HTTP 1.0

This topic describes configuration options for using the HTTP 1.0 transport. In this section:

General

The options available under the General tab vary depending on the tool you are configuring.

Endpoint

The Endpoint option is available for SOAP Clients. Choose a definition from the drop-down menu:

- **WSDL**: Specifies the endpoint defined in the WSDL.
- **Default:** Specifies the endpoint defined in the Test or Action Suite.
- **Custom:** Allows you to set any custom endpoint.
- **UDDI serviceKey:** Describes what UDDI serviceKey is used to reference this server endpoint in the UDDI registry specified in the Preferences panel's WSDL/UDDI tab.

Router Endpoint

The Router Endpoint option is available for Messaging Clients. It specifies the endpoint URL for the server. Choose **Fixed** from the drop-down menu and manually specify the URL in the field provided. The endpoint can be specified as a fixed value, parameterized value, or scripted value. For details about scripting values, see Extensibility and Scripting Basics.

Method

The Method setting is available for REST Clients. It specifies which method is used to processes the request. This field is disabled if the **Constrain to WSDL** check box is selected. The method to invoke can be specified as a fixed value, parameterized value, or scripted value. For details about scripting values, see Extensibility and Scripting Basics.

SOAPAction

The SOAPAction setting is available for SOAP Clients and specifies how the server processes the request. This field is disabled if the **Constrain to WSDL** option is enabled.

Message Exchange Pattern

Enable the **Expect synchronous response** option if a response body is expected. This option is enabled by default because an HTTP response header is always expected. If this option is not enabled, then a one-way message is sent. The service may send a notification header (typically "HTTP/1.0 202 Accepted") in response.

Connection Settings

Specifies Keep-Alive or Close connections.

- **Keep-Alive connection:** Adds a "Connection: Keep-Alive" header to request a keep-alive connection if the server supports it. This is required for NTLM and Digest HTTP authentication.
- **Close connection (default):** Outputs no additional HTTP headers and performs a regular HTTP 1.0 exchange. This is the default behavior for HTTP 1.0.

The connection setting enabled will also be reused for a single invocation of a test suite from the GUI or the command line.

Redirect Settings

Enable the **Follow HTTP redirects** option for messaging clients to automatically follow HTTP redirects. Disable this option if you want to perform an action or validation on the original request/response traffic (instead of working only with the final request/response pair).

Compression Settings

Compression settings are available for messaging clients. They specify whether to compress requests and decompress responses. You can enable the following compression setting options:

- **Gzip request payload:** Gzips the request payloads being sent over the network. Data sent to attached tools will not be compressed. Note that compression does not apply to SOAP Clients configured to send attachments or in MTOM mode.
- **Decompress gzip-encoded response payload:** Decompresses response payloads that have "Content-Encoding: gzip" as a header field. Attached tools will receive the uncompressed data.

URL Parameters

The URL Parameters settings are available for Messaging Client tools. The interface enables you to add parameters to the URL for a GET request. After clicking the **Add** button, you can specify **Parameters/Value** pairs in the dialog that opens. If a data source is available, you can parameterize the values as well.

> **Message Client URL Parameter Format**
>
> URL query parameters are formatted according to the "application/x-www-form-urlencoded" content type. Space characters are replaced with '+'. Non alpha numeric characters are replaced with a percent sign followed by two hexadecimal digits representing the character code. Names and values are separated by '=' and name-value pairs are separated by '&'.
>
> If you want to use a different format, query parameters can also be specified directly at the end of the tool's endpoint URL (instead of in the URL Parameters section). For example, you could use http://host:8080/path?a=1&b=2&c=3

Security

Security settings for the transport are spread across the following tabs.

Client side SSL

Enable the **Use client key store** option to specify the key store used to complete the handshake with the server.

HTTP Authentication

Enable the **Perform authentication** option to set up basic, NTLM, Digest, or Kerberos authentication. You can enable the **Use Global Preferences** optio n to use the authentication settings configured in the Security Preferences (see Security Settings) or choose an authentication type from the Type drop-down menu to configure authentication settings that apply to the client. You can specify the following types:

- Basic
- NTLM
- Kerberos
- Digest

For **Basic, NTLM,** or **Digest**, enter the **Username** and **Password** to authenticate the request.

