

HTTPClient_Services

Application service module that provides OpenInsight HTTP communication features.

Syntax

```
Response = HTTPClient_Services(@Service, @Params)
```

Returns

The meaning of the response value depends on the service.

Parameters

Parameter	Description
@Service	The name of the service being requested. Required.
@Params	Generic parameters. Refer to a specific service to determine the actual parameters used.

Remarks

[HTTPClient_Services](#) provides very useful application services so OpenInsight applications can make HTTP requests to other web servers. The response to each request is kept in memory until the next request is made, thus allowing the developer to get various HTTP response results when it is convenient (such as the body, status code, or response header values).

Services

Service	Description
---------	-------------

SendHTTPRequest	Usage: HTTPClient_Services('SendHTTPRequest', Method, URL, HeaderList, Body, ProxyUser, ProxyPassword, UseAsynchronous, ClientToUse, ClientCertPath, ResponseIsBinary)																																	
	Comments: Calls the indicated HTTP web service routine. This is the primary service of this module. All other services either support this SendHTTPRequest or are dependent upon SendHTTPRequest completing a request. The arguments are defined as follows:																																	
	<table><tr><th>Argument</th><th>Description</th><th>Importance</th></tr><tr><td>Method</td><td>The HTTP method to submit to the server.</td><td>Required</td></tr><tr><td>URL</td><td>The URL receiving the HTTP request.</td><td>Required</td></tr><tr><td>HeaderList</td><td>An @FM/@VM list of request header names and values.</td><td>Optional</td></tr><tr><td>Body</td><td>The request body to be sent to the server.</td><td>Optional</td></tr><tr><td>ProxyUser</td><td>Username needed to authenticate against a proxy server</td><td>Optional</td></tr><tr><td>ProxyPassword</td><td>Password needed to authenticate against a proxy server.</td><td>Optional</td></tr><tr><td>UseAsynchronous</td><td>Flag to determine if the HTTP request should be processed asynchronously. Default is True.</td><td>Optional</td></tr><tr><td>ClientToUse</td><td>HTTP client to use for the request. Options are <i>ClientXMLHTTP</i>, <i>ServerXMLHTTP</i>, or <i>PowerShell</i>. If True\$ is passed in then <i>ClientXMLHTTP</i> will be used. If False\$ is passed in then <i>ServerXMLHTTP</i> will be used. If any other value is passed in (including an empty string) then <i>ServerXMLHTTP</i> will be used. This will allow for backwards compatibility.</td><td>Optional</td></tr><tr><td>ClientCertPath</td><td>Path to a client-side certificate. This is usually in Location\Certificate Store\Subject format.</td><td>Optional</td></tr><tr><td>ResponseIsBinary</td><td>Boolean flag to indicate if the response is expected to be binary. Default is False\$.</td><td>Optional</td></tr></table>	Argument	Description	Importance	Method	The HTTP method to submit to the server.	Required	URL	The URL receiving the HTTP request.	Required	HeaderList	An @FM/@VM list of request header names and values.	Optional	Body	The request body to be sent to the server.	Optional	ProxyUser	Username needed to authenticate against a proxy server	Optional	ProxyPassword	Password needed to authenticate against a proxy server.	Optional	UseAsynchronous	Flag to determine if the HTTP request should be processed asynchronously. Default is True.	Optional	ClientToUse	HTTP client to use for the request. Options are <i>ClientXMLHTTP</i> , <i>ServerXMLHTTP</i> , or <i>PowerShell</i> . If True\$ is passed in then <i>ClientXMLHTTP</i> will be used. If False\$ is passed in then <i>ServerXMLHTTP</i> will be used. If any other value is passed in (including an empty string) then <i>ServerXMLHTTP</i> will be used. This will allow for backwards compatibility.	Optional	ClientCertPath	Path to a client-side certificate. This is usually in Location\Certificate Store\Subject format.	Optional	ResponseIsBinary	Boolean flag to indicate if the response is expected to be binary. Default is False\$.	Optional
	Argument	Description	Importance																															
	Method	The HTTP method to submit to the server.	Required																															
	URL	The URL receiving the HTTP request.	Required																															
	HeaderList	An @FM/@VM list of request header names and values.	Optional																															
	Body	The request body to be sent to the server.	Optional																															
	ProxyUser	Username needed to authenticate against a proxy server	Optional																															
	ProxyPassword	Password needed to authenticate against a proxy server.	Optional																															
UseAsynchronous	Flag to determine if the HTTP request should be processed asynchronously. Default is True.	Optional																																
ClientToUse	HTTP client to use for the request. Options are <i>ClientXMLHTTP</i> , <i>ServerXMLHTTP</i> , or <i>PowerShell</i> . If True\$ is passed in then <i>ClientXMLHTTP</i> will be used. If False\$ is passed in then <i>ServerXMLHTTP</i> will be used. If any other value is passed in (including an empty string) then <i>ServerXMLHTTP</i> will be used. This will allow for backwards compatibility.	Optional																																
ClientCertPath	Path to a client-side certificate. This is usually in Location\Certificate Store\Subject format.	Optional																																
ResponseIsBinary	Boolean flag to indicate if the response is expected to be binary. Default is False\$.	Optional																																
Returns: The HTTP response body from the server. The HTTP response body can also be retrieved via the GetResponseBody service.																																		
SetRequestHeaderFields	Usage: HTTPClient_Services('SetRequestHeaderFields', HeaderList) Comments: Sets the Request Header Fields as indicated by the HeaderList argument. The HeaderList should not be formatted as a list rather than an associated multivalue array. This is easier for the developer to manage in the code that calls this service. Returns: N/A																																	
SetRequestHeaderField	Usage: HTTPClient_Services('SetRequestHeaderField', Name, Value) Comments: Sets the indicated Request Header Field with the indicated value. Returns: N/A																																	
GetRequestHeaderFields	Usage: HTTPClient_Services('GetRequestHeaderFields') Comments: Returns all of the Request Header Field names and values. This returns an @FM/@VM list of names and values. This is called from the SendHTTPRequest service automatically. The developer does not need to use this service. Returns: An @FM/@VM list of request header names and values.																																	
GetRequestHeaderField	Usage: HTTPClient_Services('GetRequestHeaderField', Name) Comments: Returns the value previously set for the indicated Request Header Field. Returns: The value previously set for the indicated Request Header Field.																																	

