AHC

Async Http Client (AHC) Component

Available from Camel 2.8

The **ahc** component provides HTTP based endpoints for consuming external HTTP resources (as a client to call external servers using HTTP). The component uses the Async Http Client library.

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

xml

URI format

The default ports are **`80`** for HTTP and **`443`** for HTTPS.

Query options can be specified on the URI using the following format: **`?option=value&option=value&...`**

AhcEndpoint Options

confluenceTableSmall

Name	Default Value	Description
`binding`	`null`	To use a custom **`org.apache.camel.component.ahc.AhcBinding`**.
`bridgeEndpoint`	`false`	If the option is **`true`**, then the **`Exchange.HTTP_URI`** header is ignored, and use the endpoint's URI for request. You may also set the **`throwExcpetionOnFailure`** to be false to let the **`AhcProducer`** send all the fault response back.
`bufferSize`	`4096`	**Camel 2.10.3:** The initial in-memory buffer size used when transferring data between Camel and the **`ahc`** Client.
`client`	`null`	To use a custom **`com.ning.http.client.AsyncHttpClient`**.
`clientConfig`	`null`	To configure the **`AsyncHttpClient`** to use a custom **`com.ning.http.client.AsyncHttpClientConfig`** instance. This instance replaces any instance configured at the component level.
`clientConfig.x`	`null`	To configure additional properties of the **`com.ning.http.client.AsyncHttpClientConfig`** instance used by the endpoint. **Note**: that configuration options set using this parameter will be merged with those set using the **`clientConfig`** parameter or the instance set at the component level with properties set using this parameter taking priority.
`clientConfig.realm.x`	`null`	**Camel 2.11:** To configure realm properties of the **`com.ning.http.client.AsyncHttpClientConfig`** The options which can be used are the options from **`com.ning.http.client.Realm.RealmBuilder`**. e.g., to set the scheme, you should configure **`clientConfig.realm.scheme=DIGEST`**
`connectionClose`	`false`	**Camel 2.18:** If this option is **`true`**, the producer will add a Connection Close header to HTTP Request
`cookieHandler`	`null`	**Camel 2.19:** Configure a cookie handler to maintain a HTTP session
`sslContextParameters`	`null`	**Camel 2.9:** Reference to a **`org.apache.camel.util.jsse.SSLContextParameters`** in the Registry. This reference overrides any configured **`SSLContextParameters`** at the component level. See Using the JSSE Configuration Utility. **Note**: configuring this option will override any SSL/TLS configuration options provided through the **`clientConfig`** option at the endpoint or component level.
`throwExceptionOnFailure`	`true`	Option to disable throwing the **`AhcOperationFailedException`** in case of failed responses from the remote server. This allows you to get all responses regardless of the HTTP status code.
`transferException`	`false`	If enabled and an Exchange failed processing on the consumer side, and if the caused **`Exception`** was send back serialized in the response as a **`application/x-java-serialized-object`** content type (for example using Jetty or SERVLET Camel components). On the producer side the exception will be deserialized and thrown as is, instead of the **`AhcOperationFailedException`**. The caused exception is required to be serialized.

AhcComponent Options

confluenceTableSmall

Name	Default Value	Description
`binding`	`null`	To use a custom **`org.apache.camel.component.ahc.AhcBinding`**.
`client`	`null`	To use a custom **`com.ning.http.client.AsyncHttpClient`**.
`clientConfig`	`null`	To configure the **`AsyncHttpClient`** to use a custom **`com.ning.http.client.AsyncHttpClientConfig`**.
`sslContextParameters`	`null`	**Camel 2.9:** To configure custom SSL/TLS configuration options at the component level. See Using the JSSE Configuration Utility for more details. **Note**: configuring this option will override any SSL/TLS configuration options provided through the **`clientConfig`** option at the endpoint or component level.

Note: setting any of the options on the **`AhcComponent`** will propagate those options to the **`AhcEndpoint`**(s) being created. However, the **`AhcEndpoint`** can also configure/override a custom option. Options set on endpoints will always take precedence over options from the **`AhcComponent.`**

Message Headers

confluenceTableSmall

Name	Type	Description
`Exchange.CONTENT_ENCODING`	`String`	The HTTP content encoding. Is set on both the **`IN`** and **`OUT`** message to provide a content encoding, such as **`gzip`**.
`Exchange.CONTENT_TYPE`	`String`	The HTTP content type. Is set on both the **`IN`** and **`OUT`** message to provide a content type, such as **`text/html`**.
`Exchange.HTTP_CHARACTER_ENCODING`	`String`	Character encoding.
`Exchange.HTTP_PATH`	`String`	Request URI's path, the header will be used to build the request URI with the **`HTTP_URI`**. However, if the path is start with **`"/"`**, the HTTP producer will try to find the relative path based on the **`Exchange.HTTP_BASE_URI`** header or the **`exchange.getFromEndpoint().getEndpointUri();`**
`Exchange.HTTP_QUERY`	`String`	**Camel 2.11**: URI parameters. Will override existing URI parameters set directly on the endpoint.
`Exchange.HTTP_RESPONSE_CODE`	`int`	The HTTP response code from the external server. Is **`200`** for **`OK`**.
`Exchange.HTTP_URI`	`String`	URI to call. Will override existing URI set directly on the endpoint.

