

Verzió: 3.6

Dátum: 2023-09-25

Figyelem! Az API-t első használat előtt engedélyezze bejelentkezés után a Beállítások menüpontban!

Bármilyen felmerülő probléma, kérdés, észrevétel esetén szívesen segítünk az info@bipkampany.hu címen!

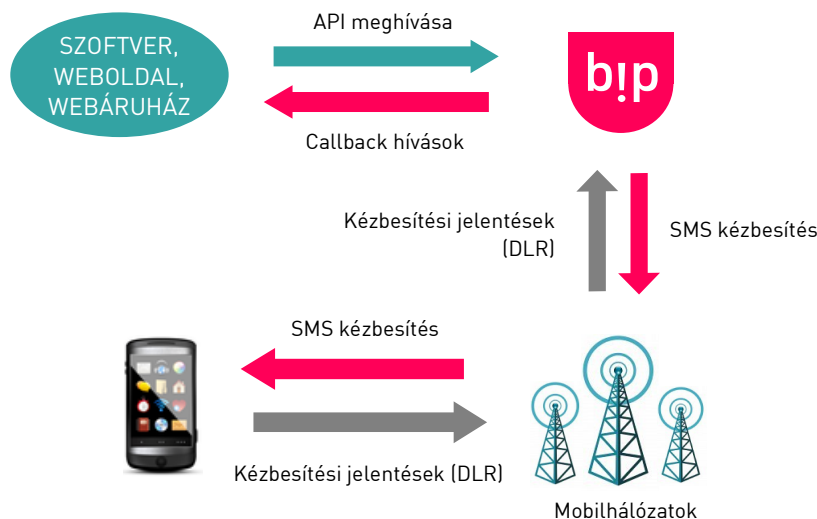
Tartalomjegyzék

1 Az API-ról.....	2
1.1 A BIP fiók beállítása API használathoz.....	2
1.2 Az API elérése.....	2
Az API HTTP műveleteinek meghívása.....	2
Az API válasza HTTP GET hívások esetén.....	3
Az e-mail-SMS átjáró meghívása (SMTP API) és válaszok.....	3
Gyors hibaelhárítás.....	3
1.3 Az SMS üzenetekről: ékezetes és speciális karakterek, kódlap kezelése.....	3
1.3.1 Üzenetküldés alap GSM karakterkészlettel (alapértelmezés).....	4
1.3.2 Üzenetküldés Unicode karakterkészlettel.....	4
2 API műveletek.....	5
2.1 SMS műveletek.....	6
2.1.1 SMS feladása: a „sendsms” művelet.....	6
2.1.2 SMS-ben küldhető karakterek lekérdezése: a „getcharset” művelet.....	8
2.1.3 Időzített üzenet visszavonása: a „cancelsms” művelet.....	9
2.2 Két faktoros autentikáció SMS-ben (2FA).....	10
2.2.1 Token küldése SMS-ben: a „tfasendtoken” művelet.....	11
2.2.2 Token visszaellenőrzése: a „tfavalidatetoken” művelet.....	12
2.2.3 Token érvénytelenítése: a „tfainvalidatetoken” művelet.....	13
2.3 Fiókkal kapcsolatos műveletek.....	14
2.3.1 Profiladatok lekérdezése: a „getprofile” művelet.....	14
2.3.2 Egyenleg lekérdezése: a „getbalance” művelet.....	15
3 API hibakódok.....	16
4 Callback szolgáltatás.....	18
4.1 A visszajelzések fogadásának feltételei.....	18
4.2 A callback URL meghívása.....	18
4.3 Példa egy üzenet feladására és a meghívott callback URL-ekre.....	18
5 PHP osztály a BIP API-hoz.....	20
5.1 A PHP osztály metódusai.....	20
5.2 A PHP osztály attribútumai.....	21
5.3 Példák PHP nyelven.....	21
6 E-mail-SMS átjáró (SMTP API).....	22
Az e-mail formája.....	22
Hibakezelés.....	22
6.1 Üzenetküldés a részletes e-mail API-val.....	23
6.2 Hibakeresés (sendsmsdebug).....	25
6.3 Egyszerűsített üzenetküldés az e-mail API-val.....	26
7 REST SMS API.....	27
7.1 Támogatott HTTP fejlécek.....	28
7.2 Támogatott REST API műveletek.....	29
7.3 REST API hibakódok.....	30
8 SMS fogadás.....	32
8.1 SMS fogadás API (bejövő üzenetek feldolgozásához).....	33

1 Az API-ról

Az API a BIP SMS szolgáltatása és partnereink saját szolgáltatásainak, szoftvereinek, weboldalainak, webáruházainak összekapcsolására alkalmas szabványos felület.

Az API működését legkönnyebben az alábbi ábrával tekinthetjük át:



Az ábrán a nyilak színei az adott folyamat kezdeményezőjét jelzik.

1.1 A BIP fiók beállítása API használathoz

Az API-t a legelső használat előtt engedélyeznünk kell fiókunkban:

- a weboldalon belépve a *Beállítások* alatt az *API hozzáférés engedélyezve* beállítást állítsuk *igen*-re. Az itt látható egyedi API kulcsot kell használnunk az összes művelet **key** paraméterében.
- **IP beállítás:** az IP cím beállításával korlátozhatjuk a hozzáférhetőséget. Így csökkenthető a visszaélések kockázata, ezért használatát javasoljuk.
- **Callback URL:** a BIP SMS rendszere egyszerű HTTP hívásokon keresztül vissza tudja adni a feladott üzenetekről későbbiekben keletkező státuszjelentéseket. A fogadásukra szolgáló URL-t a *Callback URL* mezőben állíthatjuk be. A *Callback teszt kérése* gomb próbahívást végez a megadott címre, és megjeleníti a teljes meghívott URL-t és a kapott választ is (HTTP fejléceket és tartalmat).

1.2 Az API elérése

A BIP rendszerét megszólíthatjuk HTTP GET hívásokkal, de REST API-n keresztül (ld. 7. fejezet), a PHP nyelven elérhető objektumkönyvtárral (ld. 5. fejezet), vagy akár e-mail segítségével is (ld. 6. fejezet).

A HTTP API közvetlen címe:

<https://api.bipkampany.hu/>

Az API HTTP műveleteinek meghívása

- az API-t **HTTP GET** metódussal érhetjük el (a paramétereket URL-ben adjuk át), vagy REST API felületen (ld. 7. fejezet).
- az összes paramétert **UTF-8 kódlappal** kell elküldeni, és az API is UTF-8 válaszokat ad.
- HTTP GET esetén az értékeket ne felejtsük el **URL formába kódolni** (pl. `rawurlencode()` függvénnyel PHP használata esetén)

- Az API meghívásakor érdemes körültekintőnek lenni, és figyelni a következőkre:
 - naplózzuk lokálisan az API hívásokat (akár egy egyszerű szövegállományban) és a kapott válaszokat is
 - mi történik, ha helyi vagy távoli hálózati hiba lép fel: pl. megváltozott tűzfalszabály, szolgáltatói kimaradás, serverleállás? Kezeljük le az ilyen eseteket!

Az API válasza HTTP GET hívások esetén

- Az alapértelmezés esetén a válasz URL formában kódolva érkezik (pl. `nev1=ertek1&nev2=ertek2`), de választható JSON formátum is. A REST API válaszai a REST elvek és a REST fejezetben leírtak szerint érkeznek.
- Az API minden műveletre válaszul küld egy `result` (OK vagy ERR) és `code` értéket. A `result` értéke alapján eldönthető, sikeres volt-e a művelet, a kód értéke alapján pedig pontosan beazonosíthatjuk egy esetleges hiba okát.
- Hiba esetén az API e-mailben is küld értesítést a regisztrációkor használt email címre, vagy arra a címre, amit a webes felület Beállítások pontjában beállítunk az API beállítások között, mint API hiba e-mail cím. Az e-mail értesítések küldése nem letiltható, a cél a minden körülmények között helyesen működő API csatlakozások elősegítése.

Ez a megoldás segít felderíteni az eredetileg jól beállított, de valamiért későbbiekben mégis problémás helyzeteket (pl. értesítések kifogyott egyenleg esetén).

Az e-mail-SMS átjáró meghívása (SMTP API) és válaszok

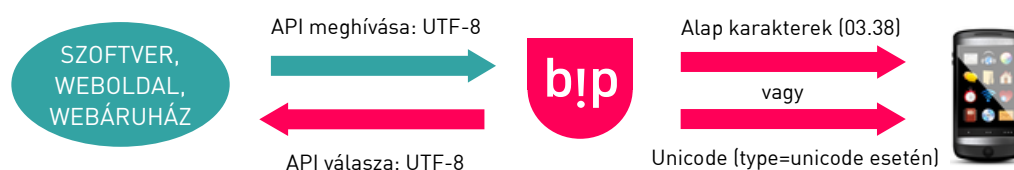
- E-mail alapon is elérhető az API egy részletes és egy egyszerűsített formában is. A részletek a 6. fejezetben olvashatók.

Gyors hibaelhárítás

Ha az API-t használó kódunkba hiba csúszott (pl. hirtelen több ezer üzenetet kezd el küldeni ugyanarra a számra – volt már rá példa), az API kikapcsolásával átmenetileg letiltható saját API hozzáférésünk, amíg a hibás folyamatok véget nem érnek.

1.3 Az SMS üzenetekről: ékezetes és speciális karakterek, kódlap kezelése

Ahogy minden alkalmazásban, az SMS-ek esetében is kényes kérdés a kódlapok és ékezetes karakterek kezelése, fejlesztés közben kiemelten figyeljünk rá.



- Az API UTF-8 kódolásban várja és adja vissza az adatokat.
- A BIP SMS rendszere miután megkapta az adatokat, kétféle módon tudja eljuttatni az üzeneteket a mobilkészülékekhez:
 - alap GSM karakterkészlettel (GSM03.38 szabvány), ami az alapértelmezés
 - és Unicode karakterkészlettel (`type=unicode` paraméter használata esetén). Bár így kevesebb karaktert küldhetünk el egy üzenetben, de az akár kínai vagy arab írásjelekből is állhat, hiszen ekkor a teljes Unicode karakterkészlet a rendelkezésünkre áll.

1.3.1 Üzenetküldés alap GSM karakterkészlettel (alapértelmezés)

Ehhez a módszerhez nincs szükség semmilyen beállításra, az üzenetfeladás alapesetben így zajlik. Ennél az üzenettípusnál kizárólag a következő karaktereket használhatjuk az üzenet szövegében (természetesen a karakterkészlet része a szóköz is):

```
@!$%&'()*+,-./0123456789:;<=>?;
!"#%&'\ '()* * , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? ;
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÄÖŃÜŞ;
abcdefghijklmnopqrstuvwxyz
äöñüà
```

Bár a fenti készletnek nem része a magyar ékezetes karakterek nagy része, mi a felhasználóink kényelme érdekében az alábbi, nem GSM 03.38 részét képező karaktereket is elfogadjuk, amit a mobilkészülékeken olvashatóság kedvéért átalakítunk:

```
áíéóúÁÍÚÓóÚú
```

A fenti karaktereket – a mobilkészülékek SMS küldő rendszerével azonos módon – az alábbiakra alakítja át a BIP:

```
àìèòùÀÌUOöÜü
```

A fejlesztést gyorsítandó a teljes támogatott karakterkészlet lekérdezhető az API `getcharset` műveletével (ennek eredményét lehetőleg helyben cache-eljük le legalább egy napos lejárattal).