For **Kerberos**, enter the **Service Principal** to authenticate the request. If the correct username and password, or the correct service principal, are not used, the request will not be authenticated.

- **Use Global Preferences:** Alternatively, you can select **Use Global Preferences** if you have set Global HTTP Authentication Properties within

OAuth Authentication

Configure the OAuth Authentication settings for clients that connect to services that perform authentication under OAuth 1.0a. For OAuth 2.0, authentication is configured in the REST Client's Resource and Payload tabs. Refer to OAuth Authentication for additional details. You can configure the following settings:

- **Perform Authentication:** Enabling this option indicates that OAuth Authentication should be performed. An Authentication field containing OAuth specific information will be added to the HTTP Header.
- **Consumer Key and Secret Configuration:** The Consumer Key and Consumer Secret are the credentials that the client uses to validate itself with the server. The Consumer Key is unique to each client using it. Both of these are required at all steps.
- **OAuth Authentication Mode:** Specifies what step of the OAuth Scenario you'd like to perform.
    - **Obtain Request Token:** Requests the Request Token from the server using the Consumer Key and Secret.
    - **Scope:** Restricts what information may be accessed. This information in embedded into the Consumer Key.
    - **Exchange Request Token for Access Token:** Exchanges the Request Token plus the verification code for the Access Token.
- **Request Token:** Specifies Temporary Request Token credentials obtained from the server (used to exchange for the Access Token).
- **Request Token Secret:** Specifies Temporary Request Token credentials obtained from the server (used to exchange for the Access Token).
- **Verification Code:** Specifies the verification code provided by the server; this confirms that the resource owner will grant permission.
    - **Sign Request for OAuth Authentication:** Uses the specified Access Token and Access Token Secret to give the client access to the user's private resources.
- **OAuth Parameters:** Allows you to specify additional parameters on the OAuth Token— for example, the timestamp and nonce.

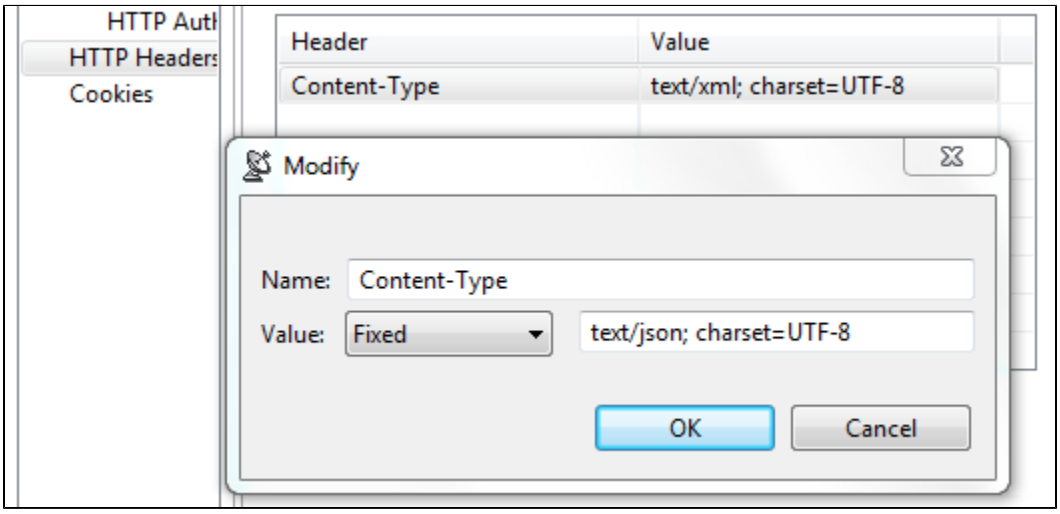
For details on using OAuth authorization, see Using OAuth Authentication.

HTTP Headers

You can specify HTTP Headers to include with your request. Use the following controls to add header names and values:

- **Add:** Click to add a custom HTTP header. The header name is case insensitive.
- **Modify:** Click to modify the selected HTTP header. A dialog box will display, allowing you to modify the Name and Value of the header. If the tool is using a data source, values for the header can be accessed from the data source.
- **Remove:** Click to delete the selected HTTP header.

