

HTTP Callback Server Script

The HTTP Callback Server Script is a script for the Scripting Connector in Diafaan SMS Server. The script starts a basic HTTP server on port 8080 and processes HTTP callback messages from Diafaan SMS Server (basic and full editions only). This script can be used to change or block all incoming and outgoing SMS messages in Diafaan SMS Server and can also be used for message routing, user/password validation and user accounting.

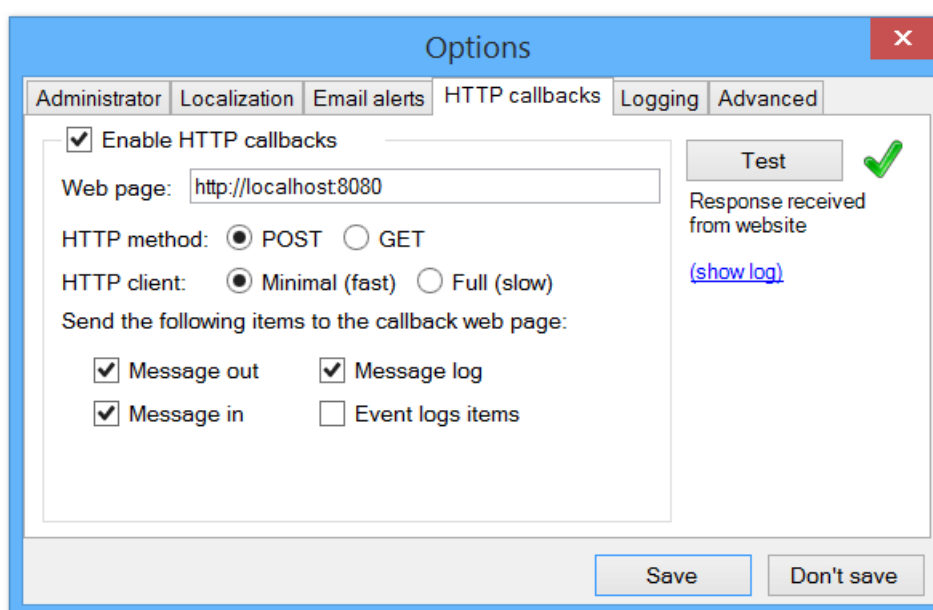
Add the script to the Scripting Connector

The HTTP Callback Server Script replaces the normal connector script. After replacing the script, the Compile button can be used to check if the script works correctly.

The 'GetHttpCallbackResponse' method in the script is called when Diafaan SMS Server sends an HTTP callback. The default script returns the standard 'result=1' positive response. This response means that the message or user is accepted without changing one of the message or user parameters.

Enable the HTTP Callback feature

The HTTP Callback feature can be enabled in the main options of Diafaan SMS Server (select Actions-Options-HTTP callbacks). The HTTP Callback Server Script listens to port 8080 on the local host on web page '<http://localhost:8080>'.



Changing the script

The script must be changed to to anything useful, in a simple script only the 'GetHttpCallbackResponse' has to be changed but in a more complicated script other parts of the script might have to be changed too.

It is very important to keep the processing time for each call as short as possible, preferable within 50ms. This is especially important when the script makes calls to external resources like a database or a web server. When the HTTP callback takes more than 5 seconds the message is rejected completely.

Example scripts

Example 1, allow only messages to British phone numbers

```
private string GetHttpCallbackResponse(NameValueCollection queryList)
{
    string response = "result=1";
    string action = GetQueryParameter(queryList, "action");

    if (action == "message_out") {
        string to = GetQueryParameter(queryList, "to");
        if (!to.StartsWith("+44")) response = "result=0";
    }
    return response;
}
```

Example 2, send messages with number 'group' to multiple numbers

```
private string GetHttpCallbackResponse(NameValueCollection queryList)
{
    string response = "result=1";
    string action = GetQueryParameter(queryList, "action");

    if (action == "message_out") {
        string to = GetQueryParameter(queryList, "to");
        if (to == "group") {
            response = "result=1&to=" + HttpUtility.UrlEncode("+35xxxxxxxx1");
            string from = GetQueryParameter(queryList, "message");
            string message = GetQueryParameter(queryList, "message");
            string messageType = GetQueryParameter(queryList, "message_type");
            string userId = GetQueryParameter(queryList, "user_id");
            string userInfo = GetQueryParameter(queryList, "user_info");
            PostSendMessage("+35xxxxxxxx2", from, message, messageType,
                "", userId, userInfo, 0);
            PostSendMessage("+35xxxxxxxx3", from, message, messageType,
                "", userId, userInfo, 0);
            PostSendMessage("+35xxxxxxxx4", from, message, messageType,
                "", userId, userInfo, 0);
        }
    }
    return response;
}
```

Example 3, remove first word from the message text

```
private string GetHttpCallbackResponse(NameValueCollection queryList)
{
    string response = "result=1";
    string action = GetQueryParameter(queryList, "action");

    if (action == "message_out") {
        string message = GetQueryParameter(queryList, "message");
        int firstSpace = message.IndexOf(' ');
        if (firstSpace >= 0) {
            message = message.Substring(firstSpace).Trim();
            response = "result=1&message=" + HttpUtility.UrlEncode(message);
        }
    }
    return response;
}
```

Example 4, route messages through different gateways

```
private string GetHttpCallbackResponse(NameValueCollection queryList)
{
    string response = "result=1";
    string gateway1 = HttpUtility.UrlEncode("GSM Modem Gateway 1");
    string gateway2 = HttpUtility.UrlEncode("GSM Modem Gateway 2");
    string gateway3 = HttpUtility.UrlEncode("SMPP Gateway");
    string modemGateways = HttpUtility.UrlEncode("GSM Modem Gateway 1;GSM Modem
Gateway 2");
    string allGateways = "";
    string action = GetQueryParameter(queryList, "action");

    if (action == "message_out") {
        string to = GetQueryParameter(queryList, "to");
        if (to.StartsWith("+44")) {
            response += "&gateway=" + gateway1;
        }
        else if (to.StartsWith("+32")) {
            response += "&gateway=" + gateway2;
        }
        else if (to.StartsWith("0")) {
            response += "&gateway=" + modemGateways;
        }
        else {
            response += "&gateway=" + allGateways;
        }
    }
    return response;
}
```