Üzenethossz

Az SMS feladásakor használható maximális üzenethossz 459 karakter. Ez 3 darab SMS felhasználását jelenti, amelyek összefűzve, egyetlen egybefüggő üzenetként érkeznek meg a mobiltelefonra.

Üzenethossz határok:

- 1 db SMS: 160 karakter
- 2 db SMS: 306 karakter
- 3 db SMS: 459 karakter

1.3.2 Üzenetküldés Unicode karakterkészlettel

Teljes körű magyar ékezetes karaktértámogatás vagy speciális ékezetes és egyéb karakterek (kínai, arab, egyéb jelek) küldése csak az Unicode típusú üzenetküldéssel lehetséges.

Ezt a `type=unicode` paraméter megadásával érhetjük el a `sendsms` műveletnél. Ennek következményei:

- A teljes magyar ékezetes karakterkészleten túl akár kínai vagy más speciális karaktereket tartalmazó üzeneteket is küldhetünk.
- Ilyenkor a BIP rendszere nem végzi el az GSM 03.38-as üzenettípusnál leírt karakterátalakítást, hiszen a minden megkapott karaktert elfogadunk és kézbesítünk is.

Üzenethossz

Mivel az Unicode kódolású üzenetek multibyte kódolással dolgoznak (ez a módszer két byte-on ábrázol egy karaktert), emiatt kb. felére csökken az SMS-enként küldhető karakterek száma:

- 1 db SMS: 70 karakter
- 2 db SMS: 134 karakter
- 3 db SMS: 201 karakter

2 API műveletek

Minden API műveletnél kötelező az azonosításhoz szükséges **key** paraméter (az egyedi API kulcs a webes belépést követően a Beállítások oldalon található meg). Az egyéb paramétereket műveletenként mutatjuk be, példákkal illusztrálva.

A leggyakrabban használt műveletek gyors áttekintése:

sendsms – SMS feladása <https://api.bipkampany.hu/sendsms>

Kötelező paraméterek: **key, message, number**

Opcionális paraméterek: **senderid, format, referenceid, type, callback, timetosend**

Példa: `https://api.bipkampany.hu/sendsms?message=Hello%20World&number=36309991111&senderid=TesztFelado&key=SAJAT_API_KEY`

Részletek: [SMS feladása →](#)

getcharset – engedélyezett karakterek lekérdezése <https://api.bipkampany.hu/getcharset>

Kötelező paraméterek: **key**

Opcionális paraméterek: **format**

Példa: `https://api.bipkampany.hu/getcharset?key=SAJAT_API_KEY`

Részletek: [Karakterkészlet lekérdezése →](#)

cancelsms – SMS visszavonása <https://api.bipkampany.hu/cancelSMS>

Kötelező paraméterek: **key, referenceid**

Opcionális paraméterek: **format**

Példa: `https://api.bipkampany.hu/cancelSMS?key=SAJAT_API_KEY&referenceid=XXX`

Részletek: [Időzített üzenet visszavonása →](#)

getbalance – egyenleg lekérdezése <https://api.bipkampany.hu/getbalance>

Kötelező paraméterek: **key**

Opcionális paraméterek: **format**

Példa: `https://api.bipkampany.hu/getbalance?key=SAJAT_API_KEY`

Részletek: [Egyenleg lekérdezése →](#)



Emlékeztető: minden paraméterértéket UTF-8 kódlappal és URL-kódolt formában várunk.

2.1 SMS műveletek

2.1.1 SMS feladása: a „sendsms” művelet

Az SMS küldéshez szükséges URL:

<https://api.bipkampany.hu/sendsms>

sendsms paraméterek

🚩 - kötelező paraméter

key 🚩	Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
message 🚩	A feladni kívánt üzenet, maximum 459 karakter hosszúságban (3 db SMS felhasználásával összefűzött üzenet). Alapértelmezésben a GSM 03.38 szerinti karakterek engedélyezettek (a használható karakterek lekérdezhetők a <code>getcharset</code> művelettel, ld. a 2.3 pontban). Egyéb karakterekhez <code>type=unicode</code> beállítás szükséges, ekkor a maximális hossz 201 karakter. További részletekért lásd még az 1.3 pontot a dokumentációban.
number 🚩	A címzett telefonszáma nemzetközi formátumban, pl. 36309991111. A telefonszám kizárólag számokat tartalmazhat. A rendszer hibát jelez, ha nem támogatott célhálózatot (mobil körzetszámot) észlel, vagy ha a címzett leiratkozott az SMS fogadásról az adott fiókban.
senderid	A mobilkészüléken feladóként megjelenő feladóazonosító (telefonszám vagy szöveg). Ha nincs megadva, alapértelmezésben a <code>BipKampany</code> feladóazonosítót használjuk. A webes felületen igényelhető saját feladóazonosító a Beállítások menüpontban.
type	Az üzenet típusa. Jelenleg egy értéket támogat: <code>unicode</code> . Ha nem adjuk meg, az üzenet GSM 03.38 karakterkészlettel kerül kézbesítésre, <code>type=unicode</code> esetén azonban a teljes Unicode karakterkészletet használhatjuk az üzenet szövegében. További részletekért lásd az 1.3 pontot a dokumentációban.
format	A válasz formátuma, lehetséges értékek: <code>json</code> vagy <code>url</code> . Ha <code>json</code> , akkor JSON kódolt értékeket adunk vissza, ha <code>url</code> , akkor URL paraméterekben kódolt választ küldünk. Alapértelmezés: <code>url</code> .
callback	Ha BIP által indított HTTP hívás segítségével információt kérünk vissza az üzenetek állapotáról, itt kell felsorolnunk azokat az állapotkódokat, amelyekről a callbacket kérjük. Az állapotkódokat vesszővel elválasztva soroljuk fel. Ha mindegyik állapotot kérjük, használjuk az <code>ALL</code> értéket a felsorolás helyett. További részletek a callbacket ismertető fejezetben.
referenceid	Kötelező <code>callback</code> vagy <code>cancelSMS</code> művelet esetén. Egy tetszőleges felépítésű, tartalmú egyedi üzenetazonosító: lehet pl. egy adatbázis rekord azonosítója, vagy valamilyen összetett sztring, pl. „523_5328_1” <ul style="list-style-type: none">Ezzel az azonosítóval azonosítható az üzenet, ha időzítve adtuk fel, de szeretnénk visszavonni (<code>cancelSMS</code> művelet).Ha a <code>callback</code> paramétert beállítottuk, az API visszaadja ezt az azonosítót a HTTP hívás részeként a Beállításoknál megadott callback URL-re. Ha ilyen URL nincs beállítva, a callback nem lehetséges.
timetosend	Időzítés időpontja, amikor az üzenet kézbesítésre kerül. Formátum: <code>ÉÉÉÉ-HH-NN óó:pp:mm</code> , pl. <code>2015-12-11 08:10:12</code> .

Például:

https://api.bipkampany.hu/sendsms?message=Hello%20World&number=36309991111&senderid=BipKampany&key=SAJAT_API_KEY

Példa az SMS átjáró válaszára:

```
result=OK&code=0&message=A k%C3%A9r%C3%A9st sikeresen teljes%C3%ADtett%C3%Bck&price=25&split=1
```

vagy JSON formátum (`format=json`) esetén:

```
{
  "result": "OK",
  "code": 0,
  "message": "A k\u00e9r\u00e9st sikeresen teljes\u00edtett\u00fck",
  "price": 25,
  "split": 1
}
```

A válaszban szereplő értékek:

név	lehetséges értékek
result	Értéke OK ha minden rendben, vagy ERR hiba esetén.
code	Értéke siker esetén 0 , hiba esetén egy hibakód (lásd a hibakód táblázatban).
message	A válasz üzenet szövegesen, UTF-8 kódlappal.
price	Az üzenet ára (összefűzött, 160 karakternél hosszabb üzenetnél az felhasznált SMS-ek összesített ára).
split	Ennyi SMS összefűzésével volt kézbesíthető az üzenet (értéke 1, 2 vagy 3 lehet).

2.1.2 SMS-ben küldhető karakterek lekérdezése: a „getcharset” művelet

A karakterkészlet lekérdezéséhez szükséges URL:

<https://api.bipkampany.hu/getcharset>

getcharset paraméterek

🔴 - kötelező paraméter

key	Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
format	A válasz formátuma, lehetséges értékek: <code>json</code> vagy <code>url</code> . Ha <code>json</code> , akkor JSON kódolt értékeket adunk vissza, ha <code>url</code> , akkor URL paraméterekben kódolt választ küldünk. Alapértelmezés: <code>url</code> .

Például:

https://api.bipkampany.hu/getcharset?key=SAJAT_API_KEY

Példa az SMS átjáró válaszára:

```
result=OK&code=0&message=A k%C3%A9r%C3%A9st sikeresen teljes%C3%ADtott%C3%Bck&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%BFabcdefghijkmnopqrstuvwxyz%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
```

vagy JSON formátum (`format=json`) esetén:

```
{
  "result": "OK",
  "code": 0,
  "message": "A k\u00e9r\u00e9st sikeresen teljes\u00edtett\u00fck",
  "charset": "@\u00a3$\u00a5\u00e8\u00e9\u00f9\u00ec\u00f2\u00c7\u00d8\u00f8\u00c5\u00e5\u0394_\u03a6\u0393\u039b\u03a9\u03a0\u03a8\u03a3\u0398\u039e\u00c6\u00e6\u00df\u00c9 !\"#\u00a4&'()* ,-.\/0123456789:;<=>?\u00a1ABCDEFGHIJKLMNOPQRSTUVWXYZ\u00c4\u00d6\u00d1\u00dc\u00a7\u00bfabcdefghijkmnopqrstuvwxyz\u00e4\u00f6\u00f1\u00fc\u00e0\u00e1\u00ed\u00e9\u00f3\u00fa\u00c1\u00cd\u00da\u00d3\u0151\u0150\u0171\u0170"
```

Válaszként az üzenetfeladáshoz hasonlóan a szokásos `result`, `code`, és `message` értéket kapunk vissza kiegészítve a `charset` értékekkel.

A `charset` tartalma az elfogadott karakterkészlet összes karaktere egyetlen sztringben, UTF-8 kódolással.

A `result` ennél a műveletnél csak azonosítási hiba vagy hiányzó paraméter esetén veszi fel az `ERR` értéket.

2.1.3 Időzített üzenet visszavonása: a „cancelsms” művelet

Ha egy üzenetet a `timetosend` és a `referenceid` paraméterek megadásával időzítve adtuk fel, az üzenetet szükség esetén visszavonhatjuk (ha az még nem került kiküldésre).

