



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Administrator Extension Developer's Guide

Web Services Elements

12/14/2025

Web Services Elements

Note: The following pages include more information on how to use the `<script>` tag in the Execution Block.

- [JavaScript Usage](#)
- [JavaScript Representation of Configuration Objects](#)

`<http:request>`

Child Elements

- `<http:header>`
- `<body>`

Attributes

Name	Type	Description	Default	Values
url	URL address	The URL of the web service in GAX (for example: "\${BaseURL} + '/api/opm/parameters/'").	—	—
method	string (required)	The action invoked by the web service.	—	GET, POST, PUT, DELETE
result	variable (required)	A JavaScript variable that stores the result of the web service request, including the full HTTP response.	—	• code—The response status code, such as 200 "OK" or 304 "Not Modified" • content—The response content. • location—The redirect URL. This is only required when status code is 301 "Moved Permanently" or 307

Name	Type	Description	Default	Values
				"Temporary Redirect". Note: A status code value from 100 to 399 is defined as a successful operation. A status code value of 400 or higher is defined as a failed operation.

<http:header>

This element defines one or more headers for the HTTP request. It is optional.

Attributes

Name	Type	Description	Default	Values
name	string (required)	The name of the header.	Content-type	This can be any standard HTTP header name. Click here for a complete list. For GAX, typical values might be: • Accept • Content-type
value	string (required)	The value of the header.	application/json	This can be any standard HTTP header value. Click here for examples.

<body>

This element defines the body to send in the HTTP request. It is mandatory if the <http:request> value is POST or PUT.

The content between <body> and </body> is treated as type message-body and sent with the HTTP request. The content is in string format. For example, to send a variable var in JSON format, you might use the following: <body>\${JSON.stringify(var)}</body>.