



Logbook API Development Guide

Ver. 2.0

November, 10, 2014 by Dani Manchado

1) Why logbook API?

Since the beginning of MyeQSL.net as QSL exchange portal and electronic logbook, I already had in mind the development of a remote application to control the logbook, and thus able to integrate different applications, whether desktop, mobile or web. This is the first result.

2) How it works?

Before using the API, you need to be logged in on the remote system, you communicate a unique session key, and with her and your callsign, you can access the various control options.

3) Login

First is to make a call to **<http://www.myeqsl.net/remote/login.php>** to log data with user registration MyeQSL.net

Two "POST" fields (Or "GET") are needed to log in MyeQSL and able to use the remote system. One has to be called "c", which is the callsign of the user, and the other is to be called "p", which is the md5 of the password.

If successful, the system returns a 32-byte hexadecimal key, is the session key. It is necessary for any management system.

An example of the Post query:

```
"c=33KK999&p= ff4c48e9995f076b2uub2a90ae28d5aa"
```

An example of the Get query:

```
"http://www.myeqsl.net/remote/login.php ?c=33KK999&p= ff4c48e9995f076b2uub2a90ae28d5aa"
```

4) Read the logbook

Post method

To download your logbook, you need access to the following URL and pass some parameters in the POST query: **<http://www.myeqsl.net/remote/getlogbook.php>**

POST query parameters:

"c" => Is your callsign.

"s" => Is your unique session of 32 bytes key.

"fc" => Is a "Follow code". If in your application need follow the calls to the remote system, you can add a string in this parameter to follow the order.

If successful, the system returns a string like the following for each of the entries that are in the logbook:

```
** IN: 1/2 <br>           // Start, with the entry number of total.  
30SAT003<br>           //FROM  
30LOM276<br>           //TO  
2233<br>           //UTC Time
```

```

20130223<br>           //Date
59<br>                 //RST received if QSO is confirmed
47<br>                 //RST sent
Bla bla<br>           //Comment sent
More blab la<br>       //Comment received if QSO is confirmed
27565<br>             //Frequency
1<br>                 //Confirmed: 0=Not confirmed, 1=Confirmed, 2=Deny
USB<br>               //Mode
Getlog<br>           //Return the Follow code
** END OF ENTRY<br>   //End of entry

```

Get method

To download your logbook, you need access to the following URL and pass some parameters in the GET query: <http://www.myeqsl.net/remote/getlogbook2.php>

POST query parameters:

“c” => Is your callsign.

“s” => Is your unique session of 32 bytes key.

“fc” => Is a “Follow code”. If in your application need follow the calls to the remote system, you can add a string in this parameter to follow the order.

If successful, the system returns a string like the following for each of the entries separates with LN char (13) that are in the logbook:

```

30SAT003           //FROM
30LOM276           //TO
2233              //UTC Time
20130223          //Date
59                //RST received if QSO is confirmed
47                //RST sent
Bla bla           //Comment sent
More blab la      //Comment received if QSO is confirmed
27565             //Frequency
1                 //Confirmed: 0=Not confirmed, 1=Confirmed, 2=Deny
USB               //Mode
Getlog            //Return the Follow code
23                //Entry number

```

5) Read the Inbox

Post method

To download your inbox, you need access to the following URL and pass some parameters in the POST query: <http://www.myeqsl.net/remote/getinbox.php>

POST query parameters:

“c” => Is your callsign.

“s” => Is your unique session of 32 bytes key.

“fc” => Is a “Follow code”. If in your application need follow the calls to the remote system, you can add a string in this parameter to follow the order.

If successful, the system returns a string like the following for each of the entries that are in the logbook:

```
** IN: 1/2 <br>           // Start, with the entry number of total.  
30SAT003<br>           //FROM  
30LOM276<br>           //TO  
2233<br>           //UTC Time  
20130223<br>           //Date  
59<br>           //RST received if QSO is confirmed  
47<br>           //RST sent  
Bla bla<br>           //Comment sent  
More blab la<br>           //Comment received if QSO is confirmed  
27565<br>           //Frequency  
1<br>           //Confirmed: 0=Not confirmed, 1=Confirmed, 2=Denny  
USB<br>           //Mode  
Getlog<br>           //Return the Follow code  
765<br>           //QSO index. This is needed to confirm or deny a QSO  
** END OF ENTRY<br>           //End of entry
```

Get method

To download your inbox, you need access to the following URL and pass some parameters in the GET query: <http://www.myeqsl.net/remote/getinbox2.php>

GET query parameters:

“c” => Is your callsign.

“s” => Is your unique session of 32 bytes key.

“fc” => Is a “Follow code”. If in your application need follow the calls to the remote system, you can add a string in this parameter to follow the order.