Az üzenet visszavonásához szükséges URL:

<https://api.bipkampany.hu/cancelsms>

cancelsms paraméterek

★ - kötelező paraméter

<code>key</code> ★	Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
<code>referenceid</code> ★	Az az üzenetazonosító, amelyet az üzenet feladásakor használtunk. Több üzenetazonosítót is megadhatunk vesszővel elválasztva. Ügyeljünk sok azonosító megadásánál arra, hogy az így összeálló URL hossza ne legyen több 65000 byte-nál.
<code>format</code>	A válasz formátuma, lehetséges értékek: <code>json</code> vagy <code>url</code> . Ha <code>json</code> , akkor JSON kódolt értékeket adunk vissza, ha <code>url</code> , akkor URL paraméterekben kódolt választ küldünk. Alapértelmezés: <code>url</code> .

Például:

https://api.bipkampany.hu/cancelsms?referenceid=1234&key=SAJAT_API_KEY

vagy több üzenet egyidejű visszavonásához:

https://api.bipkampany.hu/cancelsms?referenceid=1234,1235,1236,1237 &key=SAJAT_API_KEY

Példa az SMS átjáró válaszára:

```
result=OK&code=0&message=A k%C3%A9r%C3%A9st sikeresen teljes%C3%ADtott%C3%BCK
```

vagy JSON formátum (`format=json`) esetén:

```
{  
  "result": "OK",  
  "code": 0,  
  "message": "A k\u00e9r\u00e9st sikeresen teljes\u00edtett\u00fck",  
}
```

Válaszként az üzenetfeladáshoz hasonlóan `result`, `code`, és `message` értéket kapunk vissza. Hibaüzenet akkor fordulhat elő, ha a megadott azonosító nem létezik, vagy már elküldésre került az üzenet.

A két lehetséges `result`, `code`, és `message` válaszkombináció:

result	code	message
OK	0	A kérést sikeresen teljesítettük
ERR	108	A megadott referenciaid-vel függőben lévő üzenet nem található

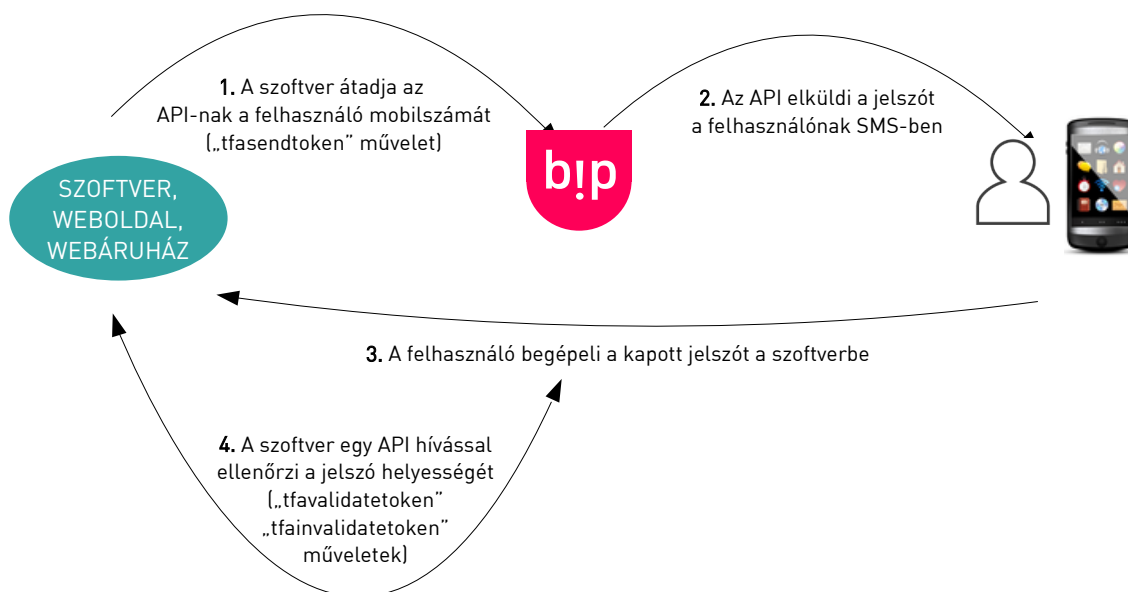
2.2 Két faktoros autentikáció SMS-ben (2FA)

Két- (vagy multi-) faktoros autentikációval egy tetszőleges rendszer belépési folyamatát tehetjük sokkal biztonságosabbá.

A felhasználó – miután már azonosította magát a szoftverben jellemzően felhasználónévvel és jelszóval – a BIP API-n keresztül SMS-ben kap egy egyszer használatos jelszót („token”), amit a rövid lejáratú időn belül a belépés során kell felhasználnia.

A token a BIP API generálja, tartja nyilván és azon keresztül ellenőrizhető, így alacsonyabb a feltörhetőség, mintha a token a beléptetést végző rendszer generálta volna.

A folyamat vázlatosan:



Az ilyen belépési folyamatoknál szokásos best practice-ek:

- a belépést követően érdemes a `tfainvalidatetoken` hívással érvényteleníteni az API oldalán a token, ezzel csökkentve az esetleges visszaélések lehetőségét a token lejáratú idejéig
- a mobilszolgáltatónál időről időre tapasztalható kiesések miatt érdemes lehet gondolni vész megoldásra is (pl. ha az SMS bármilyen okból nem érkezik meg a felhasználónak, vagy a szoftver és a BIP API közötti hálózati kapcsolat esik ki bármilyen okból):
 - gond esetén a szoftvert működtető cég ügyfélszolgálatára beengedheti egyedileg a felhasználót a rendszerbe
 - átmenetileg kikapcsolhatóvá lehet tenni az SMS alapú autentikációt
 - vagy olyan hosszú lejáratú idejű jelszavakat lehet kiosztani előre a felhasználóknak, melyeket helyben tárol el a szoftver (vagyis ezt nem a BIP API tartja nyilván). Így probléma esetén jóval alacsonyabb lesz az ügyfélszolgálati megkeresések száma, hiszen a felhasználók maguk is meg tudják oldani a helyzetet.

2.2.1 Token küldése SMS-ben: a „tfasendtoken” művelet

Ezzel a művelettel egy két- vagy multifaktoros autentikációhoz (beléptetéshez) kapcsolódó SMS küldést végezhetünk el. Meghívást követően a megadott számra a BIP API egy token (rövid ideig érvényes kódot) küld ki SMS-ben. Biztonsági okokból a generált token a BIP API nem adja vissza - ezzel biztosítható, hogy a kód ne kerülhessen jogosulatlan kezekbe, és csak a címzett mobilkészülékén jelenjen meg.

Következő lépésként a beléptetést végző rendszerben be kell kérni a felhasználótól az SMS-ben megkapott token. Ezt a `tfavalidate` művelettel ellenőrizhetjük le (ld. a következő pontban).

A generált token kód jellemzői:

- A token alapértelmezésben 6 karakter hosszú, melyet az API a kísérszöveg után fűz vagy a kísérszövegbe behelyettesít.
- Paraméterekkel a szöveg szabadon megadható, és a token hossza is szabadon állítható.
- A token kis ékezet nélküli betűkből és számokból áll, pl: `9u49a5`.
- A rendszer a hibalehetőségek csökkentése érdekében nem használja a tokenekben a következő, könnyen összetéveszthető karaktereket: l, i, j, o, 0, 1.

A művelethez szükséges URL:

<https://api.bipkampany.hu/tfasendtoken>

tfasendtoken paraméterek

★ - kötelező paraméter

key	★ Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
message	★ Ezt az üzenetet küldi ki a rendszer, amely után fűzi a generált token szövegét. Ha az üzenet szövegében a <code>%token%</code> szöveget beírjuk, az API ennek helyére helyettesíti be majd a generált tokenkódot.
	A paraméterre emellett a <code>sendsms</code> művelet leírása érvényes (üzenethossz, használható karakterek, stb.).
number	★ A címzett telefonszáma nemzetközi formátumban, pl. 36309991111. A telefonszám kizárólag számokat tartalmazhat. A rendszer hibát jelez, ha nem támogatott célhálózatot (mobil körzetszámot) észlel.
	A rendszer a kétfaktoros autentikáció esetén nem veszi figyelembe a címzett korábbi leiratkozását az SMS-ek fogadásáról (ha történt ilyen a fiókban), hiszen a bejelentkezésre vonatkozó igény egy újabb SMS fogadási szándéknak/hozzájárulásnak tekinthető, amit másképp ellehetetlenítenénk.
appid	★ Kötelező paraméter, értéke egy tetszőleges tartalmú alkalmazás azonosító (pl. <code>NetbankLogin</code> vagy <code>webshopAdmin</code>). A BIP API-val így több saját rendszerét is kétfaktoros autentikációval láthatja el anélkül, hogy több BIP fiókot kellene létrehoznia. (Az <code>appid</code> használata nélkül ha egy felhasználó rövid időn belül két saját rendszerébe is megpróbálna belépni, a rendszer nem tudná kiválasztani az adott számra küldött még érvényes két token közül, hogy melyik rendszerben melyik az érvényes, amikor a <code>tfavalidate</code> művelettel a token ellenőrzése zajlik).
expiration	A token érvényességi ideje percben. Alapértelmezés szerint (vagy érvénytelenül megadott érték esetén) a tokenek 10 percig érvényesek. Értéke 1 és 1440 között lehet (1 perc és 1 nap közötti érvényesség).
tokenlength	A token hossza, alapértelmezés szerint (vagy érvénytelenül megadott érték esetén) 6 karakter. Értéke 1 és 16 között lehet.
senderid	A <code>sendsms</code> művelet paraméterezése szerint.
type	A <code>sendsms</code> művelet paraméterezése szerint.
format	A <code>sendsms</code> művelet paraméterezése szerint.

callback	A <code>sendsms</code> művelet paraméterezése szerint.
referenceid	A <code>sendsms</code> művelet paraméterezése szerint.
timetosend	A <code>sendsms</code> művelet paraméterezése szerint (ennél a műveletnél használatát nem javasoljuk)

Például:

```
https://api.bipkampany.hu/tfasendtoken?key=SAJAT_API_KEY
&number=36201234567&appid=WebshopLogin&message=A belépéshez szükséges kód:
&tokenlength=4&expiration=15
```

Az SMS átjáró lehetséges válaszai teljesen megegyeznek a `sendsms` művelet lehetséges válaszaival, hiszen itt is egy üzenetfeladás zajlik le.

2.2.2 Token visszaellenőrzése: a „`tfvalidatetoken`” művelet

A `tfasendtoken` művelettel kiküldött SMS token visszaellenőrzésére szolgál. A BIP API a művelet meghívásakor a felhasználó által begépett tokent várja paraméterben, és vagy a 0-s „OK” kóddal („token rendben”) válaszol, vagy valamilyen hibával. A token az érvényességi időn belül végig érvényes marad, így ha pl. hálózati okokból nem sikerül a `tfvalidatetoken` válaszát fogadni, még újrapróbálkozhat a felhasználó a belépéssel újabb token küldése nélkül is. A `tfainvalidatetoken` művelettel a tokent sikeres belépést követően egyszerűen érvényteleníthetjük.

