

MAIL1CLICK API - rel 1.35

```
hQIMAwhtcmSLuLpnAQ//VKAuukqmommgqFEdthRMVORODqX6oXyVsUMmkfLyNTq/  
2vOreTmg18JsMty6tpoJ5CjkykDGR9mPg79Ggh1BRdSiqSSQR17oudKwilpJbAmk  
MFUkoVTtzGEfEAfOV0Pfilaf+ntJawYxOaxmHZvtyG9iojsQjOrA4S+3i4K4lpj4  
A/tj7nrDfL47r2cQ83JszWsQVe2CqTLLQz8saXfGoGJILREPFoF/uPS0sg5TyKYJ  
j53iCUrOgptAX0UHraW0HZkwHzWkfwI2NGpkR4D2JGjhWH5gNr71paxBY6jb0vVK  
WJGpMFnOITQO2GppUSf2XNe9McDMtJ7+UDwDznpltg8SLDoEly34JA06LuJq7V2  
AHd21zKjVQBkuKW+xsnoCyDBjbZnZXUQk4u/XL3SAp5dRd5qKVIT2DIisRf2GBKk  
aAntsJiHDY1Bs+sixIjvEoAXCWhbKebE1KILXnE7fnMy1hEu+d623+u1LKIMGEgh  
oDDNSLzTq88rj9nyZ4N8D7s20NPOI1n9xmB1VPJmcd4fg22gLB9cjLmFq79tR4b  
ZqfKM/6bOnGNXUJvKv3/RtF5F0iD/ZZWew+HmRB3BM/BwkblRg4HLF8Ze+ZdunOv  
US1dlSgk3fyOSQxGFQ1kv/kFT+ERD5WX/uY5qLj25xzLiuGwoWJEKgI5z4M1fRGF  
AgwD2TXnPQpf82wBD/42HvBq271b6tdHRpLR7qDPaWUT/1b6VxQtZ0jCdC5sC6XS  
qFPHx0UPUoubFB0TNO1KF9dkP4NS0149SprPGxGzOoS+L+Fen2HJSJV1k1kWIARi
```

This document is the application program interface reference to MAIL1CLICK.

This interface has been used to write the client application available on our website.

Preface:

The encryption/decryption core is kept on the server side and the client should post and read information using an https connection and passing the login/password needed to encrypt/decrypt by the RSA keys stored on the servers.

The clients application should communicate with Mail1Click Servers using **https url request** and receiving back answers in a simple xml format. You can test the protocol using a normal browser.

kmdeletemessage.php
kmdeletemessage.php

Protocol Description:

- **Create new Email address:**

To create a new email address the app should call this url:

<https://kmengine.mail1click.com/kmnewaccount.php>

?login=xxxxx@xxxxxxxxxxx

&password=xxxxxxxxxxxx

&language=xx

&timezone=xx

&otheremail=xxxxxxxxxx@xxxxxx.xxx

the “login” should be the email address including the domain (ex. mail1click.com) if not passed the domain it will be added automatically mail1click.com

The login/email must be new and not already present in our files.

The “language” must be in lower case 2 letters as iso reference codes (en for english,etc,etc.)

MAIL1CLICK API - rel 1.35

The timezone could any value from -23 to +23, it reflects the difference of time compared to Greenwich time.

The field "otheremail" should be normal email address where advice the user for new encrypted messages (optional).

The email should contain only numbers, lower case alpha chars and .- _ as possible different chars from alpha.

The engine will answer if ok:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>
```

or if them email is already in use:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ALREADY PRESENT</ERROR>
</KRYPTOMAIL>
```

if there is any other error:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ANY OTHER ERROR MESSAGE</ERROR>
</KRYPTOMAIL>
```

-Login Authentication:

Your app should authenticate itself receiving the correct server url where all others calls has to be done. It should calling the following url:

```
https://kengine.mail1click.com/kauthenticate.php
?login=xxxxxxxx@xxxxxx.xxx
&password=xxxxxxxx
```

the application will return if the login is correct:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<SERVERNAME>xxxxxxxx</SERVERNAME>
<TOKEN>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</TOKEN>
</KRYPTOMAIL>
```

The TOKEN value should be used as password in all following calls to the API.

MAIL1CLICK API - rel 1.35

if the login is wrong, it will return instead:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
</KRYPTOMAIL>
```

This function can be used to check the validity of a login and password and **ALL THE FOLLOWING URL CALLS MUST BE DONE to the server name passed back.**

- Send Email Message:

To send a message your app should call the following url with some optional parameters using “post” method:

https://xxxservername_passed_back_from_authentication_xxxxx?kmpostmessage.php
parameters:

```
login =XXXXXXXXXXXX@XXXXXXXX.XXX (if not sent the domain it will be add  
mailclick.com automatically  
password=XXXXXXXXXXXX  
recipients=XXXXXX_recipient's email address_XXXXX  
cc=XXXXXXX_recipient's email addresses separated from comma (,) in carbon copy_XXXX  
bcc=XXXXXXX_recipient's email addresses separated from comma (,) in blind carbon  
copy_XX  
subject=XXXXXX_Subject_text(no html)_XXXXXXX  
msg=XXXXXX_text_message_in html_format_XXXXXXX  
files=xx (optional number of files attached)  
encryped=Y/N (optional default N, if Y the text and the subject are already encrypted in  
base64)  
msgid=xxxxxxxxxxxxxxxx (optional to be used if you've already uploaded some files to link  
them to the message)
```

The engine will send back this messages:

if some parameters are wrong:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR DESCRIPTION</ERROR>  
<ERROR>.....</ERROR>  
</KRYPTOMAIL>
```

An error field will be present for every problem found in the post.

