BIP KAMPÁNY

API documentation

Version: 2.02
Date: 2016-02-12

Important - first please turn on your API access in the Settings menu after logging in!

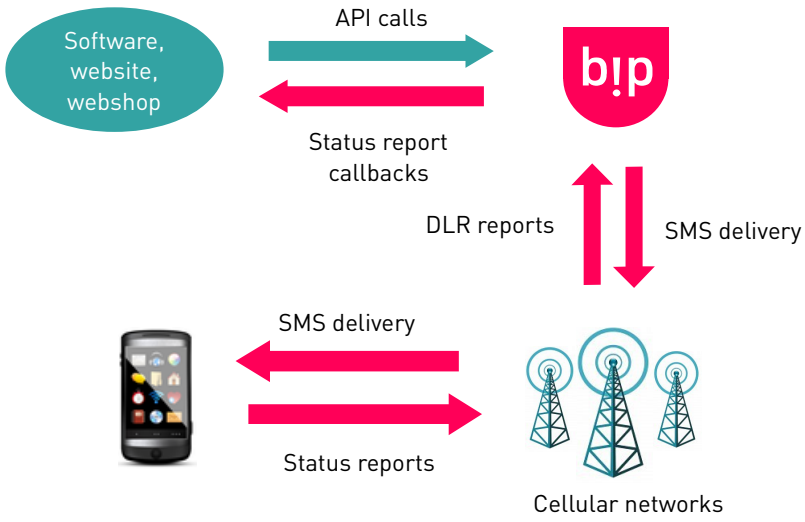
Should you have any problem, question or suggestions, we're here to help at info@bipkampany.hu!

Table of contents

1 About the API

The API is an easy way connect your own service, software, website or webshop to BIP SMS and make it SMS capable.

The overview of the API workflow:



1.1 The settings of your API access

Before using the API first you'll need to turn it on and finetune according to your needs:

- log in and open Settings in the menu. Find *API hozzáférés engedélyezve* and set it *yes*.
- **IP settings:** to secure your account you can use IP settings to restrict API access to specific IP addresses. This is a recommended setting and will help to lower the risks of frauduluent access.
- **Callback URL:** the BIP SMS service is able to provide SMS status reports using simple HTTP calls. The URL where you're about to accept such calls can be set in the *Callback URL* field. By clicking the *Callback teszt kérése* button you can initiate a test call with the URL: this tool will display the URL being called and also the answer received by BIP (including HTTP headers and the content itself). This is as an easy way to spot server configuration issues and to develop your callback receiver code without sending lots of SMSes.

1.2 Accessing the AP

The BIP API is HTTP based, and the service is also accessible using the ready-made PHP library (the PHP library documentation is in Chapter 5).

The HTTP API URL is:

http://api.bipkampany.hu/
or
https://api.bipkampany.hu/

Invoking HTTP API actions

- use **HTTP GET** to invoke any action, provide parameters in the URL.
- all parameters must be in UTF-8, and the API will also respond in UTF-8.
- don't forget to **URL encode parameters** (eg. `rawurlencode()` function in PHP)

- Be aware of the typical issues during API calls:
  - always log API calls (even a simple text file is perfect) and the results too.

- What will your code do when there's a network outage – even local and/or remote: eg. firewall misconfiguration, internet provider failure, server failure? Always handle such cases!

API responses

- By default API will respond in URL encoded format (eg. `name1=value11&name2=value2`). You can choose JSON format too.
- The API will always return with at `result` value and a `code` value. The `result` will always be either success or error (`OK` or `ERR`), while the `code` value holds the specific error code.

What to do in an emergency?

Errors always happen: if your code starts sending hundreds of message to the same mobile number by mistake (which a real life example), you can turn off API temporarily to disable our own access. You can turn it back on once the issue is resolved.

1.3 About SMS messages: accents and special characters, handling charsets

As in every field of IT, handling special characters is a slippery question in the SMS technology too – make sure you take care of it.

- The API always expects UTF-8 data.
- After receiving the message data, BIP can deliver messages using:
  - basic GSM charset (GSM03.38 standard), which is the default
  - and Unicode charset (by using `type=unicode`). This mode allows less characters in a single SMS, but is a perfect (and actually the only) way to send even Chinese or Arabic characters, as the entire Unicode charset is available.