A BIP API méri, hogy egy percen belül egy adott appid telefonszám kombinációra hány `tfvalidatetoken` kérés érkezik. Ha egy percen belül 15-nél több ilyen kérés érkezik, azt a BIP API automatizált brute force betörési kísérletnek tekinti, és a további ellenőrzési lehetőségeket a következő percre elutasítja. Így óránként max. 900 ilyen kérést dolgoz fel az API telefonszámonként, emiatt rövid tokenlejárati idő esetén közel nulla az esélye egy ilyen támadás sikerének (6 karakteres token esetén pl. 225 órát igényelne az összes lehetőség végigpróbálgatása, miközben 10 perces lejáratú tokennél csak 150 próbálkozási lehetőség van).

A művelethez szükséges URL:

```
https://api.bipkampany.hu/tfvalidatetoken
```

tfvalidatetoken paraméterek

🚩 - kötelező paraméter

key	🚩 Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
appid	🚩 A <code>tfasendtoken</code> műveletnél megadott <code>appid</code> paraméter.
number	🚩 A felhasználó telefonszáma nemzetközi formátumban, pl. 36309991111.
token	🚩 A felhasználó által begépett token.
format	A válasz formátuma, lehetséges értékek: <code>json</code> vagy <code>url</code> . Ha <code>json</code> , akkor JSON kódolt értékeket adunk vissza, ha <code>url</code> , akkor URL paraméterekben kódolt választ küldünk. Alapértelmezés: <code>url</code> .

Például:

```
https://api.bipkampany.hu/tfvalidate?key=SAJAT_API_KEY
&number=36201234567&appid=WebshopLogin&token=a1b2c3
```

A lehetséges **result**, **code**, és **message** válaszkombinációk:

result	code	message
OK	0	A kérést sikeresen teljesítettük
ERR	310	A token lejárt.
ERR	311	A megadott token nem egyezik meg a kiküldött tokennel.
ERR	312	A megadott számhoz és appid-hez nem találtunk token, vagy már lejárt.
ERR	313	Túl sok token validálási kérés egy percen belül.

2.2.3 Token érvénytelenítése: a „tfainvalidatetoken” művelet

A **tfasendtoken** művelettel kiküldött token érvénytelenítésére szolgál. A beléptetés biztonságosságát növeli, ha a sikeresen beléptetett felhasználóknál ezzel a művelettel érvénytelenítjük a token, hiszen így a token megszerzésével (pl. leolvasásával) nem tud más felhasználó hozzáférni a rendszerhez az érvényességéből még hátralévő idő alatt.

A művelethez szükséges URL:

<https://api.bipkampany.hu/tfainvalidatetoken>

tfainvalidatetoken paraméterek

🚩 - kötelező paraméter

key	🚩	Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
appid	🚩	A tfasendtoken műveletnél megadott appid paraméter.
number	🚩	A beléptett felhasználó mobilszáma.
format		A válasz formátuma, lehetséges értékek: json vagy url . Ha json , akkor JSON kódolt értékeket adunk vissza, ha url , akkor URL paraméterekben kódolt választ küldünk. Alapértelmezés: url .

Például:

[https://api.bipkampany.hu/tfainvalidate?key=SAJAT_API_KEY
&number=36201234567&appid=WebshopLogin](https://api.bipkampany.hu/tfainvalidate?key=SAJAT_API_KEY&number=36201234567&appid=WebshopLogin)

A lehetséges **result**, **code**, és **message** válaszkombinációk:

result	code	message
OK	0	A kérést sikeresen teljesítettük
ERR	312	A megadott számhoz és appid-hez nem találtunk token, vagy már lejárt.

A felhasználás szempontjából ez esetben a hibás válasz is elfogadható, hiszen ha a token már nem találja meg az API nyilvántartva, akkor biztosan nem is lehet érvényes.

2.3 Fiókkal kapcsolatos műveletek

2.3.1 Profiladatok lekérdezése: a „getprofile” művelet

Ezzel a művelettel lekérdezhethetjük a saját regisztrációnkhoz tartozó információkat a rendszertől: cégszámok, postázási cím, engedélyezett IP cím(ek), feladóazonosító és API információk.

A lekérdezéshez szükséges URL:

<https://api.bipkampany.hu/getprofile>

getprofile paraméterek

📌 - kötelező paraméter

key 📌	Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
format	A válasz formátuma, lehetséges értékek: json vagy url . Ha json , akkor JSON kódolt értékeket adunk vissza, ha url , akkor URL paraméterekben kódolt választ küldünk. Alapértelmezés: url .

Például:

https://api.bipkampany.hu/getprofile?key=SAJAT_API_KEY

Példa az SMS átjáró válaszára:

```
result=OK&code=0&message=A k%C3%A9r%C3%A9st sikeresen teljes%C3%ADtott%C3%Bck&name=TestUser&company=TestCompany&billing_address1=Example%20str%201&billing_city=New%20York&billing_country=United%20States&billing_address2=&billing_zipcode=12345&postal_address1=&postal_address2=&postal_zipcode=&postal_city=&postal_country=&postal_name=&senderids=BipKampany&api_ipaddresses=&api_callbackurl=
```

vagy JSON formátum (format=json) esetén:

```
{  "result": "OK",  "code": 0,  "message": "A k\u00e9r\u00e9st sikeresen teljes\u00edtett\u00fck",  "name": "TestUser",  "company": "TestCompany",  "billing_address1": "Example str 1",  "billing_address2": "",  "billing_city": "New York",  "billing_country": "United States",  "billing_zipcode": "12345",  "postal_address1": "",  "postal_address2": "",  "postal_zipcode": "",  "postal_city": "",  "postal_country": "",  "postal_name": "",  "senderids": "BipKampany",  "api_ipaddresses": "",  "api_callbackurl": ""}
```

Válaszként az üzenetfeladáshoz hasonlóan **result**, **code**, és **message** értéket kapunk vissza. Hibaüzenet akkor fordulhat elő, ha a hozzáférési adatokat hibásan adtuk meg (ld. API hibakódok a 3-as fejezetben).

2.3.2 Egyenleg lekérdezése: a „getbalance” művelet

Az egyenleg lekérdezéséhez szükséges URL:

<https://api.bipkampany.hu/getbalance>

getbalance paraméterek

🚩 - kötelező paraméter

key 🚩 Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).

format A válasz formátuma, lehetséges értékek: **json** vagy **url**. Ha **json**, akkor JSON kódolt értékeket adunk vissza, ha **url**, akkor URL paraméterekben kódolt választ küldünk. Alapértelmezés: **url**.

Például:

https://api.bipkampany.hu/getbalance?key=SAJAT_API_KEY

Példa az SMS API válaszára:

```
result=OK&code=0&message=A k%C3%A9r%C3%A9st sikeresen teljes%C3%ADtott%C3%Bck&currency=HUF&balance=450000&limit=300000
```

vagy JSON formátum (format=json) esetén:

```
{
  "result": "OK",
  "code": 0,
  "message": "A k\u00e9r\u00e9st sikeresen teljes\u00edtett\u00fck",
  "currency": "HUF",
  "balance": 450000,
  "limit": 300000
}
```

Válaszként tehát az üzenetfeladáshoz hasonlóan **result**, **code**, és **message** értéket kapunk vissza ami itt kiegészül a **currency**, **balance** és **limit** értékekkel:

paraméter neve	leírás	példa	
result	Művelet sikeressége	OK	vagy ERR
code	Válasz kódja	0	vagy 304
message	Válasz szövege	A kérést sikeresen teljesítettük	vagy Egyenlege lejárt, vagy egyenlegének beállítása hiányos.
balance	Egyenleg értéke	30.00	
currency	Az egyenleg pénzneme	HUF	
limit	Postpaid ügyfél esetén a kiegyenlítés nélkül felhasználható maximális egyenleg, pénzneme a currency paraméterben található. Prepaid ügyfél esetén értéke 0.	300000	

3 API hibakódok

A hibaüzeneteket célszerű az API-t megszólító alkalmazásban üzenetenként is naplózni (pl. a hibakódokat a kiküldött üzeneteket tartalmazó adatbázistábla egy szöveges mezőjébe gyűjteni, a legutolsó hibakódot pedig egy szám típusú mezőben tárolni), mivel így esetleges hibát követően újra könnyen le tudjuk válogatni az újraküldendő üzeneteket.

Ezentúl érdemes lehet minden ERR típusú eredménynél e-mailben vagy egyéb módon azonnali értesítést beállítani, ami segíthet a mielőbbi hibaelhárításban.

0 - sikeres művelet

result	code	message
OK	0	A kérést sikeresen teljesítettük

10x – paraméterezési hibák (formátumhibák)

result	code	message
ERR	100	Kötelező paraméter hiányzik: <code>paraméternév</code>
ERR	101	A <code>paraméternév</code> paraméterben csak számok szerepelhetnek.
ERR	102	A telefonszám nem támogatott vagy nem nemzetközi formátumban van.
ERR	103	Az üzenet hosszabb a maximális 459 karakternél.
ERR	104	A megadott feladóazonosító (<code>senderid</code>) nem engedélyezett
ERR	105	Az üzenet szövege tiltott karaktereket tartalmaz
ERR	106	A callback paraméter hibás
ERR	107	A megadott időzítési időpont hibás
ERR	108	A megadott referenciaid-vel függőben lévő üzenet nem található (<code>cancelsms műveletnél</code>)
ERR	109	Ismeretlen paraméter: " <code>paraméternév</code> "
ERR	110	Ismeretlen művelet (target érték): " <code>művelet</code> "
ERR	111	A " <code>format</code> " paraméter értéke érvénytelen: " <code>érték</code> "
ERR	115	Az email üres vagy tartalma nem feldolgozható formátumban van. (<code>email API használata esetén</code>)

30x – művelet végrehajtás közben történt hibák (hozzáférési, egyenleg vagy egyéb hibák)

result	code	message
ERR	300	A felhasználói név vagy jelszó hibás, vagy a hozzáférés nem engedélyezett
ERR	301	A(z) <code>ipcím</code> IP címről nem adható fel üzenet. Engedélyezze a címet a Beállítások menüpontban!
ERR	302	Jelenlegi egyenlege (<code>jelenlegi egyenleg</code>) nem elegendő az üzenet kiküldéséhez (<code>üzenet díja</code>)
ERR	303	Az Ön hitelkerete nem teszi lehetővé az üzenet kiküldését
ERR	304	Egyenlege lejárt, vagy egyenlegének beállítása hiányos.
ERR	305	Az API kulcs hibás, az API hozzáférés nincs bekapcsolva, vagy hozzáférés letiltva.
ERR	310	A token lejárt. (<code>tfvalidate műveletnél</code>)
ERR	311	A megadott token nem egyezik meg a kiküldött tokenel. (<code>tfvalidate műveletnél</code>)

ERR	312	A megadott számhoz és appid-hez nem találtunk tokenet, vagy már lejárt. (<i>tfvalidate műveletnél</i>)
ERR	313	Túl sok token validálási kérés egy percen belül. (<i>tfvalidate műveletnél</i>)
ERR	315	A címzett ebben a fiókban leiratkozott az SMS fogadásról (te lefonszám).

