

## Bramka SMS:

- *Obsługiwanych ponad 600 sieci w ponad 200 krajach Świata*
- *SMSy z własnym polem nadawcy*
- *Raporty doręczeń*
- *Obsługa długich wiadomości SMS*
- *Obsługa WAP PUSH*
- *Obsługa FAKSÓW*
- *Odbieranie wiadomości SMS*

# Dokumentacja interfejsu HTTP API

Wersja 4.3 [1 stycznia 2011]

## SPIS TREŚCI:

<b>1. Historia zmian w dokumentacji.....</b>	<b>3</b>
<b>2. Wprowadzenie.....</b>	<b>3</b>
2.1. Krok 1 – Rejestracja konta w GSMService.pl.....	4
2.2. Krok 2 – Aktywacja subkonta API .....	4
<b>3. Podstawowe funkcje dostępne w API .....</b>	<b>4</b>
3.1. Wysyłanie wiadomości SMS .....	5
3.2. Wysyłanie wiadomości WAP PUSH .....	6
3.3. Wysyłanie wiadomości FAX.....	7
3.4. Status wysłanych wiadomości.....	8
3.5. Sprawdzanie stanu konta.....	9
3.6. Sprawdzenie kosztu SMS'a .....	10
<b>4. Definiowanie zawartości Pola Nadawcy.....</b>	<b>11</b>
<b>5. Kody statusów API .....</b>	<b>12</b>
<b>6. Odbieranie wiadomości SMS <sup>nowość</sup> .....</b>	<b>13</b>
<b>7. Przykłady prostej implementacji HTTP API w języku PHP .....</b>	<b>13</b>
7.1. Wysyłanie wiadomości SMS .....	13
7.2. Wysyłanie wiadomości WAP PUSH .....	14
7.3. Wysyłanie wiadomości FAX.....	14
7.4. Sprawdzenie statusu wiadomości .....	14
7.5. Sprawdzenie kosztu wiadomości SMS.....	15
7.6. Sprawdzenie stanu konta.....	15
7.7. Skrypt odbierający wiadomość SMS po wywołaniu adresu Callback.....	15
<b>8. Dodatkowe informacje i kontakt .....</b>	<b>16</b>

## SPIS TABEL:

Tab. 1. Objasnienie parametrów modułu do wysyłania wiadomości SMS. ....	5
Tab. 2. Objasnienie parametrów modułu do wysyłania wiadomości WAP PUSH. ....	6
Tab. 3. Objasnienie parametrów modułu do wysyłania wiadomości FAX.....	7
Tab. 4. Objasnienie parametrów modułu do odczytu statusów wysłanych wiadomości.....	8
Tab. 5. Objasnienie parametrów modułu do sprawdzania stanu konta. ....	9
Tab. 6. Objasnienie parametrów modułu do sprawdzenia kosztu wysyłki wiadomości.....	10
Tab. 7. Objasnienie zwracanych kodów statusów.....	12

## 1. Historia zmian w dokumentacji

Wersja	Data	Wprowadzone zmiany
4.3	01.01.2011	Koszt wiadomości jest teraz wyrażony w zł netto (poprzednio w punktach). Poprawiono przykład implementacji skryptu odbierającego wiadomości SMS.
4.2	21.04.2010	Dodano możliwość wysyłki SMSów Ekonomicznych (type=3) oraz opisano usługę odbioru wiadomości SMS na podany adres Callback (wraz z przykładową implementacją w PHP).
4.1	04.03.2010	Uaktualniono dokumentację HTTP API dostosowując ją do zmian w serwisie.
4.0	10.04.2008	Nowa wersja HTTP API, zastępująca całkowicie wersję 3.1, która nie będzie już rozwijana. Od tej pory równolegle funkcjonują wersje HTTP API 3.1 oraz 4.0. Zmiany w wersji 4.0 obejmują możliwość wysyłki FAXów.
3.1	18.03.2008	Dodano możliwość wysyłki wiadomości WAP PUSH.
3.0	15.01.2008	Dostosowano dokumentację interfejsu API do nowej platformy bramki SMS, dodano możliwość wysyłania SMSów ze zdefiniowanym czasem doręczenia wiadomości.