1.3.1 Sending messages using the GSM charset (which is the default)

You don't need to use any parameters or settings to send messages this way as it's the default mode. You can only use the following characters in this mode (the *space* character is also a part of the charset):

```
@£$¥èéùìòÇØøÅåΔ_ΦΓΛΩΠΨΣΘΞÆæßÉ
!"#¤%&\'()*+,-./0123456789:;<=>?¡
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÄÖÑÜ§¿
abcdefghijklmnopqrstuvwxyz
äöñüà
```

Even though you may find that your national characters are not included in the list above, the BIP service will convert typically used several national characters to their appropriate alternatives that are available in the GSM 03.38 charset:

```
áíéóúÁÍÚÓŐŰű
```

The characters above are converted to their counterparts – the same way how mobile devices do it – to these characters:

```
àìèòùÅIUOöÖüÜ
```

To help development you can get the entire available charset by using the `getcharset` action.

- This action was intended to be used during development and **occasional** checks.
- By checking the charset using the `getcharset` API call occasionally you can compare a charset you already use with the actually available charset – this way you can easily find out if there are new characters available in the API. It only happens when we occasionally include further character translations to support new countries.
- The allowed number of `getcharset` requests is **one request per day**.

The length of a message

The maximum length of an SMS is 459 characters. Depending on the length the message may be sent using max. 3 SMSes. Such messages are displayed as a complete single long message on the mobile devices. Message length limits are:

- 1 SMS: 160 characters
- 2 SMSes: 306 characters
- 3 SMSes: 459 characters

1.3.2 Sending messages using the Unicode charset

Only the Unicode mode provides proper support for special and national (even Chinese, Arabic, etc.) characters in SMSes.

Use the `type=unicode` parameter to use this mode while invoking the `sendsms` action to:

- send messages containing any Unicode character
- Unicode mode means that BIP will not alter characters in any way, all characters are accepted and delivered.

Length of Unicode messages

Unicode SMSes use multibyte encoding - each character is represented using two bytes. Therefore the number of characters available in a Unicode message is less than in plain GSM charset message:

- 1 SMS: 70 character
- 2 SMSes: 134 character
- 3 SMSes: 201 character

2 API actions: sendsms, getbalance, getcharset, cancelsms

You'll need to provide the `email` and `password` parameters for all API calls. All other parameters are detailed for each action with examples.

A quick overview of the actions:

sendsms – sending an SMS

http://api.bipkampany.hu/sendsms

Required parameters:	`email`, `password`, `message`, `number`
Optional parameters:	`senderid`, `format`, `referenceid`, `type`,`callback`,`timetosend`
Example:	http://api.bipkampany.hu/sendsms?message=Hello%20World&number=36309991111&senderid=TesztFelado&email=email@example.com&password=yourpassword

getbalance – balance checking

http://api.bipkampany.hu/getbalance

Required parameters:	`email`, `password`
Optional parameters:	`format`
Example:	http://api.bipkampany.hu/getbalance?email=email@example.com&password=yourpassword

getcharset – get SMS character set

http://api.bipkampany.hu/getcharset

Required parameters:	`email`, `password`
Optional parameters:	`format`
Example:	http://api.bipkampany.hu/getcharset?email=email@example.com&password=yourpassword

cancelsms – cancelling a timed SMS

http://api.bipkampany.hu/cancelsms

Required parameters:	`email`, `password`, `referenceid`
Optional parameters:	`format`
Example:	http://api.bipkampany.hu/cancelsms?email=email@example.com&password=yourpassword&referenceid=XXX

⚠ **Don't forget:** all parameters must be in UTF-8 and URL encoded format

2.1 Sending an SMS: the „sendsms" action

The URL needed to send an SMS message:

http://api.bipkampany.hu/sendsms

sendsms parameters		✪ - required parameter
email	✪	Your e-mail address
password	✪	Your password
message	✪	The message with a maximum length of 459 characters (sent as a concatenated message using 3 SMSes). By default GSM 03.38 characters are allowed (to query the available characters use the `getcharset` action, see Chapter 2.3). For further message characters use `type=unicode` – in this case the maximum SMS length is 201 characters. For further details see Chapter 1.3.
number	✪	The phone number in international format, eg. 36309991111 (where 36 is the country code for Hungary). The phone number must include digits only and no punctuation. Providing mobile phone numbers that belong to unsupported mobile networks will result in an error.
senderid		The message sender that will be shown on the mobile device (a phone number or an alphanumeric string). Get your own sender under the Settings menu on the web interface.
type		Message type. The only supported value currently is `unicode`. When this parameter is not specified, the message will be delivered using GSM 03.38 character set. When `type=unicode` is specified, the entire Unicode character set may be used in the message parameter. For further details see Chapter 1.3.
format		The format of the API response, possible value: `json` or `url`. Using `json` results in JavaScript Object Notation (JSON) formatted response, using `url` results in values encoded as URL parameters. Default value: `url`.
callback		Should you need status reports about your message, you need to specify the status codes you want to get back. Status codes must be separated by a comma. To get all available status report, use `ALL` instead of a status code list. For further details see the chapter about callbacks.
referenceid		Mandatory if `callback` parameter is used or if you want to use `cancelsms` later. This is a arbitrary string, that should be unique on your side: you may pass a database ID or any other string, eg. „523_5328_1" • You can use this ID later to cancel your timed SMS (`cancelsms` action). • If you've used the `callback` parameter, the API will return this value as part of the HTTP callback to the URL you've set as callback URL under Settings (without such an URL callback is not possible).
timetosend		Using this parameter you may specify the exact date and time when you want the message to be delivered. If the time has already passed, the message will be delivered instantly. Format: `YYYY-MM-DD hh:mm:ss`, eg. `2015-12-11 08:10:12`.

Example:
http://api.bipkampany.hu/sendsms?message=Hello%20World&number=36309991111
&senderid=BipKampany&email=email@example.com&password=yourpassword

API response example:

```
result=OK&code=0&message=Completed+successfully&price=25
```

or the same response in JSON format (`format=json`):

```
{
    "result":  "OK",
    "code":    0,
    "message": "Completed successfully",
    "price":   25
}
```

Values of the response are:

name	possible values
result	`OK` for successful API call, `ERR` for any error
code	`0` on success, and an error code on error (see error code table later).
message	The response as text, in UTF-8 codepage.
price	The price of the message. If the message length exceeds the length of a single SMS, the price will include the costs of all SMSes required to send the message.

2.2 Balance check: the „getbalance" action

The URL needeed to query your balance:

http://api.bipkampany.hu/getbalance

getbalance parameters		✪ - required parameter
email	✪	Your e-mail address
password	✪	Your password
format		The format of the API response, possible value: `json` or `url`. Using `json` results in JavaScript Object Notation (JSON) formatted response, using `url` results in values encoded as URL parameters. Default value: `url`.

Example:
http://api.bipkampany.hu/getbalance?email=email@example.com&password=yourpassword

API response example:

```
result=OK&code=0&message=Completed+successfully&currency=HUF&balance=450000&limit=300000
```

or the same response in JSON format (`format=json`):

```
{
    "result": "OK",
    "code": 0,
    "message": "Completed successfully",
    "currency": "HUF",
    "balance": 450000,
    "limit": 300000
}
```

The response is similar to a response received when submitting a message: you'll get `result`, `code`, and `message`. However, you'll also received `currency`, `balance` and `limit` too:

name	description	example		
result	Action result	`OK`	or	`ERR`
code	Result code	`0`	or	`304`
message	Result message	`Completed successfully`	or	`Your balance had expired, is missing or is unset.`
balance	Actual balance	`30.00`		
currency	Currency of balance	`EUR`		
limit	For prepaid clients it contains `0`. For postpaid clients this value holds the limit of maximum unpaid balance	`1000`		

2.3 Get available SMS characters: the „getcharset" action

To get the list of characters allowed to be used while sending SMSes:

http://api.bipkampany.hu/getcharset

getcharset parameters	✪ - required parameter	
email	✪	Your e-mail address
password	✪	Your password
format		The format of the API response, possible value: `json` or `url`. Using `json` results in JavaScript Object Notation (JSON) formatted response, using `url` results in values encoded as URL parameters. Default value: `url`.