50x – szolgáltatás oldali hiba

result	code	message
ERR	500	A rendszer átmeneti hibája miatt a szolgáltatás jelenleg nem elérhető. Munkatársaink értesítést kaptak, és megkezdik a probléma elhárítását

4 Callback szolgáltatás

Az API-n keresztül kiküldött üzenetről visszajelzést tudunk biztosítani. Ezt a szolgáltatóktól visszakapott DLR-ek (delivery receipt, kézbesítési jelentés) biztosítják aszinkron módon – tehát a DLR-ek a feladást követően, de előre nem meghatározható időben keletkeznek.

4.1 A visszajelzések fogadásának feltételei

- Léteznie kell egy HTTP protokollon keresztül elérhető címnek (scriptnek), ami lekezeli a visszajelzéseket.
- A fogadó oldal teszteléséhez a webes felület **Beállítások** oldalán segédeszközt is adunk: a megadott URL-re minta callback hívást végző funkcióval üzenetfeladás nélkül is könnyen elkészíthető a szükséges programkód.
- Csak API-n keresztül küldött üzenetekről adunk visszajelzést – a BIP Kampány webes felületén át küldött üzenetekről a webes felületen olvashatóak a visszajelzések.
- A küldéskor hívott `sendsms` műveletben meg kell adnunk
 - a `callback` paramétert, amellyel beállítható, hogy az adott üzenet mely státuszairól kérünk visszajelzést
 - és a `referenceid` paramétert, amit a visszajelzés fogadásakor változatlanul formában visszaadunk – ezzel azonosítható be az a feladott üzenet, amelyhez a visszajelzést átadjuk.

4.2 A callback URL meghívása

A BIP SMS a beérkező állapotüzeneteket a megadott URL-re adja át az alábbiakban bemutatott GET paraméterekkel.

Ha az URL nem elérhető, a callbacket a későbbiekben újrapróbáljuk: 5 perc elteltével percenként még 55 percig, majd egy-egy alkalommal 1, 2, 3, 4, 24, 48 és 72 óra elteltével). A három napnál régebbi, kézbesíthetetlen visszajelzéseket rendszerünk nem próbálja meg tovább kiküldeni.

4.3 Példa egy üzenet feladására és a meghívott callback URL-ekre

(az olvashatóság kedvéért a paramétereket ezúttal **nem** kódoltuk URL formátumra):

- 1) A callback számára beállított callback URL értéke legyen a példa kedvéért a következő:

<https://example.com/callback.aspx>

- 2) Üzenet feladásakor használnunk kell a `referenceid` és a `callback` paramétereket is:

```
https://api.bipkampany.hu/sendsms
?key=SAJAT_API_KEY           &number=36309991111
&message>Hello World         &referenceid=123
&callback=SENT,ACCEPTED,WAITING,DELIVERED,REJECTED,UNDELIVERABLE,MISC,STOPPED,EXPIRED,STATUSLOST
```

Ha minden üzenetállapotot szeretnénk megkapni, nem kell felsorolnunk őket minden alkalommal, ehelyett használhatjuk egyszerűen az `ALL`-t is:

```
&callback=ALL
```

- 3) A BIP a beérkező állapotjelzéseket folyamatosan feldolgozza és továbbkézbesíti. A BIP a `sendsms` műveletben átvett `referenceid=123`-as paramétert visszaadja a hívásokban:

```
https://example.com/callback.aspx
?referenceid=123           &number=36309991111   &status=SENT
&message=Távoli SMSC-nek továbbítva &price=30.00         &timestamp=2011-12-31 22:51:21
```

```
https://example.com/callback.aspx
?referenceid=123           &number=36309991111   &status=ACCEPTED
&message=Távoli SMSC által elfogadva &price=30.00         &timestamp=2011-12-31 23:01:31
```

```
https://example.com/callback.aspx
?referenceid=123           &number=36309991111   &status=DELIVERED
&message=Kézbesítve       &price=30.00         &timestamp=2011-12-31 23:59:59
```

A callback URL-t a BIP rendszere mindig a következő paraméterekkel hívja meg:

Paraméter neve	Leírás	Példa
referenceid	Az API-nak az <code>sendsms</code> művelet során a <code>referenceid</code> paraméterben átadott tetszőleges üzenetazonosító	5447
number	Telefonszám	36309991111
status	Visszajelzés kódja (lásd a következő táblázatban)	DELIVERED
message	Visszajelzés szövege (UTF-8 kódolással)	Kézbesítve
price	Üzenet ára (mindig ugyanaz az összeg, nem adódik össze, nem változik a callbackek során)	30.00
timestamp	A visszajelzés beérkezésének időpontja: nem változik, így ha a callback célokra szolgáló cím átmenetileg nem elérhető, a megismételt callback kísérletek során is az eredeti időpontot tartalmazza.	2012-02-01 21:59

A `status` paraméter lehetséges értékei:

status	message	magyarázat
❗ SENT	Távoli SMSC-nek továbbítva	Az SMS központok közötti üzenetátadás megkezdődött
❗ ACCEPTED	Távoli SMSC által elfogadva	Az SMS központok átvették az üzenetet – ha a mobilszámhoz tartozó előfizetés él, és a készüléket az érvényességi időn belül bekapcsolva tartják, az üzenet célbaér. Kikapcsolt készülék esetén visszakapcsoláskor a szolgáltatók saját ismétlési mintáik szerint próbálják újraküldeni az üzenetet.
❗ WAITING	Várakozás a céltelefonra	Átmenetileg nem elérhető készülékek esetén.
✅ DELIVERED	Kézbesítve a mobilkészülékre	A mobilkészüléktől visszajelzést érkezett az üzenet sikeres átvételéről fogadásáról
❌ REJECTED	Távoli SMSC által visszautasítva	Hálózati kapcsolathiba vagy szolgáltatói rendszerhibák esetén
❌ UNDELIVERABLE	Kézbesíthetetlen a mobilkészülékre	A mobilkészülék az érvényességi időn belül nem volt elérhető (pl. lejárt prepaid kártya, nem létező telefonszám, kikapcsolt készülék, térerőn kívül tartózkodott, adatátviteli hiba, stb.)
❌ MISC	Egyéb hiba miatt kézbesíthetetlen	Szolgáltató oldali általános hibaüzenet.
❌ STOPPED	A kiküldés leállítva	Csak a szolgáltatók által kezdeményezett leállítás esetén.
❌ EXPIRED	Érvényességi idő lejárt	Az üzenet az érvényességi időn belül nem volt kézbesíthető.
❌ STATUSLOST	A státusz elveszett	Ritka, csak szolgáltatói hiba esetén.

5 PHP osztály a BIP API-hoz

A HTTP API „köré” elkészítettünk egy egyszerű PHP osztályt, aminek segítségével az SMS küldés még könnyebben megvalósítható.

5.1 A PHP osztály metódusai

Ha egy paramétert át szeretnénk ugrani a metódushívásokban – pl. a `sendSMS` metódusban nincs szükségünk a `$timetosend` beállítására, de a `$type`-ot szeretnénk megadni –, használjunk `null` értéket az adott paraméter helyén.

⇒ konstruktor

```
void BipKampanyAPI::__construct( string $key )
```

Konstruktor.

Paraméterek:

`$key` – hozzáféréshez használt egyedi API kulcs

⇒ sendSMS()

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

SMS feladása.

Paraméterek:

`$number` – a telefonszám nemzetközi formátumban (pl. 36309991111)

`$message` – az üzenet UTF-8 kódolásban

`$senderid` – egy már jóváhagyott feladóazonosító (pl. BipKampany)

`$timetosend` – opcionális: az üzenet időzítése ÉÉÉÉ-HH-NN óó:pp:mp formátumban (pl. 2011-11-25 12:08:59)

`$type` - opcionális: az üzenet típusa. Egyetlen támogatott értéke: `unicode`

`$callback` - opcionális: a kért callback jelentések típusának felsorolása, vagy `ALL`, ha mindegyik kézbesítési jelentéstípust kérjük.

`$referenceid` - opcionális: a callback hívások során a BIP által változatlan formában visszaadott tetszőleges azonosító. Ha megadjuk, kötelező a `$callback` paraméter megadása is.

Visszatérési értéke: az API válasza tömbként. Hiba esetén kivételt vált ki.

⇒ getBalance()

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

Egyenleg lekérdezése.

Visszatérési értéke: az API válasza tömbként, benne a `balance` és a `currency` tömbindexek tartalmazzák az egyenleget. Hiba esetén kivételt vált ki.

⇒ cancelSMS()

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

Korábban időzítéssel feladott üzenet visszavonása.

Paraméterek:

`$referenceids` – az az üzenetazonosító, amelyet az üzenet feladásakor használtunk. Több üzenetazonosítót is megadhatunk vesszővel elválasztva.

Ügyeljünk sok azonosító megadásánál arra, hogy az így összeálló string hossza ne legyen több 65000 byte-nál.

Visszatérési értéke: az API válasza tömbként. Hiba esetén kivételt vált ki.

⇒ `getCharset()`

`array BipKampanyAPI::getCharset()`

Küldés során használható karakterek lekérdezése.

Visszatérési értéke: az API válasza tömbként, benne a `charset` tömbindex tartalmazza a támogatott karaktereket (UTF-8 kódlapban). Hiba esetén kivételt vált ki.

⇒ `getLastResult()`

`mixed BipKampanyAPI::getLastResult()`

Az utoljára megkapott API válasz értékének lekérdezése.

Visszatérési értéke: null, ha még nem volt API által visszaadott válasz (azaz ha API művelet előtt próbáltuk lekérdezni). Ha már volt válasz, a választ kapjuk vissza tömbként.

5.2 A PHP osztály attribútumai

⇒ `logFilename`

`string BipKampanyAPI::logFilename`

Naplóállomány teljes elérési útja és neve. Ha ki van töltve, a BIP API ebbe az állományba naplóz minden meghívandó HTTP URL-t a meghívást megelőzően és sikeres hívás esetén a visszakapott választ is egy `var_export()`-ált tömb formájában.

⇒ `callingMethod`

`string BipKampanyAPI::callingMethod`

A HTTP API meghívásának módja.

Alapértelmezés szerinti értéke `file_get_contents`, ami a PHP beépített `file_get_contents()` függvényét használja. Ennek használatához a PHP `allow_url_fopen` beállításának `on` állapotban kell lennie - az osztály ezt is ellenőrzi, és ha ettől eltér, kivételt vált ki.