SetResponseHeaderField	Usage: HTTPClient_Services('SetResponseHeaderField', Name, Value) Comments: Sets the indicated Response Header Field with the indicated value. This is called from the SendHttpRequest service automatically after retrieving the response headers fields from the server. Returns: N/A
GetResponseHeaderFields	Usage: HTTPClient_Services('GetResponseHeaderFields') Comments: Returns all of the Response Header Field names and values. This returns an @FM/@VM list of names and values. Returns: An @FM/@VM list of response header names and values.
GetResponseHeaderField	Usage: HTTPClient_Services('GetResponseHeaderField', Name, Delimiter) Comments: Returns the value previously set for the indicated Response Header Field. The Name argument is case-insensitive but if the indicated Response Header Field has not been set then an error condition will be set. Returns: The value previously set for the indicated Response Header Field.
GetCookie	Usage: HTTPClient_Services('GetCookie', Name, IgnoreAttributes) Comments: Returns the value for the indicated cookie name. If the IgnoreAttributes argument is True the cookies attributes will be removed. See this section of RFC6265 and following for examples of cookie attributes. The default value for IgnoreAttributes is False. Returns: See comments.
GetCookies	Usage: HTTPClient_Services('GetCookies', Delimiter) Comments: Returns all cookie strings from the response headers using the indicated delimiter. If the Delimiter argument is empty, all cookie strings will be delimited with @FM. Returns: See comments.
SetResponseStatus	Usage: HTTPClient_Services('SetResponseStatusCode', Code) Comments: Sets the response status code generated by the most recent HTTP request. This is called from the SendHttpRequest service automatically after retrieving the response status from the server. Returns: N/A
GetResponseStatus	Usage: HTTPClient_Services('GetResponseStatusCode') Comments: Gets the response status code generated by the most recent HTTP request. Returns: The response status code generated by the most recent HTTP request.
SetResponseStatusPhrase	Usage: HTTPClient_Services('SetResponseStatusPhrase', Phrase) Comments: Sets the response status phrase generated by the most recent HTTP request. This is called from the SendHttpRequest service automatically after retrieving the response status phrase from the server. Returns: N/A

GetResponseStatusPhrase	Usage: HTTPClient_Services('GetResponseStatusPhrase') Comments: Gets the response status phrase generated by the most recent HTTP request. Returns: The response status phrase generated by the most recent HTTP request.
SetResponseBody	Usage: HTTPClient_Services('SetResponseBody', Body) Comments: Sets the response body generated by the most recent HTTP request. This is called from the SendHttpRequest service automatically after retrieving the response body from the server. Returns: N/A
GetResponseBody	Usage: HTTPClient_Services('GetResponseBody') Comments: Gets the response body generated by the most recent HTTP request. This is the same response that the SendHttpRequest service returns. Returns: The response body generated by the most recent HTTP request.
SetTimeoutDuration	Usage: HTTPClient_Services('SetTimeoutDuration', TimeoutDuration) Comments: Sets the timeout duration that will be used before an HTTP request will abort. This is only applicable if the request is asynchronous. If the timeout duration is empty, a default setting of 30 minutes (1800 seconds) will be set. Returns: N/A
GetTimeoutDuration	Usage: HTTPClient_Services('GetTimeoutDuration') Comments: Returns the timeout duration that will be used before an HTTP request will abort. This is only applicable if the request is asynchronous. If the timeout duration is empty, a default setting of 30 minutes (1800 seconds) will be returned. Returns: The timeout duration that will be used before an HTTP request will abort.
ClearResponseSettings	Usage: HTTPClient_Services('ClearResponseSettings') Comments: Clears all of the global common variables used for responses. This is called from the SendHttpRequest service automatically. The developer does not need to use this service. Returns: N/A
ClearRequestSettings	Usage: HTTPClient_Services('ClearRequestSettings') Comments: Clears all of the global common variables used for requests. This is called from the SendHttpRequest service automatically. The developer does not need to use this service. Returns: N/A

Params

The proper use of the generic arguments are defined in the definition of each service above.