If successful, the system returns a string like the following for each of the entries separates with the LN char (13) that are in the logbook:

```
30SAT003           //FROM  
30LOM276           //TO  
2233               //UTC Time  
20130223           //Date  
59                 //RST received if QSO is confirmed  
47>               //RST sent  
Bla bla           //Comment sent  
More blab la       //Comment received if QSO is confirmed  
27565             //Frequency  
1                 //Confirmed: 0=Not confirmed, 1=Confirmed, 2=Denny  
USB               //Mode  
Getlog            //Return the Follow code  
765               //QSO index. This is needed to confirm or deny a QSO
```

6) Add a QSO in the logbook

To add a QSO in the logbook, you need access to the following URL and pass some parameters in the

- POST query: <http://www.myeqsl.net/remote/addqso.php>
- GET query: <http://www.myeqsl.net/remote/addqso2.php>

POST/GET query parameters:

- "c" => Is your callsign.
- "s" => Is your unique session of 32 bytes key.
- "fc" => Is a "Follow code". If in your application need follow the calls to the remote system, you can add a string in this parameter to follow the order.
- "in " =>Is the QSO callsign Ex: 30TTT544
- "da" => Is the UTC Date(YYYYMMDD format) Ex: 20130312
- "ut" => Is the UTC time (HHMM format), Ex: 2235
- "fr" => Is the frequency, Ex: 11M or 27654
- "mo" => Is the mode, Ex: RTTY
- "rs" => Is the RST Sent, Ex: 599 or -15
- "co" => Is the comments sent, Ex: "Thank you Peter, 73s"
- "fc" => Is the follow code

If successful, the system returns a string with the follow code.

7) Confirming or denying a contact

With MyeQSL.net in remote system, you can accept or deny a contact that has reached the inbox, to do so, access the following URL with POST parameters indicated:

<http://www.myeqsl.net/remote/confirmqso.php>

POST query:

- "c" => Is your callsign.
- "s" => Is your unique session of 32 bytes key.
- "fc" => Is a "Follow code". If in your application need follow the calls to the remote system, you can add a string in this parameter to follow the order.
- "ix" => Is the QSO index indicated in the inbox download.
- "cf" =>The confirmation: "1" is confirmed, "2" is deny

If successful, the system returns a string with the follow code.

8) Get a QSL card from the Inbox or Confirmed in the Logbook

With MyeQSL.net in remote system, you download a QSL image from the inbox, to do so, access the following URL with:

- POST parameters indicated:
<http://www.myeqsl.net/remote/getqslfrom.php>
- GET parameters:
<http://www.myeqsl.net/remote/getqslfrom2.php>

POST or GET query:

- "c"** => Is your callsign.
- "s"** => Is your unique session of 32 bytes key.
- "fr"** =>The QSO callsign
- "ut"** =>The UTC time when the contact has made.

If successful, the system returns a string with image URL. Now you can download with any method.

9) Get a QSL card sent

With MyeQSL.net in remote system, you download a QSL image from your logbook, to do so, access the following URL with:

- POST parameters indicated:
<http://www.myeqsl.net/remote/getqslto.php>
- GET parameters indicated:
<http://www.myeqsl.net/remote/getqslto2.php>

POST / GET query:

- "c"** => Is your callsign.
- "s"** => Is your unique session of 32 bytes key.
- "to"** =>The QSO callsign
- "ut"** =>The UTC time when the contact has made.

If successful, the system returns a string with image URL. Now you can download with any method.

The Tweet Cluster

Development guide

1) Why a cluster integrated with Twitter?

A classic cluster system, a network of servers is required to keep running, but if we use a network that already exists, it is robust and enduring in time, we are thinking of the social network Twitter.

2) Requirements for the content of tweets

A tweet, so that it is integrated into the cluster system, must contain the following: A hash tag "#CBCluster" which will be based on what the search, and that starts with the text "Spot: " ("S", "p", "o", "t", ":", " ", 6 characters) because if omit this, did not understand as a valid cluster entry.

Tweet format

It must contain the following fields:

"FR" indicates the callsign of who send the spot

"TO:" Indicates the contact's callsign in the spot

"UTC:" Indicates hour UTC, format HHMM, 4 characters.

"DATE:" Indicates the UTC date, format YYYYMMDD, 8 characters

"FZ:" This is the frequency, XXXXX, 5 characters

"MODE:" Indicates the mode that you send the spot.

"COM:" Comments of the spot, up to 25 characters

An example seen from Twitter:



3) MyeQSL Integration

You can use the system set up to send and receive MyeQSL spots cluster.

First is to make a call to <http://www.myeqsl.net/remote/login.php> to log data with user registration MyQSL.eu

Two "POST" fields (Or "GET") are needed to log in MyeQSL and able to use the remote system. One has to be called "c", which is the callsign of the user, and the other is to be called "p", which is the md5 of the password.

If successful, the system returns a 32-byte hexadecimal key, is the session key. It is necessary for any management system.

