



MOBBIS LLC , 9/30/2019

HTTP API 2.0

Functional Specification



1.	Introduction	3
1.1.	Document identifier	3
1.2.	Scope.....	3
1.3.	Definition of terms.....	3
2.	References	3
3.	Overview	3
4.	HTTP API interface description	4
5.	Description of the supported functions.....	5
5.1.	FUNCTION “send”	5
5.2.	FUNCTION “schedule”	5
5.3.	FUNCTION “querybalance”	5
5.4.	FUNCTION “querymessage”	5
6.	HTTP/S requests description.....	6
7.	Preconditions, restrictions	7
7.1.	Preconditions	7
7.2.	Restrictions	7
8.	Appendix	8
8.1.	SMS Status values.....	8
8.2.	Error codes	9
9.	Examples	10
9.1.	C#.NET.....	10
9.2.	Visual Basic 2005	10
9.3.	PHP 5.2.1	10
9.4.	VBScript	11

1. Introduction

1.1.Document identifier

This is the specification of the HTTP API 2.0 implemented by MOBBIS LLC.

1.2.Scope

This document contains the definition of the HTTP API, description of the HTTP requests and the possible values of return codes to provide detailed information concerning the status of the request sent to the server. It also contains the definition of the terms and acronyms used in the text and references to the useful links.

1.3.Definition of terms

HTTPS – see , <http://en.wikipedia.org/wiki/HTTPS>

SMS – see [http://en.wikipedia.org/wiki/SMS \(messaging\)](http://en.wikipedia.org/wiki/SMS_(messaging))

MOBIPACE – see <http://www.mobipace.com>

ASCII - <http://en.wikipedia.org/wiki/ASCII>

UNICODE - <http://en.wikipedia.org/wiki/Unicode>

2. References

[1]. <http://www.mobipace.com>

[2]. <http://www.mobbis.mobi>

[3]. <http://www.mobbis.am>

[4]. <http://en.wikipedia.org/wiki/Unicode>

[5]. <http://en.wikipedia.org/wiki/ASCII>

3. Overview

HTTP API is platform independent HTTP/S protocol based protocol which allows to connect/work directly to/with MOBIPACE server, and provides an opportunity to send the messages (SMS, WapPush, MMS) using simple HTTP requests through MOBBIS SMS gateway to the cell phones.

User can develop the front-end for the API independent of platform, operation system, and programming language and send SMS to more than 200 countries of the world through MOBBIS gateway.

HTTP API consists of simple HTTP requests which can be also processed e.g. by using any Web browser (IE, Mozilla Firefox, Opera...).

The responses of the requests contain detailed information about the status of the request and the code indicating the current status.

If the program doesn't support UNICODE format, requests must contain only Latin symbols. Otherwise your message will be damaged and won't be delivered to the recipient.

Full list of countries and operators to which is possible to send messages can be found on <http://www.mobipace.com> .

4. HTTP API interface description

HTTP API consists of the HTTP/S requests, which common format is:

https://www.mobipace.com/API_2_0/HTTP_API.aspx?function=FUNCTION_NAME&username=USER_NAME&password=PASSWORD&FUNCTION_PARAM_1=VALUE_1&...&FUNCTION_PARAM_N=VALUE_N,

where,

- FUNCTION_NAME is the name of corresponding request (mandatory option)
- USER_NAME/PASSWORD is the registered user login/password (mandatory option)
- FUNCTION_PARAM_N is n-th parameter of corresponding request
(Required depending on the value of FUNCTION_NAME)
- VALUE_N is a value of the FUNCTION_PARAM_N
(Required depending on the value of FUNCTION_NAME)

The functionality defined by the value of the parameter “function” (see section 6 for details).

All requests require verification by user name and password (the following part of the request is common for all) which gives an opportunity send request anytime you need instead of implementing the permanent connection with the MOBBIS gateway:

https://www.mobipace.com/API_2_0/HTTP_API.aspx?function=FUNCTION_NAME&username=USER_NAME&password=PASSWORD ...

Example:

https://www.mobipace.com/API_2_0/HTTP_API.aspx?function=querybalance&username=37491xxxxxx&password=xxxxxx

The MOBIPACE server processes this requests and returns corresponding response to the client either error message or info message.

The common format of the error messages is:

“Err: <error_code> : <Info message>”, where the <error_code> indicates the type of the issue (8.2), <Info message> contains the error message in text form.

The common format of the info messages is:

“ID: <value> : <Info message>”, where the <value> indicates the type of the info (8.2), <Info message> contains the error message in text form.

5. Description of the supported functions

This section describes all operations, which can be performed through API requests.

