Netty HTTP Server Example

Netty HTTP Server Example

Available as of Camel 2.12

This example is located in the `examples/camel-example-netty-http` 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 package the example from the command line:

```
mvn package
```

About

This example shows how to use a shared Netty HTTP Server in an OSGi environment.

There is 3 modules in this example

- shared-netty-http-server - The Shared Netty HTTP server that the other Camel applications uses.
- myapp-one - A Camel application that reuses the shared Netty HTTP server
- myapp-two - A Camel application that reuses the shared Netty HTTP server

Implementation

In the `shared-netty-http-server/src/main/resources/OSGI-INF/blueprint/http-server.xml` file we have a OSGi Blueprint XML file that defines the shared Netty HTTP server we are using. First we need to configure the options on the shared Netty HTTP server which is done using the `NettySharedHttpServerBootstrapConfiguration` class in the configuration bean. In this example we use port 8888 as the shared port number.

Then we define the shared Netty HTTP server using the `DefaultNettySharedHttpServer` class in the httpServer bean.

And finally we need to enlist the shared Netty HTTP server in the OSGi Service Registry, so we can refer and use it from other bundles.

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="
           http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd">

  <!-- netty http bootstrap configuration -->
  <bean id="configuration" class="org.apache.camel.component.netty.http.
NettySharedHttpServerBootstrapConfiguration">
     <!-- the port and host is mandatory and must be set -->
    <property name="port" value="8888"/>
    <property name="host" value="0.0.0.0"/>
    <!-- additional options -->
    <property name="backlog" value="50"/>
  </bean>

  <!-- the netty http server -->
  <bean id="httpServer" class="org.apache.camel.component.netty.http.DefaultNettySharedHttpServer"
        init-method="start" destroy-method="stop">
    <property name="nettyServerBootstrapConfiguration" ref="configuration"/>
  </bean>

  <!-- export in the OSGi server registry so we can use it from the Camel application bundles -->
  <service ref="httpServer" interface="org.apache.camel.component.netty.http.NettySharedHttpServer"/>

</blueprint>
```

The Camel route

In the two Camel applications, each have a Camel route that uses the shared Netty HTTP server. The Camel application is defined in an OSGi blueprint file, for example from myapp-one its the `myapp-one/src/main/resources/OSGI-INF/blueprint/camel-one.xml` file.

First we need to refer to the shared Netty HTTP server which was enlisted in the OSGi service registry using the reference tag as shown below.

In the Camel route, we then use the `nettySharedHttpServer` option to use the shared server, with `nettySharedHttpServer=#sharedNettyHttpServer`.

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="
           http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd">

  <!-- reference the shared http server -->
  <reference id="sharedNettyHttpServer" interface="org.apache.camel.component.netty.http.NettySharedHttpServer"
/>

  <!-- Camel application which uses the netty-http component and the shared Netty HTTP server -->
  <camelContext xmlns="http://camel.apache.org/schema/blueprint">

    <route id="http-route-one">
      <from uri="netty-http:http://localhost/one?matchOnUriPrefix=true&amp;
nettySharedHttpServer=#sharedNettyHttpServer"/>
      <transform>
        <simple>Response from Camel one using thread: ${threadName}</simple>
      </transform>
    </route>

  </camelContext>

</blueprint>
```

Must use unique context-path in Camel routes

When using the `nettySharedHttpServer` option in Camel routes, then each context-path must be **unique**. For example in this example we have 2 Camel applications, where they use unique context-path names:

- /one
- /two

This is because the shared Netty HTTP server needs to know exactly which Camel application that should route the incoming message. And therefore the context-path must be unique among all the Camel routes.

Running the example

This example runs in Apache Karaf / ServiceMix container.

To install Apache Camel in Karaf you type in the shell (we use version 2.12.0):

```
  features:chooseurl camel 2.12.0
  features:install camel
```

First you need to install the following features in Karaf/ServiceMix with:

```
  features:install camel-netty-http
```

In the Apache Karaf / ServiceMix shell type:

```
  osgi:install -s mvn:org.apache.camel/camel-example-netty-http-shared/2.12.0
```

Then you can install the Camel applications:

```
osgi:install -s mvn:org.apache.camel/camel-example-netty-myapp-one/2.12.0
osgi:install -s mvn:org.apache.camel/camel-example-netty-myapp-two/2.12.0
```

From a web browser you can then try the example by accessing the followign URLs:

```
http://localhost:8888/one
http://localhost:8888/two
```

See Also

- Examples
- Netty HTTP