Lumberjack

Camel Lumberjack Component

Available as of Camel 2.18

The **Lumberjack** component retrieves logs sent over the network using the Lumberjack protocol, from Filebeat for instance. The network communication can be secured with SSL.

This component only supports consumer endpoints.

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

pom.xml

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

URI format

```
lumberjack:host
lumberjack:host:port
```

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

Options

The Lumberjack component has no options.

The Lumberjack component supports 7 endpoint options which are listed below:

Name	Group	Default	Java Type	Description
host	consumer		`String`	**Required** Network interface on which to listen for Lumberjack
port	consumer	`5044`	`int`	Network port on which to listen for Lumberjack
bridge ErrorH andler	consumer	`false`	`boolean`	Allows for bridging the consumer to the Camel routing Error Handler which mean any exceptions occurred while the consumer is trying to pickup incoming messages or the likes will now be processed as a message and handled by the routing Error Handler. By default the consumer will use the `org.apache.camel.spi.ExceptionHandler` to deal with exceptions that will be logged at WARN/ERROR level and ignored.
sslCon textPa ramete rs	consumer		`SSLCon textPa ramete rs`	SSL configuration
except ionHan dler	consume r (advance d)		`Except ionHan dler`	To let the consumer use a custom `ExceptionHandler`. Notice if the option `bridgeErrorHandler` is enabled then this options is not in use. By default the consumer will deal with exceptions that will be logged at WARN/ERROR level and ignored.
exchan gePatt ern	advanced	`InOnly`	`Exchan gePatt ern`	Sets the default exchange pattern when creating an exchange
synchr onous	advanced	`false`	`boolean`	Sets whether synchronous processing should be strictly used or Camel is allowed to use asynchronous processing (if supported).

Result

The result body is a `Map<String, Object>` object.

Lumberjack Usage Samples

Example 1: Streaming the log messages

```
RouteBuilder builder = new RouteBuilder() {
    public void configure() {
        from("lumberjack:0.0.0.0").              // Listen on all network interfaces using the default port
            setBody(simple("${body[message]}")).  // Select only the log message
            to("stream:out");                      // Write it into the output stream
    }
};
```