An example of the Post or Get query:

```
"c=33KK999&p= ff4c48e9995f076b2uub2a90ae28d5aa"
```

3.1) Receive the cluster.

To receive the contents of the cluster without fail to enable a proper application on Twitter, you can use a call to <http://www.myeqsl.net/remote/gettwcluster.php> indicating the following fields in the POST QUERY:

"c": Is the user callsign

"s": Is the session key, of 32 bytes hexadecimal characters.

"qu": The number of spots you want to receive. The default is 10, but can add up to 100, the limit imposed by Twitter.

An example of the post query:

```
"c=33KK999&s= ff4c48e9995f076b2uub2a90ae28d5aa&qu=20"
```

Result of the call:

If successful, the system returns a string with the following format, spots repeated as many times as we have indicated that we want to read:

```
Spot #0<br>           //Spot number, stars in 0, first is the newest  
33VV8888<br>          //The FROM callsign (Spotter)  
66BB9999<br>          //The QSO Callsign  
2233<br>           //UTC, in format HHMM  
20131122<br>          //Date, in format YYYYMMDD  
27666<br>          //The frequency  
RTTY<br>          //The mode  
Peter thank you<br> //Comments  
MyQSL<br>          //Twitter user  
SPOTEND<br>          // "End of transmission..."
```

3.2) Send a spot to the Cluster using MyeQSL

To send a spot to the cluster, without fail to enable a proper application on Twitter, you can use a call to

- <http://www.myeqsl.net/remote/posttwcluster.php> for POST query
- <http://www.myeqsl.net/remote/posttwcluster2.php> for GET query

The get or post fields are:

"c": Is the user callsign

"s": Is the session key, of 32 bytes hexadecimal characters.

"fr": Is the spotter callsign

"to": Is the QSO callsign

"ut": Is the UTC, in format HHMM
 "da": Is the date, in format YYYYMMDD
 "fz": Is the frequency
 "mo": Is the mode
 "co": The spot comments

Example in HTML:

```
<HTML>
<HEAD>
<TITLE>Sending a new QSO to the Tweet Cluster</TITLE>
</HEAD>
<BODY>
  <form name="Sendqso" method="post" action="http://www.myeqsl.net/remote/posttwcluster.php"
  enctype="multipart/form-data" id="Form1">
    <input type="hidden" name="c" value="99HH123">
    <input type="hidden" name="s" value="ff4c48e9995f076uuiiii90ae28d5aa">
    <input type="hidden" name="fr" value="86KK777">
    <input type="hidden" name="to" value="12JJ555">
    <input type="hidden" name="ut" value="4455">
    <input type="hidden" name="da" value="20130323">
    <input type="hidden" name="fz" value="27453">
    <input type="hidden" name="mo" value="RTTY">
    <input type="hidden" name="co" value="Gracias amigo">
    <input type="submit" id="Button1" name="" value="Send Tweet Cluster"
    style="position:absolute;left:10px;top:10px;width:127px;height:25px;z-index:10;">
  </form>
</BODY>
</HTML>
```

NOTE: Do not add your session key in a web page, keep it hidden, this is only an example. There could be malicious people who would use it to hurt.

4) Integrating the Tweet Cluster in your web

You can make it simpler to use cluster if it includes a "frame" on their website. To do this, copy the following html code where you want the cluster:

```
<iframe name="InlineFrame1" id="InlineFrame1" style="width:725px;height:440px;"
src="http://www.myeqsl.net/twcluster.php?n=10" frameborder="0">Su navegador no soporta marcos incrustados
o está configurado para no mostrarlos</iframe>
```

Example:

From	QSO To	Date	UTC	Freq.	Mode	Comments	Twitter User
86KK777	12JJ555	20130323	4455	27453	RTTY	Gracias amigo	@MyQSL
30LOM276	30LOM543	20130323	1148	27555	USB	Gracias Pedrito	@MyQSL
30LOM276	55LDD987	20130323	1142	27555	USB	Thanks 73!ss	@MyQSL
30SAT003	66yy876	20130322	2055	27555	USB	Thanks 73s	@MyQSL
30LOM276	85JJ123	20130322	2041	26855	RTTY	Thanks 73s carrosoo	@MyQSL
30LOM276	123UU777	20130322	2039	27555	OLIV	Pepe	@MyQSL
33XX999	33XX999	20130322	2023	27877	USB	1234567890123456789012345	@MyQSL
30LOM276	192KK675	20130322	1859	27555	USB		@MyQSL
30LOM276	21SAT003	20130322	0926	27877	ROS	Un comentario cualquiera	@danimanchado
33XX999	33XX999	20130322	0926	27877	USB	1234567890123456789012345	@MyQSL

5) Additional information about Twitter OAuth

You can view information about deploying twitter your own application for sending and reading tweets on the system explained above.

<https://dev.twitter.com/docs/api/1.1>