Example:
http://api.bipkampany.hu/getcharset?email=email@example.com&password=yourpassword

Response example:

```
result=OK&code=0&message=Completed+successfully&charset=
%40%C2%A3%24%C2%A5%C3%A8%C3%A9%C3%B9%C3%AC%C3%B2%C3%87%C3%98%C3%B8%C3%85%C3%A5%CE
%94_%CE%A6%CE%93%CE%9B%CE%A9%CE%A0%CE%A8%CE%A3%CE%98%CE%9E%C3%86%C3%A6%C3%9F
%C3%89+%21%22%23%C2%A4%25%26%27%28%29%2A%2B%2C-.%2F0123456789%3A%3B%3C%3D%3E%3F
%C2%A1ABCDEFGHIJKLMNOPQRSTUVWXYZ%C3%84%C3%96%C3%91%C3%9C
%C2%A7%C2%BFabcdefghijklmnopqrstuvwxyz%C3%A4%C3%B6%C3%B1%C3%BC%C3%A0%C3%A1%C3%AD
%C3%A9%C3%B3%C3%BA%C3%81%C3%8D%C3%9A%C3%93%C5%91%C5%90%C5%B1%C5%B0
```

or the same response in JSON format (`format=json`):

```
{
  "result":  "OK",
  "code":    0,
  "message": "Completed successfully",
  "charset":"@\u00a3$\u00a5\u00e8\u00e9\u00f9\u00ec\u00f2\u00c7\u00d8
     \u00f8\u00c5\u00e5\u0394_\u03a6\u0393\u039b\u03a9\u03a0\u03a8
     \u03a3\u0398\u039e\u00c6\u00df\u00c9 !\"#
     \u00a4%&'()*+,-.\/0123456789:;<=>?
     \u00a1ABCDEFGHIJKLMNOPQRSTUVWXYZ\u00c4\u00d6\u00d1\u00dc\u00a7
     \u00bfabcdefghijklmnopqrstuvwxyz\u00e4\u00f6\u00f1\u00fc\u00e0
     \u00e1\u00ed\u00e9\u00f3\u00fa\u00c1\u00cd\u00da\u00d3\u0151
     \u0150\u0171\u0170"
}
```

The response is the usual `result`, `code`, `message` triplet accompanied by the `charset` value itself.

The `charset` string contains all the characters accepted by the API for sending SMS messages. This string has an UTF-8 coding as all the API output in general.

This action only responds with an error in `result` if case of authentication errors, missing parameters or syntax errors.

2.4 Cancelling an SMS: the „cancelsms" action

If an SMS has been submitted using `timetosend` and `referenceid` parameters, you can use this action to cancel the message (unless it has been already delivered).

To cancel an SMS you need to use this URL:

http://api.bipkampany.hu/cancelsms

cancelsms parameters		✪ - required parameter
email	✪	Your e-mail address
password	✪	Your password
referenceid	✪	The reference ID used when submitting the SMS. To use multiple reference IDs separate them with a comma. When using multiple reference IDs, make sure the URL gets no longer than 65000 bytes.
format		The format of the API response, possible value: `json` or `url`. Using `json` results in JavaScript Object Notation (JSON) formatted response, using `url` results in values encoded as URL parameters. Default value: `url`.

Example:
http://api.bipkampany.hu/cancelsms?
referenceid=1234&email=email@example.com&password=yourpassword

cancelling multiple SMSes:
http://api.bipkampany.hu/cancelsms?referenceid=1234,1235,1236,1237
&email=email@example.com&password=yourpassword

Response example:

```
result=OK&code=0&message=Completed+successfully
```

or the same response in JSON format (`format=json`):

```
{
    "result": "OK",
    "code": 0,
    "message": "Completed successfully",
}
```

The response is the usual `result`, `code` and `message` values. You can get an error if there's no SMS with the given reference ID or the message has already been delivered.

The possible `result`, `code`, and `message` combinations are:

result	code	message
OK	0	Completed successfully
ERR	108	Message cannot be found using the provided referenceid.

3 API response codes

We suggest logging the error responses of the API in your application (eg. error messages and codes can be added into database fields of the table that holds outgoing messages), as you can easily filter your messages later if you need to resend them for any reason.

It's also a best practice to send an e-mail about all the errors to yourself at least (or use proper logging and log monitoring software).