Message Body

Camel will store the HTTP response from the external server on the **`OUT`** body. All headers from the **`IN`** message will be copied to the **`OUT`** message, so headers are preserved during routing. Additionally Camel will add the HTTP response headers as well to the **`OUT`** message headers.

Response Codes

The value of the HTTP response code governs how Camel behaves:

- If in the range **`100..299`** Camel will regard it as a successful response.
- If in the range **`300..399`** Camel will regard it as a redirection response and throw an **`AhcOperationFailedException`** containing the relevant information.
- If greater than **`400`** Camel regards it as an external server failure and throw an **`AhcOperationFailedException`** containing the relevant information.

  throwExceptionOnFailure
  The option, **`throwExceptionOnFailure`**, can be set to **`false`** to prevent the **`AhcOperationFailedException`** from being thrown for failed response codes. This allows you to get any response from the remote server.

`AhcOperationFailedException` Details

This exception contains the following information:

- The HTTP status code.
- The HTTP status line (text of the status code).

- Redirect location, if server returned a redirect.
- Response body as a **`java.lang.String`**, if the server provides a response body.

Calling using `GET` or `POST`

The following algorithm is used to determine if either **`GET`** or **`POST`** HTTP method should be used:

1. Use method provided in header.
2. **`GET`** if query string is provided in header.
3. **`GET`** if endpoint is configured with a query string.
4. **`POST`** if there is data to send (body is not **`null`**).
5. **`GET`** otherwise**.**

Configuring the URI to Call

You can set the HTTP producer's URI directly form the endpoint URI. In the route below, Camel will call out to the external server, **`oldhost`**, using HTTP.

Java DSL:

java

Spring XML DSL:

xml

You can override the HTTP endpoint URI by adding a header with the key **`Exchange.HTTP_URI`** on the message.

java

Configuring URI Parameters

The **ahc** producer supports URI parameters to be sent to the HTTP server. The URI parameters can either be set directly on the endpoint URI or as a header with the key **`Exchange.HTTP_QUERY`** on the message.

java

Or options provided via a header:

java

How to Set the HTTP Method (`GET`/`POST`/`PUT`/`DELETE`/`HEAD`/`OPTIONS`/`TRACE`) to the HTTP Producer

The HTTP component provides a way to set the HTTP request method by setting the message header.

Example

Java DSL:

java

Spring XML DSL:

xml

Configuring `charset`

If you are using **`POST`** to send data you can configure the **`charset`** using the **`Exchange`** property:

java

URI Parameters From the Endpoint URI

In this example we have the complete URI endpoint that is just what you would have typed in a web browser. Multiple URI parameters can of course be set using the **&** character as separator, just as you would in the web browser. Camel does no tricks here.

java

URI Parameters From the Message

java

In the header value above notice that it should **not** be prefixed with **?** and you can separate parameters as usual with the **&** char.

Getting the Response Code

You can get the HTTP response code from the **`ahc`** component by getting the value from the **`OUT`** message header with **`Exchange.HTTP_RESPONSE_CODE`**.

java

Configuring AsyncHttpClient

The **`AsyncHttpClient`** client uses a **`AsyncHttpClientConfig`** to configure the client. See the documentation at Async Http Client for more details.

In Camel **2.8**, configuration is limited to using the builder pattern provided by **`AsyncHttpClientConfig.Builder`**. In Camel **2.8**, the **`AsyncHttpClientConfig`** doesn't support getters/setters so its not easy to create/configure using a Spring bean style e.g., the **`<bean>`** tag in the XML file.

The example below shows how to use a builder to create the **`AsyncHttpClientConfig`** which we configure on the **`AhcComponent`**.javaINLINEIn Camel **2.9**, the **`ahc`** component uses `Async HTTP library 1.6.4`. This newer version provides added support for plain bean style configuration. The **`AsyncHttpClientConfigBean`** class provides getters and setters for the configuration options available in **`AsyncHttpClientConfig`**. An instance of **`AsyncHttpClientConfigBean`** may be passed directly to the **`ahc`** component or referenced in an endpoint URI using the **`clientConfig`** URI parameter.

Also available in Camel **2.9** is the ability to set configuration options directly in the URI. URI options starting with **`clientConfig`** can be used to set the various configurable properties of **`AsyncHttpClientConfig`**. Options specified in the endpoint URI are merged with those specified by the **`clientConfig`** option. These options take precedence over the options specified on the URI endpoint. A copy of the **`AsyncHttpClientConfig`** is made for each new endpoint. The example below shows how to configure the **`ahc`** component using the **`clientConfig`** URI options.

java

SSL Support (HTTPS)

Using the JSSE Configuration Utility

From **Camel 2.9**, the **`ahc`** 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 **`ahc`** component.

Programmatic configuration of the component
java

Spring DSL based configuration of endpoint
xml

Endpoint See Also

- Jetty
- HTTP
- HTTP4