Másik lehetséges értéke `curl` - ilyen beállítás esetén a CURL extensiót használja az osztály a HTTP hívásokhoz. Ennek használatához a CURL kiterjesztésnek telepítve kell lennie, ezt az osztály ellenőrzi, ha hiányzik, kivételt vált ki.

5.3 Példák PHP nyelven

A weboldalunkról letölthetőek a következő API példák is PHP fejlesztők számára:

- `pe1da1_http.php` – a legegyszerűbb megvalósítás: a HTTP API meghívása a beépített `file_get_contents()` segítségével SMS küldéshez. Egyben bemutatja a minimálisan ajánlott hibakezelést is.
- Példák az API osztály használatára:
 - `pe1da2_api_egyszeru.php` – az API osztály használatának legegyszerűbb módja: példányosítás, küldés és kivételkezelés
 - `pe1da3_api_teljes.php` – az API osztály naplózás funkciójának használata, szükség esetén fejlettebb kivételkezelés, időzítés, Unicode üzenettípus használata
 - `pe1da4_api_curl.php` – CURL kiterjesztés használatának beállítása az API osztályban
 - `pe1da5_api_egyenleg.php` – egyetlen lekérdezése az API osztály segítségével
 - `pe1da6_api_karakterkeszlet.php` – karakterkészlet lekérdezése az API osztály segítségével

6 E-mail-SMS átjáró (SMTP API)

A BIP API e-mailek segítségével is elérhető kétféle módon:

- Ha várhatóan szükségünk lehet többféle paraméter használatára, a 6.1 pontban leírt részletes módszert használhatjuk, itt könnyen áttekinthetőek és megadhatóak a különféle beállítások
- Egyszerűbb esetekben a 6.3 pontban bemutatott egyszerűsített API változat is elég lehet, ahol a címzett telefonszámát és a szöveget még egyszerűbben adhatjuk meg.

Mindkét változat esetén érdemes tudni a következőket:

Az e-mail formája

- Az egyszerűbb használhatóság kedvéért az API egész intelligens módon detektálja az e-mailek tartalmát, akár a több részes és a formázásokkal teli HTML leveleket is feldolgozza (technikailag a legelső megtalált szöveges (text/plain), annak hiányában a legelső HTML (text/html) levélrész HTML-telenített változatát használja).
- Emiatt nincsenek megkötések az e-mail formátumára vonatkozóan - de ha lehetőségünk van rá, igyekezzünk a legegyszerűbb text/plain e-maileket küldeni és UTF-8-at használni.
- Ha programozottan hozzuk létre az e-maileket, használjuk a küldött levélben a helyes **Content-Type** fejléccet, így biztosan gond nélkül jutnak el az ékezetes karaktereket és jelek az API-hoz, pl:

```
Content-Type: text/plain; charset=UTF-8; format=flowed
```

Hibakezelés

Az API hiba esetén **e-mailben küld választ**. Azt, hogy milyen e-mail címre megy ki a válaszlevél, a következő sorrend alapján dönti el az API:

- ha meg van adva, akkor a **replyto** paraméterrel megadott válaszcím
- egyébként – szintén csak ha meg van adva –, akkor az a cím, amelyet az API hibák fogadására beállítottunk (webes felület Beállítások menüpontjában az API beállításai között)
- egyébként a BIP regisztrációnál (és későbbi webes belépésnél) használt cím
- bármilyen egyéb hiba esetén a levél *From*: fejléc szerinti feladójának küldjük az üzenetet.

6.1 Üzenetküldés a részletes e-mail API-val

Ennél az API változatnál leveleinket az alábbi címre kell küldenünk:

sms@bipkampany.hu

Minden egyes elküldött e-mail egy művelethívásnak felel meg.

A levélben soronként adjuk meg a paramétereket, a sorok formátuma a következő:

parameternev:parameter_erteke

Az e-mailek **témája** (*Subject*) ennél a használati módnál tetszőleges lehet, mivel azt a rendszer nem próbálja meg értelmezni.

Ha szeretnénk látni, mit lát az API levelünkből (vagyis miként értelmezi a beállításainkat), a 6.2 részben leírt hibakereső módszerrel könnyen ellenőrizhetjük.

Az üzenetküldéshez az alábbi paramétereket használhatjuk:

E-mail paraméterek üzenetküldéshez

🔴 - kötelező paraméter

target	🔴 A művelet azonosítója, értéke SMS küldéshez kötelezően <code>sendsms</code>
key	🔴 Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
message	🔴 A feladni kívánt üzenet, maximum 459 karakter hosszúságban (3 db SMS felhasználásával összefűzött üzenet). Alapértelmezésben a GSM 03.38 szerinti karakterek engedélyezettek (a használható karakterek lekérdezhetőek a <code>getcharset</code> művelettel, ld. a 2.3 pontban). Egyéb karakterekhez <code>type=unicode</code> beállítás szükséges, ekkor a maximális hossz 201 karakter. További részletekért lásd még az 1.3 pontot a dokumentációban.
number	🔴 A címzett telefonszáma nemzetközi formátumban, pl. 36309991111. A telefonszám kizárólag számokat tartalmazhat. A rendszer hibát jelez, ha nem támogatott célhálózatot (mobil körzetszámot) észlel, vagy ha a címzett leiratkozott az SMS fogadásról az adott fiókban.
senderid	A mobilkészüléken feladóként megjelenő feladóazonosító (telefonszám vagy szöveg). Ha nincs megadva, alapértelmezésben a <code>Bipkampany</code> feladóazonosítót használjuk. A webes felületen igényelhető saját feladóazonosító a Beállítások menüpontban.
type	Az üzenet típusa. Jelenleg egy értéket támogat: <code>unicode</code> . Ha nem adjuk meg, az üzenet GSM 03.38 karakterkészlettel kerül kézbesítésre, <code>type:unicode</code> esetén azonban a teljes Unicode karakterkészletet használhatjuk az üzenet szövegében. További részletekért lásd az 1.3 pontot a dokumentációban.
replyto	Az az e-mail cím, amire hiba esetén az API e-mailben hibajelentést küld.
callback	Ha BIP által indított HTTP hívás segítségével információt kérünk vissza az üzenetek állapotáról, itt kell felsorolnunk azokat az állapotkódokat, amelyekről a callbacket kérjük. Az állapotkódokat vesszővel elválasztva soroljuk fel. Ha mindegyik állapotot kérjük, használjuk az <code>ALL</code> értéket a felsorolás helyett. További részletek a callback ismertető fejezetben.
referenceid	Kötelező <code>callback</code> használata esetén. Egy tetszőleges felépítésű, tartalmú egyedi üzenetazonosító: lehet pl. egy adatbázis rekord azonosítója, vagy valamilyen összetett sztring, pl. <code>523_5328_1</code> <ul style="list-style-type: none"> Ezzel az azonosítóval lesz azonosítható később az üzenet, ha időzítve adtuk fel, de szeretnénk majd visszavonni (<code>cancelsms</code> művelet). Ha a <code>callback</code> paramétert is beállítottuk, az API visszaadja ezt az azonosítót a

HTTP hívás részeként a Beállításoknál megadott callback URL-re. Ha ilyen URL nincs beállítva, a callback nem lehetséges.

timetosend Időzítés időpontja, amikor az üzenet kézbesítésre kerül. Formátum: **ÉÉÉÉ-HH-NN óó:pp:mm**, pl. **2015-12-11 08:10:12**

A paraméterben használhatjuk a **CURRENT_DATE** szövegrészt is. Ez akkor segíthet, ha a feladó rendszerben (pl. egy CRM rendszer) nem tudunk beilleszteni változó tartalmat (aktuális dátumot), az időzítésre azonban mégis szükségünk van.

Ha pl. a CRM rendszer akár éjszaka nem szeretnénk, hogy reggel 8 óra előtt SMS üzenetet kapjanak a címzettek, használhatjuk ezt az időzítést a timetosend paraméterben: **CURRENT_DATE 08:00:00**. Így a 8:00 előtt érkező üzenetek 8:00-kor, a később érkezők pedig azonnal ki fognak menni a címzetteknek.

Példa egyszerű üzenetküldésre:

```
target:sendsms
key:SAJAT_API_KEY
number:36201234567
message>Hello World!
```

Példa további opciók használatára:

```
target:sendsms
key:SAJAT_API_KEY
number:36201234567
message>Hello World!
senderid:BipKampany
type:unicode
replyto:fejleszto@example.com
callback:ALL
referenceid:4122321
timetosend:2015-12-11 08:10:12
```

Válaszüzenet

Hiba esetén a BIP API e-mailben küld választ. A levél témájában is megjelenik a hibakód és a hibaüzenet, ezeket a 3-as fejezetben mutatjuk be részletesen.

Példa egy hibát jelző e-mailre:

A válaszlevél témája (Subject):

[BIP API] 305: Az API kulcs hibás, az API hozzáférés nincs bekapcsolva (ld. Beállítások menüpont), vagy hozzáférés letiltva.

A válaszlevél szövege:

Ez a levél fontos üzenetet tartalmaz a BIP SMS szolgáltatástól annak számára, aki az Ön cégének email->SMS forgalmáért felel (valószínűleg egy programozó vagy rendszergazda). Ha nem tudja, miért érkezett ez a levél, kérjük továbbítsa a levelet ennek a személynek! Ha velünk lépne kapcsolatba, várjuk levelét: info@bipkampany.hu

This e-mail contains an important message from BIP SMS service to the person responsible for your company's email->SMS traffic (possibly a programmer or a system administrator). If you don't know why you've received this message, please contact the programmer/developer in charge!
To contact us please write to: info@bipkampany.hu

Error details:

```
-----
result:ERR
code:305
message:Az API kulcs hibás, az API hozzáférés nincs bekapcsolva (ld. Beállítások menüpont), vagy hozzáférés letiltva.
```



```
Parsed parameters:
-----
key:SAJAT_API_KEY
number:36201234567
message>Hello World!
```

```
Original message:
-----
From sender@example.com Mon Jan 4 13:08:15 2018
Received: from localhost (localhost [127.0.0.1])
    by xxx.example.com (Postfix) with ESMTF id 613341ACD0
    for <sms@localhost>; Mon, 4 Jan 2018 13:08:15 0200 (CEST)
Received: from xxx.example.com ([127.0.0.1])
    by localhost (xxx.example.com [127.0.0.1]) (amavisd-new, port 10024)
    with ESMTF id P3lwba2Msk-q for <sms@localhost>;
    Mon, 4 Jan 2018 13:08:14 0200 (CEST)
Received: from [111.111.111.111] (hostname.example.com [222.222.222.222])
    (using TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits))
    (No client certificate requested)
    by bipkampany.hu (Postfix) with ESMTF id B21AD5C21A
    for <sms@bipkampany.hu>; Mon, 4 Jan 2018 13:08:13 0200 (CEST)
Message-ID: <AAAAAAAA.9000000@example.com>
Date: Mon, 04 Jun 2018 13:07:57 0200
From: sender@example.com
MIME-Version: 1.0
To: sms@bipkampany.hu
Subject: email API test
Content-Type: text/plain; charset=UTF-8; format=flowed
Content-Transfer-Encoding: 7bit
```

```
key:SAJAT_API_KEY
number:36201234567
message>Hello World!
```

6.2 Hibakeresés (sendsmsdebug)

A hibakeresés műveletet arra használhatjuk, hogy kiderítsük, milyen üzenetet kap meg tőlünk az API, és azt pontosan hogyan bontja fel paramétereiké (a művelet azonban nem végez teljes körű tartalmi ellenőrzést is a paraméterekben, csak a saját meghívásához szükséges paramétereit – **target**, **key** - ellenőrzi tartalmilag).