0 – successful completion

result	code	message
OK	0	Completed successfully

10x – syntax/format errors

result	code	message
ERR	100	Missing parameter: `parametername`
ERR	101	Numeric value required in parameter: `parametername`
ERR	102	Phone number must be in international format, or given country/number is not supported.
ERR	103	The message is longer that 459 characters
ERR	104	The sender ID (senderid) provided is not allowed
ERR	105	The message contains invalid characters
ERR	106	The callback parameter is invalid
ERR	107	The timetosend timestamp is invalid
ERR	108	Message cannot be found using the provided referenceid.

30x – access or balance related errors

result	code	message
ERR	300	E-mail or password is invalid, API access not yet enabled (see Settings), or access is disabled.
ERR	301	The IP address `ipaddress` is not allowed by your settings
ERR	302	Your balance (`current balance`) is less than required to send this message (`price of message`)
ERR	303	Your payment limit has been hit, please complete a payment to send more messages
ERR	304	Your balance had expired, is missing or is unset.

50x – service errors

result	code	message
ERR	500	The service is temporarily unavailable. Our staff has been notified and are working on the issue

4 Callback service

The API provides feedback for the submitted SMSes – it's called DLR (delivery report). As all DLRs asynchronous, there needs to be a receiving HTTP endpoint on your side.

4.1 What you'll need to receive callbacks

- You need to have a HTTP URL handling callbacks
- We provide a testing tool under **Settings** on the website: this way you can easily create your callback receiver endpoint without sending too much SMSes.
- Callbacks are only created for SMSes submitted through the API. Messages submitted through the website can only be tracked on the website itself.
- You'll need to provide two parameters while submitting your message with `sendsms`:
  - with `callback` you can adjust the type of statuses you'd like to receive (this way you can ignore non-final statuses)
  - the `referenceid` is simply sent back to you as received – you can use this value to identify the message in your application

4.2 Calling the callback URL

The API forwards incoming statuses to the given URL with the parameters described below.

If the URL is not accessible, the callback will be retried later with the following pattern: after 5 minutes we retry callbacks every minute until 55 more minutes, then once in 1, 2, 3, 4, 24, 48 and 72 hours. Undeliverable callbacks older than three days are not retried anymore.

4.3 Example for submitting a message and receiving callback

(to keep the URLs easily readable the URLs are not URL encoded):

1) Let's assume you have this callback URL on your side:
http://example.com/callback.aspx

2) We use `referenceid` and `callback` when submitting the SMS:

```
http://api.bipkampany.hu/sendsms
 ?email=email@example.com          &password=yourpassword
 &number=36309991111               &message=Hello World          &referenceid=123
 &callback=SENT,ACCEPTED,WAITING,DELIVERED,REJECTED,UNDELIVERABLE,MISC,STOPPED,EXPIRED,STATUSLOST
```

You may use the `ALL` shortcut to receive all message status:

```
&callback=ALL
```

3) The BIP API constantly processes and delivers message statuses. The API returns `referenceid=123` parameter that was used when submitting the SMS:

```
http://example.com/callback.aspx
 ?referenceid=123                  &number=36309991111          &status=SENT
 &message=Távoli SMSC-nek továbbítva  &price=30.00               &timestamp=2015-12-31 22:51:21

http://example.com/callback.aspx
 ?referenceid=123                  &number=36309991111          &status=ACCEPTED
 &message=Távoli SMSC által elfogadva  &price=30.00              &timestamp=2015-12-31 23:01:31

http://example.com/callback.aspx
 ?referenceid=123                  &number=36309991111          &status=DELIVERED
 &message=Kézbesítve               &price=30.00                 &timestamp=2015-12-31 23:59:59
```

The BIP API provides the following callback parameters:

parameter name	description	example
referenceid	The value of `referenceid` parameter that was used when submitting the SMS	5447
number	Phone number	36309991111
status	Status code (see table below)	DELIVERED
message	Status message (UTF-8 coding)	Kézbesítve
price	Message price (always the same amount, not to be summed up)	30.00
timestamp	The timestamp of receiving the status report by the BIP API	2016-02-01 21:59