## 2. Wprowadzenie

Dokument ten powstał w celu uproszczenia integracji **Bramki SMS GSMService.pl** z dowolną aplikacją lub systemem poprzez interfejs HTTP API. Obecnie na platformie GSMService.pl istnieje kilka różnych możliwości wysyłania wiadomości SMS:

- Bezpośrednio ze strony <http://bramka.gsmservice.pl>
- Za pośrednictwem interfejsu HTTP API
- Za pośrednictwem interfejsu SOAP API
- Za pośrednictwem usługi MAIL2SMS

W niniejszym dokumencie przedstawiono możliwości, jakie niesie za sobą wykorzystanie interfejsu HTTP API. W chwili obecnej poprzez HTTP API możliwa jest wysyłka wiadomości **SMS**, **Flash SMS**, **WAP PUSH** oraz **FAX**.

Wysyłanie wiadomości SMS/FAX, sprawdzenie jej statusu czy stanu konta poprzez HTTP API polega na wywołaniu odpowiedniego adresu WWW z jednoczesnym przekazaniem do niego pewnych parametrów. Parametry mogą zostać przekazane za pośrednictwem metody **HTTP POST** lub **HTTP GET**. Jednakże ze względów wydajnościowych oraz pewnych ograniczeń metody GET zalecamy wykorzystanie do tego celu metody HTTP POST.

Komunikacja z HTTP API może odbywać się zarówno na porcie **80** (w przypadku nieszyfrowanych połączeń) lub na porcie **443** (z szyfrowaniem **SSL**). Wszystkie wartości przekazywanych parametrów powinny zostać potraktowane uprzednio funkcją **urlencode** (w przypadku PHP), lub jej odpowiednikiem dla danego języka programowania.

**Bardzo prosimy dokładnie zapoznać się z niniejszą dokumentacją przed rozpoczęciem integracji. Ułatwi to cały proces oraz pozwoli uniknąć wielu problemów.**

Aby rozpocząć integrację dowolnej aplikacji z niniejszą Bramką SMS konieczne jest posiadanie konta w serwisie GSMService.pl i aktywacja dostępu do konta API (zwanego w dalszej części tej dokumentacji subkontem API).

Poniżej prezentujemy rok po kroku jak aktywować dostęp do interfejsu HTTP API.

## 2.1. Krok 1 – Rejestracja konta w GSMService.pl

Jeżeli nie posiadasz jeszcze konta w GSMService.pl należy je założyć. W przeciwnym wypadku przejdź do kroku 2. Aby zarejestrować nowe konto w GSMService.pl należy:

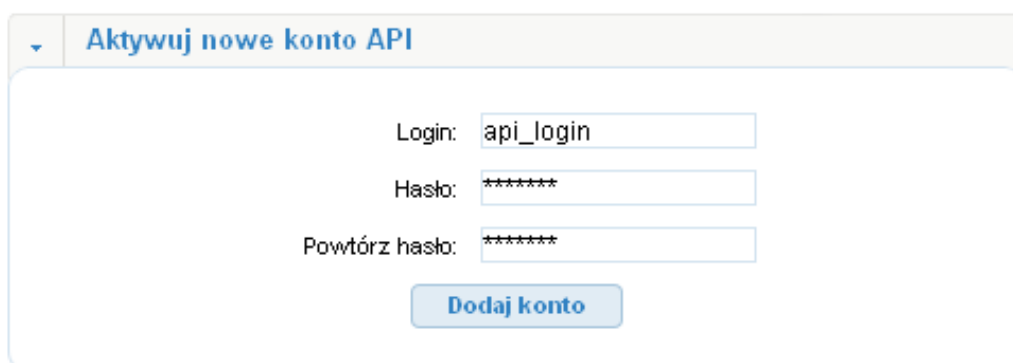
- Wejść na stronę <http://bramka.gsmervice.pl/register.php>,
- Wypełnić formularz rejestracyjny.