Használata egyszerű: csak a **target:sendsmsdebug** paramétert kell használnunk az üzenet szövegében az egyéb paraméterek mellett:

```
target:sendsmsdebug
key:kjdh325kjh2345
number:36201234567
message>Hello World!
```

A válaszul kapott e-mail pontosan ugyanazokat az információkat tartalmazza, amit üzenetküldésnél a hibaüzenetben kapunk.

E-mail paraméterek hibakereséshez

📌 - kötelező paraméter

target	📌	A művelet azonosítója, az email - SMS átjáró teszteléséhez kötelezően sendsmsdebug
key	📌	Saját API kulcs az azonosításhoz (ld. belépést követően a Beállítások menüpontban).
...?...		Tetszőleges számú és tartalmú egyéb paraméter, amit a művelet feldolgoz és a válaszban visszaad.

6.3 Egyszerűsített üzenetküldés az e-mail API-val

Legtöbb esetben és legtöbb felhasználási célra nincs szükség az API minden tudására, illetve egyes rendszerek nem teszik lehetővé, hogy annak megfelelő e-mailt küldjünk. Ezekre az esetekre fejlesztettük ki az egyszerűsített e-mail API-t, amelynek használatához a következő formában kell megírunk az e-mailt:

- Címzett: `36301234567@bipkampany.hu`
 - Téma (subject): `key=SAJAT_API_KEY`
 - Levél szövege: `Kedves ügyfelünk! Örömmel értesítjük, hogy csomagját futárszolgálatunk a holnapi napon 10-11 óra között kézbesíti.`
- természetesen a telefonszám tetszés szerint változik, a `@bipkampany.hu` végződés pedig állandó
 - a levél témájában (subject) adjuk meg egyedi API kulcsunkat a `key=` szövegrész után (ez a webes felületen a Beállítások menüpontban található meg az API hozzáférés bekapcsolásakor)
 - bár a rendszer törli a formázásokat, extra szóközöket és egyéb felesleges részeket a levéltörzsből, érdemes a levél szövegét egyszerű szöveges (text/plain) formátumban írni. A levélszöveg ne tartalmazzon semmi egyebet (sablon köszönést vagy elköszönést, aláírást, és lehetőleg képet vagy más csatolmányt se).
 - Ha mégis szükség van arra, hogy a 6.1-ben ismertetett további beállításokat használjuk, a levél témájában nemcsak a `key`-t, hanem további paramétereket is megadhatunk URL kódolt formában (bár épp emiatt ennek megadása körülményesebb lehet). Példa további paraméterek megadására a levél témájában (a paramétereket a 6.1 pontban ismertetjük):

`key=SAJAT_API_KEY&referenceid=4122321&callback=ALL`

7 REST SMS API

A BIP API műveletei egységes REST API felületen keresztül is elérhetőek.

A REST API használatára szinte minden igaz az alap HTTP API-nál leírtak közül, ezért fontos, hogy megismerjük az ott leírt részleteket is, hiszen a REST API az ott leírt API műveleteket foglalja szabványosított formátumba.

Különbségek a HTTP API-hoz képest

A REST API-n keresztül használható paraméterek szinte mindenben megegyeznek a 2. fejezetben bemutatott HTTP API paraméterezésével. A hívás formátumának eltéréséből adódó különbségek azonban vannak, ezek a következők (részletekért ld. 7.1-es pont):

- A paramétereket (payload) az API kétféle kódolással fogadja el:
 - `Content-Type: application/x-www-form-urlencoded`
 - `Content-Type: multipart/form-data`
- A felhasználói azonosításhoz az `Authorization: AccessKey SAJAT_API_KEY` fejléct kell használnunk (ez a HTTP API `key` paraméterének felel meg).
- Az `Accept:` headerrel irányítható, hogy milyen formátumban válaszoljon az API (ez a HTTP API `format` paraméterének felel meg).

REST API request példa

The screenshot shows a REST client interface with the following details:

- URL:** `https://api.bipkampany.hu/sendsms`
- Method:** POST
- Region:** US
- Buttons:** Send, Content (2), Authorization, Headers (1), Raw (8), and various utility icons.
- Request Body (Raw):**

```
POST /sendsms HTTP/1.1
Authorization: AccessKey SAJAT_API_KEY
Host: api.bipkampany.hu
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Content-Length: 32

message=teszt&number=36201234567
```
- Status:** 200 (OK) Time: 735 ms Size: 0.11 kb
- Response Body (JSON):**

```
{
  "result": "OK",
  "code": 0,
  "message": "A k\u00e9r\u00e9s sikeresen teljes\u00e9dtett",
  "price": 15.0,
  "split": 1
}
```

7.1 Támogatott HTTP fejlécek

⇒ Authorization 🚩

Az API hozzáférési jogosultság ellenőrzéséhez szükséges kötelező paraméter. A `SAJAT_API_KEY` a webes felület Beállítások oldalán az API beállításoknál található meg.

Példa:

```
Authorization: AccessKey SAJAT_API_KEY
```

Hibakezelés:

Hibás API kulcs esetén az API válasza:

header: `403 Fordibben`

```
body: { "result": "ERR", "code": 305, "message": "Az API kulcs hibás, az API hozzáférés nincs bekapcsolva (ld. Beállítások menüpont) vagy letiltva." }
```

⇒ Content-Type 🚩

Az API hívás tartalmának (body) formátuma. **Megadása kötelező.**

Az elfogadott formátumok:

```
Content-Type: application/x-www-form-urlencoded  
Content-Type: multipart/form-data
```

Hibakezelés:

Nem támogatott Content-Type esetén az API válasza:

header: `415 Unsupported Media Type`

```
body: { "result": "ERR", "code": 415, "message": "A Content-Type fejlécben szereplő MIME típus nem támogatott" }
```

⇒ Accept

Az API válaszában elvárt formátuma (URL kódolt, JSON vagy XML). Ha a fejlécben többféle formátumot is felsoroltuk, az API a `q` (quality) értéke alapján választ közülük (ha meg volt adva). Megadása nem kötelező, ekkor a válasz `application/x-www-form-urlencoded` formátumban lesz.

Példa:

```
Accept: application/x-www-form-urlencoded  
Accept: application/*  
Accept: */*
```

(mind a három fenti Accept fejléc azonos `x-www-form-urlencoded` válaszformátumot eredményez)

```
Accept: application/json;q=0.9  
Accept: application/xml;q=0.9  
Accept: application/json;q=0.9, application/xml;q=1.3
```

Hibakezelés:

Ha hiányzik az Accept header vagy a várt típust nem támogatja, akkor az API válasza:

header: `406 Method Not Allowed`

```
body: { "result": "ERR", "code": 406, "message": "Nem elfogadható MIME típus az Accept fejlécben" }
```

7.2 Támogatott REST API műveletek

Művelet API végpontja	Metódus
https://api.bipkampany.hu/sendsms	POST
https://api.bipkampany.hu/getcharset	GET
https://api.bipkampany.hu/cancelsms	POST
https://api.bipkampany.hu/getbalance	GET
https://api.bipkampany.hu/tfasendtoken	POST
https://api.bipkampany.hu/tfvalidatetoken	POST
https://api.bipkampany.hu/tfainvalidatetoken	POST

A műveletek működésének, paraméterezésének leírása a dokumentáció HTTP GET API leírásában (1. és 2. fejezet) található.

Ha egy műveletet a megjelölt metódustól eltérő módon próbáluk megszólítani, akkor az API válasza:

```
405 Method Not Allowed  
(Nem megengedett HTTP metódus)
```

7.3 REST API hibakódok

A hibaüzeneteket célszerű az API-t megszólító alkalmazásban üzenetenként is naplózni (pl. a hibakódokat a kiküldött üzeneteket tartalmazó adatbázistábla egy szöveges mezőjébe gyűjteni, a legutolsó hibakódot pedig egy szám típusú mezőben tárolni), mivel így esetleges hibát követően újra könnyen le tudjuk válogatni az újraküldendő üzeneteket.

Ezentúl érdemes lehet minden **ERR** típusú eredménynél e-mailben vagy egyéb módon azonnali értesítést beállítani, ami segíthet a mielőbbi hibaelhárításban.

Az alábbi táblázatokban a HTTP oszlop jelzi a válaszfejlécben visszaadott HTTP státuszkódot. A HTTP státuszkódok általánosak, nem tudják pontosan azonosítani az összes hibatípust, ami a rendszerünkben jelentkezik, ezért láthatóak itt a result, code és message oszlopok is, amik már a BIP saját hibakódjait és szövegelemeit tartalmazzák, és a válasz (payload) részeként érkeznek.

Megjegyzés: az alábbi táblázatok megegyeznek az HTTP GET API hibáit tartalmazó táblázatokkal (ld. 3-as fejezet), kivéve a csak itt megjelenő HTTP státuszkód oszlopot és a „REST negotiation (kapcsolatfelvétel) hibák” táblázatát, amik szintén csak ennél az API változatnál értelmezhetőek.

Sikeres művelet

HTTP	result	code	message
200	OK	0	A kérést sikeresen teljesítettük

REST negotiation (kapcsolatfelvétel) hibák

HTTP	result	code	message
405	ERR	405	Nem megengedett HTTP metódus
406	ERR	406	Nem elfogadható Content-Type az Accept fejlécben
415	ERR	415	A hívás Content-Type-ja nem támogatott

Paraméterezési hibák (formátumhibák)

HTTP	result	code	message
422	ERR	100	Kötelező paraméter hiányzik: paraméternév
422	ERR	101	A paraméternév paraméterben csak számok szerepelhetnek.
422	ERR	102	A telefonszám nem támogatott vagy nem nemzetközi formátumban van.
422	ERR	103	Az üzenet hosszabb a maximális 459 karakternél.
404	ERR	104	A megadott feladóazonosító (senderid) nem engedélyezett
422	ERR	105	Az üzenet szövege tiltott karaktereket tartalmaz
422	ERR	106	A callback paraméter hibás
422	ERR	107	A megadott időzítési időpont hibás
404	ERR	108	A megadott referenciaid-vel függőben lévő üzenet nem található (<i>cancelsms műveletnél</i>)
422	ERR	109	Ismeretlen paraméter: " paraméternév "
404	ERR	110	Ismeretlen művelet (target érték): " művelet "
422	ERR	111	A "format" paraméter értéke érvénytelen: " érték "

Művelet végrehajtás közben történt hibák (hozzáférési, egyenleg vagy egyéb hibák)

HTTP	result	code	message
422	ERR	301	A(z) <code>ipcím</code> IP címről nem adható fel üzenet. Engedélyezze a címet a Beállítások menüpontban!
422	ERR	302	Jelenlegi egyenlege (<code>jelenlegi egyenleg</code>) nem elegendő az üzenet kiküldéséhez (<code>üzenet díja</code>)
422	ERR	303	Az Ön hitelkerete nem teszi lehetővé az üzenet kiküldését
422	ERR	304	Egyenlege lejárt, vagy egyenlegének beállítása hiányos.
403	ERR	305	Az API kulcs hibás, az API hozzáférés nincs bekapcsolva, vagy hozzáférés letiltva.
422	ERR	310	A token lejárt. (<i>tfavalidate műveletnél</i>)
422	ERR	311	A megadott token nem egyezik meg a kiküldött tokennel. (<i>tfavalidate műveletnél</i>)
404	ERR	312	A megadott számhoz és appid-hez nem találtunk tokent, vagy már lejárt. (<i>tfavalidate műveletnél</i>)
422	ERR	313	Túl sok token validálási kérés egy percen belül. (<i>tfavalidate műveletnél</i>)
422	ERR	315	A címzett ebben a fiókban leiratkozott az SMS fogadásról (<code>telefonszám</code>).

