Cometd

Cometd Component

The **cometd:** component is a transport for working with the [jetty](#) implementation of the [cometd/bayeux protocol](#).
Using this component in combination with the dojo toolkit library it's possible to push Camel messages directly into the browser using an AJAX based mechanism.

Maven users will need to add the following dependency to their `pom.xml` for this component:

```
<dependency>
    <groupId>org.apache.camel</groupId>
    <artifactId>camel-cometd</artifactId>
    <version>x.x.x</version>
    <!-- use the same version as your Camel core version -->
</dependency>
```

URI format

```
cometd://host:port/channelName[?options]
```

The **channelName** represents a topic that can be subscribed to by the Camel endpoints.

Examples

```
cometd://localhost:8080/service/mychannel
cometds://localhost:8443/service/mychannel
```

where `cometds:` represents an SSL configured endpoint.

Options

Name	Default Value	Description
resourceBase		The root directory for the web resources or classpath. Use the protocol file: or classpath: depending if you want that the component loads the resource from file system or classpath. Classpath is required for OSGI deployment where the resources are packaged in the jar. Notice this option has been renamed to `baseResource` from **Camel 2.7** onwards.
baseResource		**Camel 2.7:** The root directory for the web resources or classpath. Use the protocol file: or classpath: depending if you want that the component loads the resource from file system or classpath. Classpath is required for OSGI deployment where the resources are packaged in the jar
timeout	240000	The server side poll timeout in milliseconds. This is how long the server will hold a reconnect request before responding.
interval	0	The client side poll timeout in milliseconds. How long a client will wait between reconnects
maxInterval	30000	The max client side poll timeout in milliseconds. A client will be removed if a connection is not received in this time.
multiFrameInterval	1500	The client side poll timeout, if multiple connections are detected from the same browser.
jsonCommented	true	If `true`, the server will accept JSON wrapped in a comment and will generate JSON wrapped in a comment. This is a defence against Ajax Hijacking.
logLevel	1	0=none, 1=info, 2=debug.
crossOriginFilterOn	false	**Camel 2.10:** If `true`, the server will support for cross-domain filtering
allowedOrigins	*	**Camel 2.10:** The origins domain that support to cross, if the `crosssOriginFilterOn` is `true`

filterP ath		**Camel 2.10:** The filterPath will be used by the CrossOriginFilter, if the `crosssOriginFilterOn` is `true`
disconn ectLoca lSession		**Camel 2.10.5/2.11.1: (Producer only)**: Whether to disconnect local sessions after publishing a message to its channel. Disconnecting local session is needed as they are not swept by default by CometD, and therefore you can run out of memory. In Camel 2.16.1/2.15.5 or older the default value is true. From newer versions the default value is false.

You can append query options to the URI in the following format, `?option=value&option=value&...`

Here is some examples on How to pass the parameters

For file (for webapp resources located in the Web Application directory --> cometd://localhost:8080?resourceBase=file./webapp
For classpath (when by example the web resources are packaged inside the webapp folder --> cometd://localhost:8080?resourceBase=classpath:webapp

Authentication

Available as of Camel 2.8

You can configure custom `SecurityPolicy` and `Extension`'s to the `CometdComponent` which allows you to use authentication as documented here

Setting up SSL for Cometd Component

Using the JSSE Configuration Utility

As of Camel 2.9, the Cometd component supports SSL/TLS configuration through the Camel JSSE Configuration Utility. This utility greatly decreases the amount of component specific code you need to write and is configurable at the endpoint and component levels. The following examples demonstrate how to use the utility with the Cometd component. You need to configure SSL on the CometdComponent.

Programmatic configuration of the component

```
KeyStoreParameters ksp = new KeyStoreParameters();
ksp.setResource("/users/home/server/keystore.jks");
ksp.setPassword("keystorePassword");

KeyManagersParameters kmp = new KeyManagersParameters();
kmp.setKeyStore(ksp);
kmp.setKeyPassword("keyPassword");

TrustManagersParameters tmp = new TrustManagersParameters();
tmp.setKeyStore(ksp);

SSLContextParameters scp = new SSLContextParameters();
scp.setKeyManagers(kmp);
scp.setTrustManagers(tmp);

CometdComponent commetdComponent = getContext().getComponent("cometds", CometdComponent.class);
commetdComponent.setSslContextParameters(scp);
```

Spring DSL based configuration of endpoint

```
...
  <camel:sslContextParameters
      id="sslContextParameters">
    <camel:keyManagers
        keyPassword="keyPassword">
      <camel:keyStore
          resource="/users/home/server/keystore.jks"
          password="keystorePassword"/>
    </camel:keyManagers>
    <camel:trustManagers>
      <camel:keyStore
          resource="/users/home/server/keystore.jks"
          password="keystorePassword"/>
    </camel:keyManagers>
  </camel:sslContextParameters>...

  <bean id="cometd" class="org.apache.camel.component.cometd.CometdComponent">
    <property name="sslContextParameters" ref="sslContextParameters"/>
  </bean>
...
  <to uri="cometds://127.0.0.1:443/service/test?baseResource=file:./target/test-classes
/webapp&timeout=240000&interval=0&maxInterval=30000&multiFrameInterval=1500&jsonCommented=true&logLevel=2"/>...
```

See Also

- Configuring Camel
- Component
- Endpoint
- Getting Started