with the OSGI HTTP service. If your target container is Apache Karaf or Apache ServiceMix, you can use PAX Web to setup and start an OSGI HTTP service. All Camel bundles using a Camel CXF consumer can use this HTTP service without needing to start individual Jetty instances. Another advantage is that all provided services can now share the same port.

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

You will need to compile the example first:

```
mvn install
```

Remarks:

- During the compilation phase, a unit test will be performed which simulates the client calling the web service exposed by our camel/cxf route.
- In Eclipse, I have used the following option when starting the junit test case. This option tells CXF that it must use log4j : -Dorg.apache.cxf.Logger=org.apache.cxf.common.logging.Log4jLogger

To run the example on Apache ServiceMix 4.x or Apache Karaf 2.2.x

1) launch the server

```
karaf.bat
```

Note for Karaf 2.2.x:
a) edit the etc/jre.properties file to add the following packages to be exported

```
jre-1.6=, \
com.sun.org.apache.xerces.internal.dom, \
com.sun.org.apache.xerces.internal.jaxp, \
```

b) from the same file comment out the following exports already provided by the bundles that will be imported next: javax.xml.bind*, javax.jws*, javax.xml.soap*, javax.xml.ws*, javax.activation, javax.annotation, javax.xml.stream*.

2) Add features required

```
features:addUrl mvn:org.apache.camel.karaf/apache-camel/2.9.0/xml/features
features:install war
features:install cxf
features:install camel-spring
features:install camel-jaxb
features:install camel-cxf
```

Note: Apache Camel 2.9.0 is being used above, but you should of course change the version number to the exact version of Camel being used.

3) Deploy the example bundle

```
osgi:install -s mvn:org.apache.camel/camel-example-cxf-osgi/2.9.0
```

4) Verify that your service is available using the following url in the browser.
We assume assuming the OOTB Karaf defaults you use the default PAX Web configuration which use the port 8181 for http. If you would like to use another port or https, change the configuration in ${KARAF_HOME}/etc/org.ops4j.pax.web.cfg. The immediate extension after the hostname and port ("cxf" in the below URL) is configured via the org.apache.cxf.osgi.cfg file (Please see http://team.ops4j.org/wiki//display/paxweb/Pax+Web for more information on PAX Web).

```
http://localhost:8181/cxf/camel-example-cxf-osgi/webservices/incident?wsdl
```

5) Start SOAPUI or use curl to send the request
Create a new project called camel-example-cxf-osgi
Point to the following url: http://localhost:8181/cxf/camel-example-cxf-osgi/webservices/incident?wsdl
Open the request 1 (under camel-example-cxf-osgi --> ReportIncidentBinding --> ReportIncident) and copy/paste a SOAP message generated by the unit test, for example:

--> and the message formatted that you copy in SOAPUI and send the message

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Header />
   <soap:Body>
      <ns2:inputReportIncident xmlns:ns2="http://reportincident.example.camel.apache.org">
         <incidentId>111</incidentId>
         <incidentDate>2011-03-05</incidentDate>
         <givenName>Christian</givenName>
         <familyName>Mueller</familyName>
         <summary>Bla</summary>
         <details>Bla bla</details>
         <email>cmueller@apache.org</email>
         <phone>0049 69 1234567</phone>
      </ns2:inputReportIncident>
   </soap:Body>
</soap:Envelope>
```

If you use curl, you need to save the upper message in a file (data.xml) first and use the blow command to send the message.

```
curl -X POST -T data.xml -H "Content-Type: text/xml" http://localhost:8181/cxf/camel-example-cxf-osgi
/webservices/incident
```

The response message looks like this

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
   <ns2:outputReportIncident xmlns:ns2="http://reportincident.example.camel.apache.org">
     <code>Accepted</code>
   </ns2:outputReportIncident>
  </soap:Body>
</soap:Envelope>
```

6) Check the file system
Check the folder "target/inbox/" in the Karaf base directory to see that a message has arrived.

See Also

- CXF Example OSGi Blueprint
- Examples