If the message is going to be delivered it will answer:

MAIL1CLICK API - rel 1.35

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
<MSGID>XXXXXXXXXXXXXXXXXX</MSGID>  
</KRYPTOMAIL>
```

where the MSGID is an unique id of the message sent

- Get msgid

To upload files, you need a msgid to link them to the message before to post itself. To get a new msgid you have to call this url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmgetnewmsgid.php

it will send back a message:

```
<KRYPTOMAIL>  
<MSGID>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</MSGID>  
</KRYPTOMAIL>
```

The msgid received can be used to send files attached and to post the related message.

- Send Attached Files:

To send attached files the client should call the following url with some optional parameters:

```
https://xxxservername\_passed\_back\_from\_authentication\_xxxxx/kmsendfiles.php  
?login=xxxxxxxxxxx@xxxxxxxx.xxx  
&password=xxxxxxxxxxx  
&msgid=xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
&urlreturnok=xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
&urlreturnko=xxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

The msgid should be acquired before the upload using the function: kmgetnewmsgid.php

“urlreturnok” is optional and it could be an url to call in case of success;

“urlreturnko” is optional and it could be an url to call in case of error;

It will answer in case of error:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR DESCRIPTION</ERROR>  
</KRYPTOMAIL>
```

or it will send back an html form ready to be used the upload. This html could be inserted in a iframe to be consist in any webpage or used in a browser as it is.

After the upload the application will send back the files name uploaded in this format:

MAIL1CLICK API - rel 1.35

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<FILENAME>XXXXXX_file_name_attached_XXXXXXX</FILENAME>
</KRYPTOMAIL>
```

- Send Attached dfiles DIRECTLY:

It is possible to uploads files without a previous form generated back. In this case the client calls the following url with some optional parameters and a specific encryption type in the form action. This is an html form to explain better the issue:

```
<form enctype="multipart/form-data"
action="https://xxxxservername_passed_back_from_authentication_xxxxx/kmuploadonefile.php" method="POST">
<input type="hidden" name="login" value="xxxxxxxxxxx">
<input type="hidden" name="password" value="xxxxxxxxxxx">
<input type="hidden" name="msgid" value="xxxxxxxxxxxxxxxxxxx">
<input name="userfile[]" type="file" />
</form>
```

The msgid should be acquired before the upload using the function: [kmgetnewmsgid.php](#)
After the upload the application will send back the files name uploaded in this format:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<FILENAME>XXXXXX_file_name_attached_XXXXXXX</FILENAME>
</KRYPTOMAIL>
```

or in case of error:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
</KRYPTOMAIL>
```

This is a working php sample using the curl library:

```
<?php
$url="https://kmengine1.mail1click.com/kmuploadonefile.php";
$post_data['login']="xxxxxxx@mail1click.com"; /*** A VALID EMAIL ACCOUNT
$post_data['password']="xxxxxxxxxxx"; /*** A VALID PASSOWRD
$post_data['msgid']="38DJDUYRE8Y3ADIOWQ8EIAISDJIJ3R3UR9EUAasas2282SJ";
//ANY MSGID COLLECTED BACK FROM THE SERVER
$post_data['userfile']="@/root/test.dat"; /**** ANY FILE AVAILABLE ON THE LOCAL
DISK
```

MAIL1CLICK API - rel 1.35

```
Sch = curl_init();  
curl_setopt($Sch, CURLOPT_URL, $url);  
curl_setopt($Sch, CURLOPT_POSTFIELDS, $post_data);  
curl_setopt($Sch, CURLOPT_RETURNTRANSFER, 1);  
$response = curl_exec($Sch);  
echo $response;  
?>
```

- Copy uploaded File:

In case of message forwarding it's more fast to copy a file instead of download and upload it again. To copy a file uploaded or attached to an existing message your app should call the following url:

```
https://xxxservername\_passed\_back\_from\_authentication\_xxxxx/kmcopyfile.php  
?login=xxxxxxxxxxxxxxxxxxx  
&password=xxxxxxxxxxxxxxxxxxx  
&originmsgid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
&destinationmsgid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
&filename=xxxxxxxx
```

“originmsgid” could be the short numeric value listed from kmlistmessages or the long 64 chars; “destinationmsgid” can only the 64 chars obtained from kmgetnewmsgid.php.

It will answer in case of error:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR DESCRIPTION</ERROR>  
</KRYPTOMAIL>
```

or

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
</KRYPTOMAIL>
```

- Forward Files:

You can prepare all the files attached in the origin file using this url:

```
https://xxxservername\_passed\_back\_from\_authentication\_xxxxx/kmforwardfiles.php  
?login=xxxxxxxxxxxxxxxxxxx  
&password=xxxxxxxxxxxxxxxxxxx  
&originmsgid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

“originmsgid” must be the long 64 chars;

MAIL1CLICK API - rel 1.35

It will answer in case of error:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR DESCRIPTION</ERROR>  
</KRYPTOMAIL>
```

or

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
<MSGID>XXXXXXXX NEW MSG ID MSGXXXX</MSGID>  
</KRYPTOMAIL>
```

- List Uploaded Files:

If you need to get a list of files attached (or forwarded by the previous url) to a msgid, your app should call the following url:

```
https://xxxservername\_passed\_back\_from\_authentication\_xxxxx/kmlistfiles.php  
?login=xxxxxxxxxxxxxxxxxxxx  
&password=xxxxxxxxxxxxxxxx  
&msgid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

“msgid” could be the long msgid (64 chars) or the short numeric id

It will answer in case of error:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR DESCRIPTION</ERROR>  
</KRYPTOMAIL>
```

or

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
<MSGID>XXXXXXXX</MSGID>  
<FILESNUMBER>xx</FILESNUMBER>  
<FILESLIST><FILENAME>XXXXXXXX</FILENAME>  
<FILENAME>XXXXXXXX</FILENAME>  
<FILEID>XXXXXXXX</FILEID>  
<FILENAME>.....</FILENAME>  
<FILEID>.....</FILEID>  
</FILESLIST>  
</KRYPTOMAIL>
```

- Delete File Uploaded:

MAIL1CLICK API - rel 1.35

In the case the user decides to remove the attached file before posting the message, your app should call this url:

[https://xxxservername_passed_back_from_authentication_xxxxx/kmdeletefile.php?login=xxxxxxxxxxx
&password=xxxxxxxxxxxxxxxx
&msgid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
&filename=xxxxxxxxxxxxxxxx](https://xxxservername_passed_back_from_authentication_xxxxx/kmdeletefile.php?login=xxxxxxxxxxx&password=xxxxxxxxxxxxxxxx&msgid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx&filename=xxxxxxxxxxxxxxxx)

The parameters “filename” and “msgid” should be the same used to upload the file.

It will answer in case of error:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR DESCRIPTION</ERROR>  
</KRYPTOMAIL>
```

or

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
</KRYPTOMAIL>
```

-List Messages:

To list the messages available the client should call the following url with some optional parameters:

[https://xxxservername_passed_back_from_authentication_xxxxx?kmlistmessages.php?login=xxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxx
&folder=xx
&nm=xx
&lm=xxxxxx
&decrypted=Y](https://xxxservername_passed_back_from_authentication_xxxxx?kmlistmessages.php?login=xxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxx&folder=xx&nm=xx&lm=xxxxxx&decrypted=Y)

where:

folder is the id for the folder's messages to read, if not set it assumes 0, the default folder are:

0 for inbox, 1 for sent, 2 for trash;

using “all” as value to the folder parameter, you can list the messages from all folders.

nm is the number of maximum message to list if not indicated or 0, it will list all the available messages starting from the most recent and ordered by date/time;

lm is the last message id read and the list sent back will start from the next message in order of date/time receipt, if not passed it will start from the newest message;

MAIL1CLICK API - rel 1.35

decrypted is a flag to send the message decrypted the message data on the server side (optional and can be Y/N)

The server will send back this messages:

if the login or password is wrong:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>LOGIN OR PASSWORD WRONG</ERROR>
</KRYPTOMAIL>
```

if the login and password are correct:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<MSGID>XXXXXXXX</MSGID>
<MESSAGE>
<MSGSENDER>XXXXXXXX@XXXX.XXX</MSGSENDER>
<MSGRECIPIENTS>XXXXXXXX@XXXX.XXX</MSGRECIPIENTS>
<MSGCC>XXXXXXXXXX@XXXXXX.XXXX</MSGCC>
<MSGBCC>XXXXXXXXXX@XXXXXX.XXXX</MSGBCC>
<MSGSUBJECT>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</MSGSUBJECT>
<MSGDT> XXXX/XX/XX XX:XX</MSGDT>
<MSGFILES>XX</MSGFILES>
<MSGREAD>YES</MSGREAD>
<MSGFLAG>YES</MSGFLAG>
</MESSAGE>
.....
.....
</KRYPTOMAIL>
```

If the folder read is empty it will show:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<EMPTY>NO MESSAGES FOUND</EMPTY>
</KRYPTOMAIL>
```

MSGID is the unique id of the message to be used to download the content for example;

MSGSENDER is the email address of the message's sender;

MSGRECIPIENT is the email address of the recipient (To: field in standard email);

MSGSUBJECT is the subject written from the sender (it could be empty!);

MSGDT is the date and time of message in the format year-month-day hour:minutes

MSGFILES contains the number of attached files if available

MSGREAD exists and contains only Y if read.

MSGFLAG is the category message indicated from a letter:

-empty for normal message;

-A "Important"

-B "Work"

MAIL1CLICK API - rel 1.35

- C "Personal"
- D "Todo"
- C "Later"

it can be used to assign a different color on the interface.

- Search Messages:

To search messages in the mailbox, your app should call this url:

https://xxxservername_passed_back_from_authentication_xxxxx/kmsearchmessages.php?login=xxxxxxxxxx&password=xxxxxxxxxxxxxx&sender=xxxxxxxxxx@xxxxxxxxxxx.xxx&recipient=xxxxxxxx@xxxxxxxx.xxx&datefrom=YYYY-MM-DD hh:mm:ss&dateto=YYYY-MM-DD hh:mm:ss&folder=xx&subject=xxxxxxxxxx&msg=xxxxxxxxxxxxxxxxxxxxx

Each field of search can be partial, the date fields are expected in mysql format as indicated.

The parameters "filename" and "msgid" should be the same used to upload the file.

It will answer in case of error:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
</KRYPTOMAIL>
```