These controls are used to override header fields. For example, you can override the Content-Type header field by specifying the desired name and value via these controls.



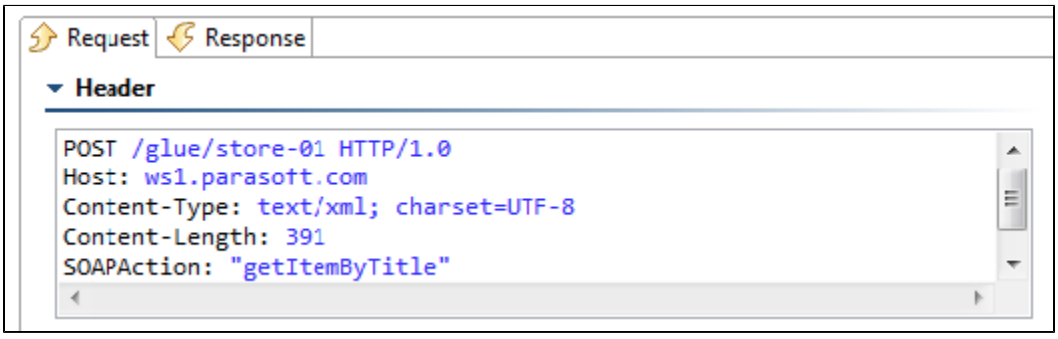
The following headers are added by default and can be overridden using these UI controls.

Host

This header value contains the host name and port number from the HTTP endpoint or resource URL.

Content-Type

This header specifies the media type of the outgoing message. This header is only sent if the outgoing message contains a body which is controlled by the HTTP method. A body is sent for POST, PUT, and DELETE methods and not for GET, OPTIONS, HEAD, or TRACE. The default value is determined based on the type of message being sent. The content-type of a SOAP message will vary depending on the SOAP version, "text/xml" for SOAP 1.1 or "application/soap+xml" for SOAP 1.2. Other XML messages will use "text/xml" by default. JSON messages will use "application/json". A message configured using the Table view will use "application/x-www-form-urlencoded". A message sent with MIME attachments will contain a "multipart" content-type with "start" and "boundary" parameters. Messages belonging to EDI, Fixed Length, CSV, or Custom message formats will have the media type for the message format.



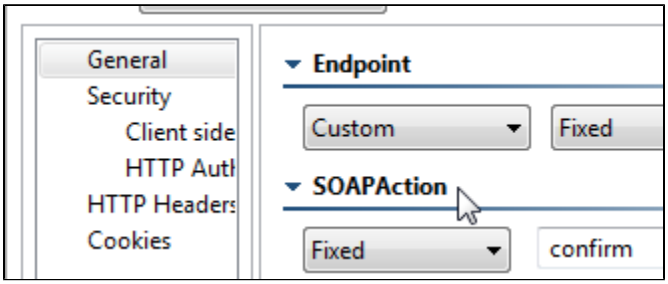
Content-Length

The size of the outgoing message in bytes.

The following HTTP headers are configured conditionally. They are configured outside of this table or have values that must be dynamically generated.

SOAPAction

This HTTP header is sent for SOAP 1.1 only. It is set in the SOAPAction field of the General settings



Authorization

This header is constructed automatically based on the HTTP Authentication and OAuth settings specified in your preferences (under Security> HTTP Authentication and OAuth). The value for NTLM, Digest, and Kerberos authentication will vary depending on various factors, including dynamically-generated challenge responses and security tokens.

Connection

This header is added to the message with value of `Keep-Alive` if **Keep-Alive connection** is enabled. This header is not sent if **Close connection** is enabled (this is the default). Keep-Alive must be enabled for NTLM and Digest HTTP authentication.

Proxy-Authorization

This header is constructed based on the Proxy Authentication settings in the preferences and whether the server indicated that proxy authentication is required.

Cookies

Choose **Custom** from the Cookies menu and enable the **Reset existing cookies before sending request** option to clear the current cookies so that next HTTP invocations start a new session.