The detailed descriptions of the corresponding HTTP/S functions and requests given in the section [6](#).

The descriptions of the error/info codes, which can be returned by the API, described in the section [8.2](#).

5.1. FUNCTION “send”

This function used to send the message through the MOBBIS gateway to the mobile phone.

It is possible to specify the type of the message – Wap Push or SMS.

All the messages have their own unique message reference codes (ID), which can be used in future to request the status of the message. Response of the sent request is either message reference of the message generated by the API automatically, or the error code in case of any errors.

It is possible to send both UNICODE and ASCII messages (Latin symbols).

If the size of one message exceeds 160 latin symbols or 70 unicode symbols (the default maximum length of the SMS message), the message will be automatically split into corresponding count of messages and then sent to the recipient. In this case, recipient will receive one message, but you will be charged for many.

Note, that the message will be considered as UNICODE message if it contains even one UNICODE symbol.

5.2. FUNCTION “schedule”

Schedule the message to send on specified date/time to the specified recipient. Note, that the date/time should be specified in UTC in according to “Sortable date/time pattern” (e.g. 2009-06-15T13:45:30). The function returns the message reference of the scheduled message to handle the message in the future. Or, error info if there are some issues.

If the size of one message exceeds 160 symbols (the maximum length of the SMS message), the message will be automatically split into corresponding count of messages and then sent to the recipient sequentially.

It is possible to send both UNICODE and ASCII messages.

Note, that the message will be considered as UNICODE message if it contains even one UNICODE symbol. In this case the count of the messages can be increased.

Note, for registered individual persons the sender field of the message will be set to username (cell phone number).

5.3. FUNCTION “querybalance”

Return the current balance of the user in the type of the currency set for the specified user, or error code in case of any issues.

5.4. FUNCTION “querymessage”

Return the status of the message by the specified message reference, or error code in case of any issues.

6. HTTP/S requests description

The table below lists the all supported values of the parameter “function” and the list of the HTTP request parameters, which are necessary to successfully process request.

Function	Parameters	Response	Description
send	<ol style="list-style-type: none"> username - Username of the registered user. password - password of the specified user. sender – the name of the sender recipient – the phone number of the recipient. This string should contain only digits. body – ASCII or UNICODE text wap_push – specifies whether the message should be considered as Wap Push message. Will be considered as false if not specified 	<p>Message ID by the following syntax “ID: xxxx” if gateway accepts the SMS, otherwise error code by the following syntax “Er: code : <i>error text</i>”.</p>	<p>Sends SMS text from the specified sender to the specified mobile, and decreases the balance of the current user.</p> <p>Note: sender from which you are sending message must be registered and accepted by MOBBIS, otherwise system will reject SMS</p> <p>wap_push - parameter is optional</p>
schedule	<ol style="list-style-type: none"> username - Username of the registered user. password - password of the specified user. sender – the name of the sender recipient – the phone number of the recipient. This string should contain only digits. body – ASCII or UNICODE text wap_push – specifies whether the message should be considered as Wap Push message. Will be considered as false if not specified sendingdate — the scheduled date/time in UTC according to “Sortable date/time pattern” (e.g. 2009-06-15T13:45:30). 	<p>Message ID by the following syntax “ID: xxxx” if system successfully schedules the message, otherwise error code by the following syntax “Err: code : <i>error text</i>”.</p>	<p>Schedules the SMS message to send on the specified date/time</p> <p>Note: sender from which you are sending message must be registered and accepted by MOBBIS, otherwise system will reject SMS</p> <p>wap_push - parameter is optional</p>

Function	Parameters	Response	Description
querybalance	<ol style="list-style-type: none"> 1. username - Username of the registered user. 2. password – password of the specified user. 	The balance of the specified user by the following syntax “ID: balance : <i>info text</i> ” if user is registered, otherwise error code (<i>Err: code : Error message</i>).	Returns the balance of the specified user.
querymessage	<ol style="list-style-type: none"> 1. username - Username of the registered user. 2. password - password of the specified user. 3. messageref – the reference (ID) of the message which status should be returned 	Status ID of the SMS by the following syntax “ID: statusID : <i>Info message</i> ” or error code by the following syntax “ <i>Err: code : Error message</i> ”. The list of the possible status values are described in section 8.1	Returns the status of the message

7. Preconditions, restrictions

7.1.Preconditions

It is required to sign an agreement with MOBBIS LLC for registering a new account as well as registering Sender name/s for outgoing messages.

7.2.Restrictions

HTTP API 2.0 is copyrighted. This product is free only for using.

HTTP API 2.0 can be used in your commercial or noncommercial products.