or

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<MSGID>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</MSGID>
</KRYPTOMAIL>
```

- Search Messages extended answer:

To search messages in the mailbox and get more information about the messages found, your app should call this url:

https://xxservername_passed_back_from_authentication_xx/kmsearchmessagesextended.php?login=xxxxxxxxxx&password=xxxxxxxxxxxxxx&sender=xxxxxxxxxx@xxxxxxxxxxx.xxx&recipient=xxxxxxxx@xxxxxxxx.xxx&datefrom=YYYY-MM-DD hh:mm:ss

MAIL1CLICK API - rel 1.35

[&dateto=YYYY-MM-DD hh:mm:ss](#)

[&folder=xx](#)

[&subject=xxxxxxxxxx](#)

[&msg=xxxxxxxxxxxxxxxxxx](#)

[&msgsubj=xxxxxxxxxxxxxxxxxx](#)

[&decrypted=Y \(default to decrypt the subject or N to do not decrypt it\)](#)

[Each field of search can be partial, the date fields are expected in mysql format as indicated.](#)

The parameters “filename” and “msgid” should be the same used to upload the file.
The “msgsubj” can be used to find a string in the subject “or” in the message body.

It will answer in case of error:

<KRYPTOMAIL>

<ANSWER>KO</ANSWER>

<ERROR>ERROR DESCRIPTION</ERROR>

</KRYPTOMAIL>

or

<KRYPTOMAIL>

<ANSWER>OK</ANSWER>

<MSGID>XXXXXXXX</MSGID>

<MESSAGE>

<MSGSENDER>[XXXXXXXX@XXX.XXX](#)</MSGSENDER>

<MSGRECIPIENTS>[XXXXXXXX@XXX.XXX](#)</MSGRECIPIENTS>

<MSGCC>[XXXXXXXX@XXXXXX.XXX](#)</MSGCC>

<MSGBCC>[XXXXXXXX@XXXXXX.XXX](#)</MSGBCC>

<MSGSUBJECT>XXXXXXXXXXXXXXXXXXXXXXXXXX</MSGSUBJECT>

<MSGDT> XXXX/XX/XX XX:XX</MSGDT>

<MSGFILES>XX</MSGFILES>

<MSGREAD>YES</MSGREAD>

<MSGFLAG>YES</MSGFLAG>

</MESSAGE>

.....

.....

</KRYPTOMAIL>

- Get Message:

To get a message the client should call the following url with some optional parameters:

https://xxxservername_passed_back_from_authentication_xxxx?kmgetmessage.php?

[login=XXXXXXXX@XXXXXX.XXX](#)

&password=XXXXXXXXXXXX

&msgid=XXXXXXXXXXXXXXXXXXXXXXXXXXXX

&decrypted=Y

where:

msgid is the unique id of the message.

decrypted is a flag to indicate to decrypt the message data on the server side (optional default=N)

The server will answer with:

MAIL1CLICK API - rel 1.35

if some parameters are wrong:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
<ERROR>.....</ERROR>
</KRYPTOMAIL>
```

or

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<MSGID>XXXXXXXX</MSGID>
<MESSAGE>
<MSGSENDER>XXXXXXXX@XXXX.XXX</MSGSENDER>
<MSGRECIPIENTS>XXXXXXXX@XXXX.XXX</MSGRECIPIENTS>
<MSGCC>XXXXXXXX@XXXX.XXX</MSGCC>
<MSGBCC>XXXXXXXX@XXXX.XXX</MSGBCC>
<MSGSUBJECT>XXXXXXXXXXXXXXXXXXXXXXXXXX</MSGSUBJECT>
<MSGTEXT>XXXXXX.....IN HTML FORMAT _XXXXXXX</MSGTEXT>
<MSGDT> XXXX/XX/XX XX:XX</MSGDT>
<MSGFILES>XX</MSGFILES>
<MSGREAD>YES</MSGREAD>
<MSGFLAG>YES</MSGFLAG>
<MSGFOLDER>xx</MSGFOLDER>
</MESSAGE>
<FILESATTACHED>
<FILEID>xxxx_id_xxxx</FILEID><FILENAME>xxx_filename_xxx</FILENAME>
<FILEID>.....</FILEID><FILENAME>.....</FILENAME>
</FILESATTACHED>
</KRYPTOMAIL>
```

The <FILExx> tags will be present only if there is almost one attached file.

- Download Attached Files:

To download a file the application should call the following url and parameters:

```
https://xxxservername\_passed\_back\_from\_authentication\_xxxx?kmdownloadfile.php?login=XXXXXXXX@XXXXXXXX.XXXpassword=XXXXXXXXXX&id=XXXXXXXXXX
```

where:

id is the unique id of the file obtained from kmgetmessage.php
the server will answer with:

if some parameters are wrong:

MAIL1CLICK API - rel 1.35

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
<ERROR>.....</ERROR>
</KRYPTOMAIL>
```

or it will download the file decrypted and setting the right mime/type in the html header sent back.