Possible values of `status` :

status	message	description
ⓘ SENT	Submitted to remote SMSC	Message delivery has been initiated successfully
ⓘ ACCEPTED	Accepted by remote SMSC	The SMS centers accepted the message. If the subscription is available for the mobile number given, the mobile device is turned on and is available in the mobile network, the message will be delivered. If the device is turned off, providers will retry delivery several times using their own retry patterns.
ⓘ WAITING	Waiting for mobile device	The mobile is temporarily not accessible
✔ DELIVERED	Delivered to mobile device	The mobile device reported successful delivery.
✖ REJECTED	Rejected by SMSC	Mobile provider issues or legal issues (eg. spam)
✖ UNDELIVERABLE	Undeliverable to device	The device wasn't accessible during delivery window (pl. prepaid subscription expired, invalid mobile number, device turned off, out of cellular network range, network data transfer issue, etc.)
✖ MISC	Miscellaneous error	General error from the provider side.
✖ STOPPED	Sending has been stopped	Mobile provider has stopped sending the message.
✖ EXPIRED	Message has been expired	Mobile provider couldn't deliver the message in the delivery time window.
✖ STATUSLOST	The status was lost	Mobile provider side issue.

5 PHP class for BIP API

We've created a simple PHP class for the BIP API which helps sending SMSes easier.

5.1 Methods of the PHP class

If you want to skip a parameter during method calls – eg. you don't need `$timetosend` in `sendSMS`, but still would like to set `$type` –, use `null` instead of the parameter to skip.

constructor

```
void BipKampanyAPI::__construct( string $email, string $password )
```

Constructor.

Parameters:

`$email` – your e-mail
`$password` – your password

sendSMS()

```
array BipKampanyAPI::sendSMS(string $number, string $message, string $senderid,
                             string $timetosend = null, string $type = null,
                             string $callback = null,
                             string $referenceid = null )
```

SMS submission.

Parameters:

`$number` – phone number in international format (eg. `36309991111`)
`$message` – the message in UTF-8 coding (check available chars too!)
`$senderid` – a sender ID (eg. `BipKampany`)
`$timetosend` – optional: timing of the SMS, YYYY-MM-DD hh:mm:ss (eg. `2015-11-25 12:08:59`)
`$type` - optional: message type. Currently only: `unicode`
`$callback` - optional: callback types separated by a comma or `ALL` to receive all callback types.
`$referenceid` - optional: the identifier that's provided to you during callbacks. Only to be used together with `$callback`.

Returns: the API reponse as an array. Raises an exception upon errors.

getBalance()

```
array BipKampanyAPI::getBalance()
```

Querying the current balance.

Returns: the API response as an array, including `balance` and `currency` holding the balance itself. Raises an exception upon errors.

cancelSMS()

```
array BipKampanyAPI::cancelSMS( string $referenceids )
```

Cancelling a timed SMS submitted earlier.

Parameters:

`$referenceids` – the reference ID used when submitting the SMS. To use multiple reference IDs separate them with a comma. When using multiple reference IDs, make sure the URL gets no longer than 65000 bytes.

Returns: the API response as an array. Raises an exception upon errors.

getCharset()

`array BipKampanyAPI::getCharset()`
Get available SMS characters.

Returns: the API response as an array, including `charset` string with the supported characters (in UTF-8 coding). Raises an exception upon errors.

getLastResult()

`mixed BipKampanyAPI::getLastResult()`
Get the latest API response.

Returns: null, if there was no API response yet (before any API calls). Otherwise it returns the response as an array.

5.2 Attributes of the PHP class

logFilename

`string BipKampanyAPI::logFilename`
Log fájl name full path and name. When set, BIP API class will log all the BIP API URLs to be called before the call takes place and also the response itself as an array dumped with `var_export()`.

callingMethod

`string BipKampanyAPI::callingMethod`
The method to use for calling the API

The default value of this attribute is `file_get_contents`, which makes the class use `file_get_contents()` function appropriately. You need to have `allow_url_fopen` set to `On` in PHP settings – the class also checks this setting and raises an exception if it's not set to `On`).

The other option is `curl` – this way the CURL extension will be used. You need to have CURL extension available, which is checked by the class and raises an exception if it's missing.

5.3 PHP examples

Available examples on our website:

- `example1_http.php` – the most basic PHP usage: send SMS using `file_get_contents()` even without the BIP API class. Also shows basic error handling.
- BIP API class examples:
  - `example_api_basic.php` – basics: send SMS, handling exceptions
  - `example_api_complete.php` – using logging, advanced exception handling, timing an SMS, sending Unicode SMS-es
  - `example_api_curl.php` – using CURL extension
  - `example_api_balance.php` – getting balance
  - `example_api_charset.php` – getting accepted characters for sending an SMS