Szolgáltatás oldali hiba

HTTP	result	code	message
500	ERR	500	A rendszer átmeneti hibája miatt a szolgáltatás jelenleg nem elérhető. Munkatársaink értesítést kaptak, és megkezdik a probléma elhárítását

8 SMS fogadás

A BIP nemcsak SMS küldésre, de tetszőleges interaktív SMS kommunikációra is tökéletesen alkalmas. Az SMS interakciók megvalósítása több kombinációkban is lehetséges:

- Ha a folyamatot már pontosan ismeri, a műszaki megvalósítást pedig rábízna, vegye fel kapcsolatot ügyfélszolgálatunkkal, ahol a konkrét igény alapján ajánlatot adunk a beállítások, adott esetben fejlesztések elvégzéséhez (elérhetőségeinket a dokumentáció címlapján és weboldalunkon is megtalálja).
- Ha rendelkezik a megfelelő ismeretekkel, vagy saját programozóját bízta meg a feladattal, a szolgáltatás igénylését ügyfélszolgálatunkon kell kezdeményeznie, azonban a működtetés logikáját saját programozója fogja megvalósítani.

Hogyan működik?

- A BIP rendszerében az Ön számára az igényektől függően fenntartunk egy telefonszámot vagy egy telefonszámot és egy kulcsszót (a kulcsszóval kiegészített telefonszám olcsóbban elérhető, azonban az SMS-eket a feladónak mindig a megadott kulcsszóval kell kezdenie, hogy az üzenet elérje Önt).
- A BIP SMS fogadás szolgáltatása nem egy egyszerű mobilkészülékbe helyezett SIM kártyát jelent: a BIP a szolgáltatók SMS központjaival közvetlen és állandó kapcsolatban van, és azokhoz a mobil antennahálózat kizárásával csatlakozik. Ez a legmagasabb műszaki színvonalat jelenti, mivel:
 - nem függ télerőtől és/vagy hálózati problémáktól (melyek egyébként rendszeresek, a mobilszolgáltatók mindennapjainak természetes részei): a BIP-hez közvetlenül a szolgáltatók gerinchálózatából érkeznek az üzenetek közvetlen, folyamatos adatkapcsolaton keresztül.
 - a szolgáltatás nincs kitéve a mobilkészülékekre vagy egyéb GSM hardverekre jellemző instabilitásnak (időnkénti lefagyások, télerővesztés)
 - nagyságrendileg nagyobb kapacitású, mint a szolgáltatók SIM alapú, SMS-enként rendszerint 1-3 másodpercet igénylő szolgáltatásai. A BIP SMS fogadás segítségével pl. országos TV-műsor szavazások is gond nélkül lebonyolíthatóak, ahol rövid idő alatt nagy mennyiségű üzenet érkezik be.

Az SMS fogadás módjai

- **A legegyszerűbb** SMS fogadási mód az üzenetek fogadása a BIP webes felületén. Ekkor a beérkező üzeneteket a BIP webes felületén tekintheti meg, ahol tetszőleges mappákba válogathatja szét a beérkező üzeneteket – így pl. SMS alapú visszajelzések ügyfélszolgálati feldolgozására is kiválóan alkalmas. A webes felületen minden egyes beérkező üzenet elérhető marad, így a fogadási folyamat bármikor könnyen ellenőrizhető.
- **A „nyers” üzenetek átadása** igénytől függően automatikusan (pl. e-mailben, HTTP GET hívással) is megtörténhet: ekkor Önnek kell gondoskodnia az üzenetek kezeléséről (pl. az SMS tartalmának tárolása saját rendszerében, SMS tartalmának értelmezése, válasz küldése).
- **Ha részben vagy egészében rák bízna** az üzenetek feldolgozását, a BIP számos esetben már kész megoldásokat nyújt. Az SMS feldolgozási folyamat során rendszerünk egyrészt a kért mód(ok)on ellenőrizheti az üzeneteket, másrészt különböző művelet(ek)et hajthat rajtuk végre:
 - az ellenőrzések a bejövő üzenet tartalmi ellenőrzését jelentik, például: megfelel-e egy adott kötelező formátumnak az üzenet szövege, tartalmaz-e az üzenet bizonyos kifejezés(ek)et, vagy a tartalma szerepel-e egy előre meghatározott listában (SMS kuponbeküldő promóciók esetén gyakori igény)
 - az üzenetekkel egy vagy több művelet is elvégezhető:
 - üzenet elhelyezése megadott mappában (a BIP webes felületén elérhető)
 - bejövő üzenet elküldése e-mailben egy vagy több Ön által megadott címre
 - válasz küldése
 - üzenet átadása HTTP GET metódus segítségével
 - feladó számának eltárolása a BIP webes címjegyzékében
 - tetszőleges egyéb feldolgozás
- Bármelyik kombinációt választja az SMS fogadáshoz, a BIP ügyfélszolgálatán kell jeleznie igényét.

A fenti lehetőségekkel akár összetett feldolgozás is megvalósítható, hiszen egyetlen üzenet feldolgozása során több ellenőrzés és művelet is aktiválódhat a kért logika szerint.

8.1 SMS fogadás API (bejövő üzenetek feldolgozásához)

A bejövő SMS-eket az SMS küldésnél is használt HTTP GET metódussal adjuk át (ha egyéb igény nem merül fel). A HTTP GET hívás során minden átadott paramétert UTF-8 kódoláson adunk át – vagyis a bejövő SMS-ek kódlapját egységes kódlapra konvertáljuk.

Az SMS fogadáshoz elő kell készíteni egy olyan, HTTP GET metódussal elérhető programot vagy scriptet, amelyet a BIP ügyfélszolgálat a fogadó telefonszám bekonfigurálásakor kérni fog. Erre a címre adja át a BIP rendszere a bejövő üzeneteket az alábbi táblázat szerinti tartalommal:

Az SMS fogadás paraméterei

🚩 - garantáltan átadott paraméter

number	🚩 A feladó telefonszáma. Minden esetben nemzetközi formátumban, jelek és 00 nélkül, például: 36201234567
message	🚩 Az SMS szövege UTF-8 kódolással. Az üzenet akár 459 karakter hosszúságú is lehet, ezért ha adatbázisban tároljuk az üzenetet, figyeljünk a megfelelő típusú mező megválasztására: MySQL-ben pl. a TEXT típus alkalmas erre a célra.
timestamp	🚩 Az üzenet fogadásának időpontja (ÉÉÉÉ-HH-NN óó:pp:mp formátumban)
keyword	Ha az SMS fogadáshoz kulcsszavas szám szolgáltatást igényelt, akkor ebben a paraméterben a kulcsszót adjuk át – ilyen esetben a <code>message</code> paraméter a kulcsszót nem tartalmazza. Például ha az igényelt kulcsszó JATEK volt, akkor a „JATEK Kiss Ilona, 1111 Budapest, Széles utca 12.” üzenet esetén a <code>keyword</code> értéke JATEK lesz, a <code>message</code> értéke pedig Kiss Ilona, 1111 Budapest, Széles utca 12.

FONTOS! A BIP rendszere az üzenet átadását akkor tekinti sikeresnek, ha a fogadó URL HTTP 200 státuszt ad vissza. Ha nem tud csatlakozni a szerverhez, vagy nem ezt a státuszt kapja vissza, akkor hálózati vagy fogadó oldali hibát feltételezve további maximum 4 alkalommal próbálja átadni az üzenetet.

Ha szükséges, a BIP rendszere más paraméterezéssel – pl. eltérő paraméternevekkel vagy kódlappal - is át tudja adni a beérkező üzeneteket, ezt értelemszerűen a szolgáltatás igénylésekor kell jeleznie ügyfélszolgálatunkon.

A BIP-től kiadott HTTP hívás aszinkron módon működik, azaz az általunk kiadott HTTP hívást megelőzően az üzenet már célbaértnek minősül. Ennek oka a stabilitás megőrzése: a rendszerünk állandó szolgáltatási végpontként üzemel az üzenetcserre szempontjából, így ha bármilyen okból a BIP és a szolgáltató közötti kapcsolat megszakadna, akkor az üzenetek sem tudnának célba érni amíg a kapcsolat újra fel nem épül (így a mobil protokollokban elvárt módon működik a szolgáltatás). Ugyanennek a műszaki szintnek a fenntartását viszont nem tudjuk megkövetelni ügyfeleinktől ill. a köztes hálózati útvonal minden elemétől: így mi elfogadjuk és pufferezzük az üzeneteket addig, amíg a fogadó oldal el nem tudja venni. Mivel ez változatos okokból akár órákig is eltarthat (pl. ügyfelünk szervere túlterhelődött, véletlen elkonfigurálás miatt nem működik, stb.), így rendszerünk ezen időszak alatt is garantálja az SMS fogadás működését.

Példák üzenet továbbadására:

1. A beküldött üzenetszöveg: „teszt”
(saját bérelt telefonszám esetén)

<https://example.com/smsfogadas.php?number36201234567&message=teszt×tamp=2019-02-01%2015%3A00%3A00>

ahol a feladó telefonszám **362012345678**, az üzenet szövege **teszt**, a fogadás időpontja pedig **2019-02-01 15:00:00**.

2. A beküldött üzenetszöveg: „JATEK teszt”
(kulcsszavas üzenetfogadás szolgáltatás esetén)

<https://example.com/smsfogadas.php?number=36201234567&message=teszt×tamp=2019-02-01%2015%3A00%3A00&keyword=JATEK>

ahol a feladó telefonszám **362012345678**, a kulcsszó értéke **JATEK**, az üzenet további szövege **teszt**, a fogadás időpontja pedig **2019-02-01 15:00:00**.