- Move Message (to a different folder)

To move a message to a different folder the client should call the following url with some optional parameters:

```
https://xxxservername_passed_back_from_authentication_xxxxx?kmmovemessage.php
&login=xxxxxxxxx$password=xxxxxxxxxxxxx
&msgid=xxxxxxxxxxxx&folderid=xx
```

The msgid is the number got from kmlistmessages.php

The folder id can be get from kmlistfolders.php

If some parameters are wrong:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
<ERROR>.....</ERROR>
</KRYPTOMAIL>
```

if correctly moved:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>
```

- Move Messages (to a different folder)

To move more messages to a different folder the client should call the following url with some optional parameters:

```
https://xxxservername_passed_back_from_authentication_xxxxx?kmmovemessages.php
&login=xxxxxxxxx$password=xxxxxxxxxxxxx
&folderid=xx
&msgid_1=xxxxxxxxxxxxx
&msgid_2=xxxxxxxxxxxxx
&msgid_...=xxxxxxxxxxxxx
```

The msgid_XXX is the number got from kmlistmessages.php, you should insert one msgid_XXX var (XXX is a progressive number starting from 1) for each message that you want delete.

MAIL1CLICK API - rel 1.35

The folder id can be get from kmlistfolders.php

If some parameters are wrong:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
<ERROR>.....</ERROR>
</KRYPTOMAIL>
```

if correctly moved:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>
```

- Delete Message (no possibility to recover it!)

To delete a message the client should call the following url with some optional parameters:

```
https://xxxservername_passed_back_from_authentication_xxxxx?kmdeletemessage.php?
login=xxxxxxxxxx
&password=xxxxxxxxxxxxx
&msgid=xxxxxxxxxxxxx
```

The msgid is the number go from kmlistmessages.php

If some parameters are wrong:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
<ERROR>.....</ERROR>
</KRYPTOMAIL>
```

if correctly moved:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>
```

- Delete Messages (no possibility to recover it!)

To delete more messages in one call, the client should call the following url with some optional parameters:

```
https://xxxservername_passed_back_from_authentication_xxxxx?kmdeletemessages.php?
login=xxxxxxxxxx
&password=xxxxxxxxxxxxx
```

MAIL1CLICK API - rel 1.35

&msgid_1=XXXXXXXXXXXXX
&msgid_2=XXXXXXXXXXXXX
&msgid_.....=XXXXXXXXXXXXX

The msgid_XXX is the number got from kmlistmessages.php, you should insert one msgid_XXX var (XXX is a progressive number starting from 1) for each message that you want delete.

If some parameters are wrong:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
<ERROR>.....</ERROR>
</KRYPTOMAIL>

if correctly moved:

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>

- Set Message Flag

To set the message flag you should call the following url:

[https://xxxservername_passed_back_from_authentication_xxxxx?kmsetmessageflag.php?](https://xxxservername_passed_back_from_authentication_xxxxx?kmsetmessageflag.php?login=XXXXXXXXXX&password=XXXXXXXXXXXXX&msgid=XXXXXXXXXXXXX&flag=X)
login=XXXXXXXXXX
&password=XXXXXXXXXXXXX
&msgid=XXXXXXXXXXXXX
&flag=X

The msgid can be the short number or the long 64 chars version.

The flag can be a letter from A to Z or nothing to reset.

As convention we use only the following letters:

A = Important

B = Work

C = Personal

D = Todo

E = Later

from F to M are reserved for future general use.

From N to Z can be used from specific customers application.

If some parameters are wrong:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR DESCRIPTION</ERROR>
<ERROR>.....</ERROR>
</KRYPTOMAIL>

MAIL1CLICK API - rel 1.35

if correctly changed:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>
```

- Get MSGID (You need it to Send files)

To upload files, you need a msgid to link them to the message. To get a new msgid you have to call this url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmgetnewmsgid.php

it will send back a message:

```
<KRYPTOMAIL>
<MSGID>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</MSGID>
</KRYPTOMAIL>
```

- Folder list/create/rename/delete:

To list the existing folders the app should call this url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmlistfolders.php?login=xxxxxxxxxxx@xxxxxxxxxxx.xxx&password=xxxxxxxx

The answer will be:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<FOLDER>
<UPPERFOLDER>0(or 1)</UPPERFOLDER>
<ID>X</ID>
<DESCRIPTION>XXXXXXXXXXXXXXXXXXXX</DESCRIPTION>
</FOLDER>
</KRYPTOMAIL>
```