8. Appendix

8.1. SMS Status values

Status	Value	Description
Pending	0	The status of the SMS message is unknown, SMS is in queue.
Sent	1	The SMS message is sent and delivered.
DeliveredToNetwork	2	The SMS message is delivered to operator.
TimeOut	3	The connection is interrupted, operator may be overloaded.
OutOfBalance	4	Out of balance, SMS is not delivered.
Scheduled	5	The SMS message will be sent at the scheduled time.
DeliveryToGateway	6	The SMS message is delivered to gateway but status unknown.
IncorrectRecipient	7	The format of the recipient is incorrect.
NotSent	8	Wrong destination or internal error.
Unknown	9	Incorrect message reference (ID)

8.2. Error codes

Error	Code	Description
IncorrectParameters	700	Incorrect parameters in user's request
AuthenticationFailed	701	Incorrect username or password
OutOfBalance	702	Not enough balance to send SMS message
IncorrectMessageRef	703	Incorrect message references
IncorrectDateTimeFormat	705	The format of the date/time is incorrect
OperationSucceed	706	The operation (activate, register) has been processed successfully
IncorrectActivationCode	707	The activation code is incorrect
OutOfDateTime	708	The specified date/time (for function 'schedule') is out of date
IncorrectCurrency	709	The specified currency is incorrect
IncorrectEmailFormat	710	The format of the specified email is incorrect
UserIsAlreadyRegistered	711	User account with specified username is already registered
OperationTimeOut	712	The message send operation timed out
UnknownError	713	Indicates an error occurred in the system
EmailIsAlreadyUsed	714	Indicates that user account with the specified email is already registered

NOTE: The errors 712 (OperationTimedOut) and 713 (Unknown) are unique errors which can occur inside of the MOBIPACE system. The text followed after those codes in the response specify the type of the issue (e.g. "Unable to connect to router").

In case you encounter these errors, please send the response message with the corresponding request to MOBIPACE support team (support@mobipace.com).

9. Examples

9.1.C#.NET

```
public static string Send(string pSender, string pRecipient, string pBody,
                        string UserName, string Password)
{
    WebClient web = new WebClient();
    web.Proxy = null;
    web.QueryString.Add("function", "send");
    web.QueryString.Add("username", UserName);
    web.QueryString.Add("password", Password);
    web.QueryString.Add("sender", pSender);
    web.QueryString.Add("recipient", pRecipient);
    web.QueryString.Add("body", pBody);

    string default_url = "https://www.mobipace.com/API_2_0/HTTP_API.aspx";
    return web.DownloadString(default_url);
}
```

9.2. Visual Basic 2005

```
Public Function Send(ByVal pSender As String, ByVal pRecipient As String,
                    ByVal pBody As String, ByVal UserName As String,
                    ByVal Password As String) As String
    Dim web As WebClient = New WebClient()
    web.Proxy = Nothing
    web.QueryString.Add("function", "send")
    web.QueryString.Add("username", UserName)
    web.QueryString.Add("password", Password)
    web.QueryString.Add("sender", pSender)
    web.QueryString.Add("recipient", pRecipient)
    web.QueryString.Add("body", pBody)

    Dim default_url As String = "https://www.mobipace.com/API_2_0/HTTP_API.aspx"
    Return web.DownloadString(default_url)
End Function
```

9.3.PHP 5.2.1

```
function Send($pSender, $pRecipient, $pBody, $UserName,$Password)
{
    $default_url = "https://www.mobipace.com/API_2_0/HTTP_API.aspx";
    $data = array('function'=>'send',
                  'username'=>$UserName,
                  'password'=>$Password,
                  'sender'=>$pSender,
                  'recipient'=>$pRecipient,
                  'body'=>urlencode(mb_convert_encoding($pBody, "ISO-8859-1")));
    $url = $default_url."?.http_build_query($data);

    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
    $d = curl_exec($ch);
    curl_close($ch);
    return $d;
}
```

```
}
```

9.4.VBScript

```
Function Send(pSender, pRecipient, pBody, UserName,Password)
    const SXH_OPTION_URL_CODEPAGE = 0

    Set sxh = CreateObject("MSXML2.ServerXMLHTTP.3.0")
    sxh.setOption SXH_OPTION_URL_CODEPAGE, "ISO-8859-1"
    URL = "https://www.mobipace.com/API_2_0/HTTP_API.aspx?function=send
    &username="&UserName&"&password="&Password&"&sender="&pSender&"&recipient="&pRecipient&
    "&body="&pBody
    sxh.open "GET", URL, FALSE
    sxh.send

    WScript.Echo sxh.responseText
    set sxh = nothing
End Function
```