

HTTP Module

The module provides the ability to make HTTP requests.

```
require 'http'
```

API Requests:

http.request()

Configures and starts a new asynchronous HTTP request. Returns connection identifier.

Name	Type	Required	Description
url	string	+	Full url for making connection (can contain custom port)
skip_security	bool	-	Skip any security checks: check date, domain, and self-signed certificates criteria. Default value: false .
type	string	-	Type of request. Possible value: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE. Default value: GET .
content_type	string	-	Type of sending content
user	string	-	User name
password	string	-	User password
headers	array of objects	-	Custom request headers
headers.key	string	+	Name of header record
headers.value	string	+	Value of header record

keep_alive	bool	-	Keep HTTP requests alive after establishing connection. Each plugin can have not more than 50 "alive" connections. Default value: false .
repeat_allowed	bool	-	Allows sequential repeat_request() call after the request is completed. Otherwise, it allows you to get multiple responses on a single request. Default value: true .
content_length	int	-	Length of content data in bytes
data	string	-	Data for sending
last	bool	-	Was all data sent or not ? Default value: true. If this field is false then an http_data_send event would be generated for sending the next part of data for this request.
redirection	int	-	Max number of allowed redirection. Default value: 3 . Max value can be: 15 .
handler	string	+	Path to script file of plugin which should be executed when any event of connection would be generated(in HUB:my_plugin/http_receive.lua format)
user_data	any	-	Any data which will be passed to receive_script and send_script. Can be of any valid type: string, number, boolean, array, object. If user_data is null, the value will not be set.
new_data_buffer	int	-	Preferred size in bytes of buffer which should be used for receiving new data via http_data_received event. Minimal possible size: 32, max possible size: 500000 (Linux) or 1024 (RTOS).
new_data_cache	bool	-	Do not send an http_data_received event unless the buffer is full or the end of response (or it's significant part) was detected. Default value: true. When disabled, events will be triggered immediately after new portions of data received (useful for custom format response streams) thus increasing the amount of events with smaller portions of data.
timeout	int	-	Timeout in milliseconds for waiting response. Default value: 30000 milliseconds. Max possible value: 360000 milliseconds.
connect_timeout	int	-	Timeout in milliseconds for establishing connection. Default value: 10000 milliseconds. Max possible value: 600000 milliseconds.

Return value: id of the request

Lua errors:

- Required parameter didn't define: `parameter_name`
- The file doesn't exist: `file_path`
- Plugin doesn't have permissions to the file: `file_path`
- Wrong format of parameter: `parameter_name`
- Limit of parameter is reached: `parameter_name`
- Limit of connections is reached

http.repeat_request()

Make one more request with an already opened connection.

Name	Type	Required	Description
id	string	+	Unique id of existing request
type	string	-	Type of request. Possible value: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE. Default value: GET .
content_type	string	-	Type of sending content
headers	array of objects	-	Custom request headers
headers.key	string	+	Name of header record
headers.value	string	+	Value of header record
keep_alive	bool	-	Keep HTTP requests alive after establishing connection. Each plugin can have not more than 50 "alive" connections. Default value: false .
repeat_allowed	bool	-	Allows sequential repeat_request() call after the request is completed. Otherwise, it allows you to get multiple responses on a single request. Default value: true .
content_length	int	-	Length of content data in bytes
data	string	-	Data for sending

last	bool	-	Does all the data was sent or not ? Default value: true. If this field is false then an http_data_send event would be generated for sending the next part of data for this request.
handler	string	+	Path to script file of plugin which should be executed when any event of connection would be generated(in HUB:my_plugin/http_receive.lua format)
user_data	any	-	Any data which will be passed to receive_script and send_script. Can be of any valid type: string, number, boolean, array, object. If user_data is null, the value will not be set.
timeout	int	-	Timeout in milliseconds for waiting response. Default value: 30000 milliseconds. Max possible value: 3600000 milliseconds.
connect_timeout	int	-	Timeout in milliseconds for establishing connection. Default value: 10000 milliseconds. Max possible value: 600000 milliseconds.

Lua errors:

- Required parameter didn't define: `parameter_name`
- The file doesn't exist: `file_path`
- The plugin doesn't have permissions to the file: `file_path`
- Wrong format of the parameter: `parameter_name`
- Limit of the parameter is reached: `parameter_name`
- Connection doesn't exist
- Sending of previous request data didn't finish

http.close_connection()

Close existing connection. A plugin can use only its own connections.

Name	Type	Required	Description
id	string	+	Unique ID of request

Return value: bool

Lua errors:

- Required parameter didn't define: `parameter_name`
- Wrong format of the parameter: `parameter_name`
- Limit of the parameter is reached: `parameter_name`
- Connection doesn't exist

http.send_data()

Method for sending the next portion of data for the existing request. A plugin can use only its own connections.

Name	Type	Required	Description
id	string	+	Unique ID of request
data	string	+	Data for sending
last	bool	-	It's the last part of data or not. Default value: true . If it is not the last part of data then the http_data_send event will be generated when existing data is sent.

Return value: Number of sent/buffered bytes

Lua errors:

- Required parameter didn't define: `parameter_name`
- Wrong format of the parameter: `parameter_name`
- Limit of the parameter is reached: `parameter_name`
- Connection doesn't exist
- Request for sending data wasn't started

http.get_connection_info()

Get the main information about a connection. A plugin can ask only about its own connections.

Name	Type	Required	Description
id	string	+	Unique ID of request

Return value:

Name	Type	Required	Description
id	string	+	Unique ID of request
url	string	+	Full url for making connection (can contain custom port)

user_data	any	-	User data which was passed to http.request() call. Can be of any valid type: string, number, boolean, array, object. If user_data is null, the value will not be set.
max_data_length	int	+	Size of data which can be send peer one script execution via http.send_data() method

Lua errors:

- Required parameter didn't define: parameter_name
- Wrong format of the parameter: parameter_name
- Limit of the parameter is reached: parameter_name
- Connection doesn't exist