or in case of errors:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ANY ERROR MESSAGE</ERROR>
.....(more ERROR TAG if present more errors).....
</KRYPTOMAIL>
```

To rename a folder the app should call this url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmrenamefolder.php?login=xxxxxxxxxxx@xxxxxxxxxxx.xxx&password=xxxxxxxx

MAIL1CLICK API - rel 1.35

&id=xxxx
&newdesc=XXXXXXXXXXXXXXXXXXXXXXXXXXXX

The answer will be

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<ERROR>FOLDER RENAMED</ERROR>
</KRYPTOMAIL>

In case of error:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ANY ERROR MESSAGE</ERROR> (ORIGINA FOLDER NOT
PRESENT/DESTINATION FOLDER ALREADY USED/NO MORE SPACE/GENERIC ERROR)
</KRYPTOMAIL>

To add a folder the app should call this url:

**[https://xxxservername_passed_back_from_authentication_xxxxx?kmaddfolder.php?
login=XXXXXXXXXX@XXXXXXXXX.XXX
&password=XXXXXXX
&description=XXXXXXX
&upperfolder=0/1](https://xxxservername_passed_back_from_authentication_xxxxx?kmaddfolder.php?login=XXXXXXXXXX@XXXXXXXXX.XXX&password=XXXXXXX&description=XXXXXXX&upperfolder=0/1)**

- description can be long max 25 chars and cannot be empty;
- upperfolder should be 1 for a folder under “Sent” folder or 0 under “Inbox” folder (default is 0 if not set)

The answer will be

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>

or in case of error:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ANY ERROR MESSAGE</ERROR>
.....(more ERROR TAG if present more errors).....
</KRYPTOMAIL>

To delete a folder the app should call this url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmdeletefolder.php?

MAIL1CLICK API - rel 1.35

login=XXXXXXXXXX@XXXXXXXXX.XXX
&password=XXXXXXXX
&id=XXXXXXXXXXXXXXXXXX

where folder is the folder description as created, the standard folder inbox,sent and trash cannot be deleted.

The answer will be
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>

In case of error:
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ANY ERROR MESSAGE</ERROR> (FOLDER NOT PRESENT/GENERIC ERROR)
</KRYPTOMAIL>

The messages will not be deleted, they'll be moved to the upper folder.

To empty the trash folder the app should call this url:
[https://xxxservername_passed_back_from_authentication_xxxx?kmemptytrash.php?](https://xxxservername_passed_back_from_authentication_xxxx?kmemptytrash.php?login=XXXXXXXXXX@XXXXXXXXX.XXX&password=XXXXXXXX)
login=XXXXXXXXXX@XXXXXXXXX.XXX
&password=XXXXXXXX

The answer will be
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>

In case of error:
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ANY ERROR MESSAGE</ERROR>
</KRYPTOMAIL>

The messages deleted cannot be recovered in any way.

To list the unread messages the app should call this url:
[https://xxxservername_passed_back_from_authentication_xxxx?](https://xxxservername_passed_back_from_authentication_xxxx?kmunreadmessagesbyfolder.php?login=XXXXXXXXXX@XXXXXXXXX.XXX&password=XXXXXXXX)
kmunreadmessagesbyfolder.php?login=XXXXXXXXXX@XXXXXXXXX.XXX
&password=XXXXXXXX

The answer will be
<KRYPTOMAIL>

MAIL1CLICK API - rel 1.35

<ANSWER>OK</ANSWER>
<FOLDER>xx</FOLDER>
<UNREAD>xxx</UNREAD>
</KRYPTOMAIL>

In case of error:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ANY ERROR MESSAGE</ERROR>
</KRYPTOMAIL>

- Password change:

To change the key password the app should call this url:

[https://xxxservername_passed_back_from_authentication_xxxx?kmpasswordchange.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx
&newpassword=xxxxxxxxxxxxxxxxxxxxxxxx](https://xxxservername_passed_back_from_authentication_xxxx?kmpasswordchange.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx&newpassword=xxxxxxxxxxxxxxxxxxxxxxxx)

The answer will be:

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>

or:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>

- Set Language:

To change the user's default language should call this url:

[https://xxxservername_passed_back_from_authentication_xxxx?kmsetlanguage.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx
&language=xx](https://xxxservername_passed_back_from_authentication_xxxx?kmsetlanguage.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx&language=xx)

where acceptable language are:

EN for english

IT for italian

HI fo Hindi

The answer will be:

MAIL1CLICK API - rel 1.35

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>

or:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>

- Get Language:

To GET the user's default language should call this url:

[https://xxxservername_passed_back_from_authentication_xxxxx?kmgetlanguage.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx
&language=xx](https://xxxservername_passed_back_from_authentication_xxxxx?kmgetlanguage.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx&language=xx)

The answer will be:

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<LANGUAGE>XX</LANGUAGE>
</KRYPTOMAIL>

or:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>

- Set TimeZone:

To change the user's default language should call this url:

[https://xxxservername_passed_back_from_authentication_xxxxx?kmsettimezone.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx
&tz=xxx](https://xxxservername_passed_back_from_authentication_xxxxx?kmsettimezone.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx&tz=xxx)

where tz can be a number in minutes of offset from utc time to your local time. The valid value is between 720 and -720.

For example if you are leaving in UAE, you are 4 hours forward to the UTC time so your offset will be -240

The answer will be:

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>

MAIL1CLICK API - rel 1.35

</KRYPTOMAIL>

or:

<KRYPTOMAIL>

<ANSWER>KO</ANSWER>

<ERROR>ERROR MESSAGE</ERROR>

</KRYPTOMAIL>

- Get TimeZone (offset from UTC):

To GET the user's TIME ZONE offset should call this url:

[https://xxxservername_passed_back_from_authentication_xxxxx?kmgettimezone.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx](https://xxxservername_passed_back_from_authentication_xxxxx?kmgettimezone.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx)

The answer will be:

<KRYPTOMAIL>

<ANSWER>OK</ANSWER>

<UTCOFFSET>XXX</UTCOFFSET>

</KRYPTOMAIL>

or:

<KRYPTOMAIL>

<ANSWER>KO</ANSWER>

<ERROR>ERROR MESSAGE</ERROR>

</KRYPTOMAIL>

- Set Other Email:

To GET the user's alternative email used to send automatic alert when a message arrive in the mailbox, use this url:

[https://xxxservername_passed_back_from_authentication_xxxxx?kmsetotheremail.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx
&otheremail=xxxxxxxxxxxxxxxx@xxxxxxxx.xxx](https://xxxservername_passed_back_from_authentication_xxxxx?kmsetotheremail.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx&otheremail=xxxxxxxxxxxxxxxx@xxxxxxxx.xxx)

The answer will be:

<KRYPTOMAIL>

<ANSWER>OK</ANSWER>

</KRYPTOMAIL>

or:

<KRYPTOMAIL>

<ANSWER>KO</ANSWER>

<ERROR>ERROR MESSAGE</ERROR>

MAIL1CLICK API - rel 1.35

</KRYPTOMAIL>

- Get Other Email:

To GET the user's alternative email used to send automatic alert when a message arrive in the mailbox, use this url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmgetotheremail.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx

The answer will be:

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<OTHEREMAIL>XX</OTHEREMAIL>
</KRYPTOMAIL>

or:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>

- Get Email Alert:

To READ the user's email alert (Y or N) , you should call the following url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmgetemailalert.php?login=xxxxxxxxxxx@xxxxxxxx.xxxx&password=xxxxxxxxxxx

The answer will be:

<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<ALERT>Y(N)</ALERT>
</KRYPTOMAIL>

or:

<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>

- Set Email Alert:

To SET the user's email alert to Y or N , you should call the following url:

https://xxxservername_passed_back_from_authentication_xxxxx?kmsetemailalert.php?

MAIL1CLICK API - rel 1.35

**login=XXXXXXXXXX@XXXXXXXX.XXX
&password=XXXXXXXXXX
&emailert=X**

where X is N or Y (for no or yes)

The answer will be:

**<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
</KRYPTOMAIL>**

or:

**<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>**

- List Contacts

To get the contact list , you should call the following url:

**[https://xxxservername_passed_back_from_authentication_xxxx?kmlistcontacts.php?
login=XXXXXXXXXX@XXXXXXXX.XXX
&password=XXXXXXXXXX
&search=XXXXXXXXXXXXXXXXXX](https://xxxservername_passed_back_from_authentication_xxxx?kmlistcontacts.php?login=XXXXXXXXXX@XXXXXXXX.XXX&password=XXXXXXXXXX&search=XXXXXXXXXXXXXXXXXX)**

Where the optional “search” is a string to be searched fort in the name or in the email address.

The answer will be:

**<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<CONTACT>
<EMAIL>XXXXXXXX@XXXXXXXX.XXX</EMAIL>
<NAME>XXXXXXXXXXXXXXXXXXXXXXXXXX</NAME>
</CONTACT>s
</KRYPTOMAIL>**

or:

**<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>**

- Add Contact

To add a contact to the contact list , you should call the following url:

MAIL1CLICK API - rel 1.35

[https://xxxservername_passed_back_from_authentication_xxxx?kmaddcontact.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx
&email=xxxxxxxxxxxxxxxx@xxxxxxxx.xxx
&name=xxxxxxxxxxxxxxxx](https://xxxservername_passed_back_from_authentication_xxxx?kmaddcontact.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx&email=xxxxxxxxxxxxxxxx@xxxxxxxx.xxx&name=xxxxxxxxxxxxxxxx)

Where email is email address to add and name is the optional name of the contact.

The answer will be:

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
</KRYPTOMAIL>
```

or:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR MESSAGE</ERROR>  
</KRYPTOMAIL>
```

- Delete Contact

To delete a contact from the contact list , you should call the following url:

[https://xxxservername_passed_back_from_authentication_xxxx?kmdeletecontact.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxxxxx
&email=xxxxxxxxxxxxxxxx@xxxxxxxx.xxx](https://xxxservername_passed_back_from_authentication_xxxx?kmdeletecontact.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxxxxx&email=xxxxxxxxxxxxxxxx@xxxxxxxx.xxx)

Where email is email address to delete.

The answer will be:

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
</KRYPTOMAIL>
```

or:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR MESSAGE</ERROR>  
</KRYPTOMAIL>
```

- Get Public key:

To get the public key of a recipient the app should call this url:

https://xxxservername_passed_back_from_authentication_xxxx?kmpublickey.php?

MAIL1CLICK API - rel 1.35

login=xxxxxxxxx
&password=xxxxxxxxxxx
&email=xxxxxxxxxxxxxxxxxxxxxxxxxxxx

The answer will be:

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
<PUBLICKEY>XXXXXXXXXXXXXXXXXXXX</PUBLICKEY>  
</KRYPTOMAIL>
```

If the email address is not present it will have assigned a public key in any case.
The application will answer always with a public key.

In case of error:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR MESSAGE</ERROR>  
</KRYPTOMAIL>
```

- Encrypt Text:

This function allows to encrypt any text using the server crypto engine with the public key of kryptomail user. The app should call this url:

https://xxxservername_passed_back_from_authentication_xxxx?kmencrypttext.php
using post or get method with the following parameters:
to=xxxxxxxx@xxxxxxxx.xxx is the email address recipient in the kryptomail domain name;
text=xxxxxxxxxxxxxxxxxxxxxxxxxxxx is the text to encrypt using the recipient public key

In case of error:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR MESSAGE</ERROR>  
</KRYPTOMAIL>
```

if the text has been encrypted it will return the following message:

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
<ENCRYPEDTEXT>XXXXXXXX</ENCRYPTEDTEXT>  
</KRYPTOMAIL>
```

The encrypted text is sent back in a format base64 to be easily used everywhere without worry about binary chars.

- Decrypt Text:

MAIL1CLICK API - rel 1.35

This function allows to decrypt any text previously encrypted with the public key.
the app should call this url:

https://xxxxservername_passed_back_from_authentication_xxxxx?kmdecrypttext.php

using post or get method with the following parameters:

login=xxxxxxxx@xxxxxxxx.xxx is the login name of the recipients of the encrypted text

password=xxxxxxxx is the password of the recipients for the encrypted text

text=xxxxxxxxxxxxxxxxxxxx is the encrypted text to be decrypted using the user's private key.
(remember to url encode!)

In case of error:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>
```

if the text has been decrypted it will return the following message:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<DECRYPEDTEXT>XXXXXXXX</DECRYPTEDTEXT>
</KRYPTOMAIL>
```

- Get Info about the Subscription

To get information about the active subscription , you should call the following url:

<https://kmengine.mailclick.com/kmgetinfosubscription.php?>

[login=xxxxxxxxxxx@xxxxxxxx.xxx](#)

[&password=xxxxxxxxxxx](#)

The answer will be:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<BYTESALLOCATED>xxxxxxxx</BYTESALLOCATED>
<BYTESUSED>xxxxxxxx</BYTESUSED>
<EXPIRINGDATEPREMIUM>xxx-xx-xx</EXPIRINGDATEPREMIUM>
</KRYPTOMAIL>
```

or:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>
```

- List Messages waiting for validation

MAIL1CLICK API - rel 1.35

The messages coming from an unknow email address stay in a waiting queue until the sender don't insert a captcha code following a link he received back in answer.
This function allows to list the messages that are still waiting for validation.

[https://kmengine.mailclick.com/kmlistmessageswaitingvalidation.php?
login=XXXXXXXXXX@XXXXXXXX.XXXX
&password=XXXXXXXXXX](https://kmengine.mailclick.com/kmlistmessageswaitingvalidation.php?login=XXXXXXXXXX@XXXXXXXX.XXXX&password=XXXXXXXXXX)

The answer will be:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<MESSAGE>
<MSGID>XXXXXXXX</MSGID>
<MSGSENDER>XXXXXXXX@XXXX.XXX</MSGSENDER>
<MSGRECIPIENTS>XXXXXXXX@XXXX.XXX</MSGRECIPIENTS>
<MSGCC>XXXXXXXX@XXXX.XXX</MSGCC>
<MSGBCC>XXXXXXXX@XXXX.XXX</MSGBCC>
<MSGSUBJECT>XXXXXXXXXXXXXXXXXXXXXXXXXX</MSGSUBJECT>
<MSGBODY>XXXXXXXXXXXXXXXXXXXXXXXXXX</MSGBODY>
<MSGDT> XXXX/XX/XX XX:XX</MSGDT>
<MSGFILES>XX</MSGFILES>
</MESSAGE>
.....
.....
</KRYPTOMAIL>
```

If the folder read is empty it will show:

```
<KRYPTOMAIL>
<ANSWER>OK</ANSWER>
<EMPTY>NO MESSAGES FOUND</EMPTY>
</KRYPTOMAIL>
```

or:

```
<KRYPTOMAIL>
<ANSWER>KO</ANSWER>
<ERROR>ERROR MESSAGE</ERROR>
</KRYPTOMAIL>
```

- Validate Message waiting for validation

This function allows to validate a message that are still waiting for validation.

[https://kmengine.mailclick.com/kmvalidatemessagewaitingvalidation.php?
login=XXXXXXXXXX@XXXXXXXX.XXXX
&password=XXXXXXXXXX](https://kmengine.mailclick.com/kmvalidatemessagewaitingvalidation.php?login=XXXXXXXXXX@XXXXXXXX.XXXX&password=XXXXXXXXXX)

MAIL1CLICK API - rel 1.35

[&msgid=xxxxxxxxxxxxxxxxxxxxxx](#) (the msg read from kmlistmessageswaitingvalidation)

The answer will be:

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
</KRYPTOMAIL>
```

or:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR MESSAGE</ERROR>  
</KRYPTOMAIL>
```

- Delete Message waiting for validation

This function allows to list the messages that are still waiting for validation.

[https://kmengine.mail1click.com/kmdeletemessagewaitingvalidation.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxx
&msgid=xxxxxxxxxxxxxxxxxxxxxx](https://kmengine.mail1click.com/kmdeletemessagewaitingvalidation.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxx&msgid=xxxxxxxxxxxxxxxxxxxxxx) (the msg read from kmlistmessageswaitingvalidation)

The answer will be:

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
</KRYPTOMAIL>
```

or:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR MESSAGE</ERROR>  
</KRYPTOMAIL>
```

- Register a device for Apple Push messages (only iOS and Mac Os/x)

This function allows the registration of device for receiving Apple Push messages when a new messages is arrived.

[https://kmengine.mail1click.com/kmserver_apnregister.php?
login=xxxxxxxxxxx@xxxxxxxx.xxx
&password=xxxxxxxx \(use the token or the password\)
&tokenapn=xxxxxxxxxxxxxxxxxxxxxx](https://kmengine.mail1click.com/kmserver_apnregister.php?login=xxxxxxxxxxx@xxxxxxxx.xxx&password=xxxxxxxx&tokenapn=xxxxxxxxxxxxxxxxxxxxxx) (it's the apn token release from Apple registration on the device)

MAIL1CLICK API - rel 1.35

The answer will be:

```
<KRYPTOMAIL>  
<ANSWER>OK</ANSWER>  
</KRYPTOMAIL>
```

or:

```
<KRYPTOMAIL>  
<ANSWER>KO</ANSWER>  
<ERROR>ERROR MESSAGE</ERROR>  
</KRYPTOMAIL>
```