Po pomyślnej rejestracji zostaniesz automatycznie zalogowany i poproszony o akceptację Regulaminu Bramki SMS. Będziesz mógł również wówczas wprowadzić dane do **Faktur VAT**, jeżeli chcesz je otrzymywać po każdym zasileniu konta. Po akceptacji Regulaminu konto zostanie utworzone.

## 2.2. Krok 2 – Aktywacja subkonta API

Po zalogowaniu się na <http://bramka.gsmervice.pl> należy:

- Wybrać z lewego MENU, z sekcji „INTEGRACJA (API)” opcję „**Aktywacja kont API**”,
- Wypełnić formularz podając dowolny login oraz hasło dla subkonta API (wpisywany login i hasło posłużą do autentykacji użytkownika przy wywoływaniu wszystkich funkcji HTTP API, dlatego też prosimy o staranne dobranie i bezpieczne przechowywanie loginu oraz hasła, aby nikt niepowołany nie uzyskał dostępu do Państwa subkonta API).



Po pomyślnym wysłaniu formularza subkonto API zostanie założone i aktywowane.

**UWAGA! Obecnie pierwsze subkonto API dla danego konta aktywowane jest bezpłatnie. Za każde kolejne subkonto pobieramy jednorazową opłatę w wysokości 10 zł brutto.**

Posiadając aktywne subkonto API można rozpocząć proces integracji. Aby wysyłać wiadomości SMS, konieczne jest zasilenie konta wybraną kwotą. Szczegóły na <http://bramka.gsmervice.pl/pricing.php>.

W celu zasilenia konta, będąc zalogowanym do Bramki SMS, należy wybrać opcję „**Doładuj konto**” z lewego MENU, a następnie postępować zgodnie ze wskazówkami.

## 3. Podstawowe funkcje dostępne w API

Wydanie polecenia (w tym wysyłanie wiadomości SMS czy FAX) polega na wywołaniu odpowiednio przygotowanego adresu URL. Adres składa się ze ścieżki do skryptu oraz z parametrów przekazywanych do niego metodą POST lub GET.

### 3.1. Wysyłanie wiadomości SMS

W celu wysłania wiadomości SMS poprzez interfejs API należy wywołać następujący adres URL z parametrami w postaci:

[https://api.gsmervice.pl/send.php?login=xxxxx&pass=xxxxx&recipient=xxxxx&text=xxxxxx&type=x&sender=x&deliv\\_time=x](https://api.gsmervice.pl/send.php?login=xxxxx&pass=xxxxx&recipient=xxxxx&text=xxxxxx&type=x&sender=x&deliv_time=x)

gdzie xxx to odpowiednie wartości parametrów. Parametry można przekazywać zarówno metodą POST, jak i GET. Zostały one objaśnione w **Tab. 1**.

**Tab. 1. Objaśnienie parametrów modułu do wysyłania wiadomości SMS.**

Parametr	Opis parametrów
<b>login</b>	Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API]
<b>pass</b>	Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API]
<b>recipient</b>	Numer telefonu odbiorcy, pod który SMS ma zostać wysłany. <b>UWAGA!</b> Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555
<b>text</b>	Treść SMSa, która ma zostać wysłana – powinna zostać odpowiednio zakodowana np. funkcją <b>urlencode</b> (w PHP). Maksymalna długość pojedynczego SMSa – 160 znaków. Maksymalna długość łączonego SMSa: 306 znaków. Uwaga! Niektóre znaki liczone są podwójnie, np. znaki: [ ] ~ ^ { } \   €. Treść powinna być zgodna z kodowaniem <b>ISO-8859-2</b> .
<b>type</b>	Rodzaj wysyłanego SMSa: 1 – SMS tradycyjny (wysyłany z własnym polem nadawcy) 2 – Flash SMS (wysyłany z własnym polem nadawcy) <b>3 – SMS Ekonomiczny (wysyłany z losowego numeru nadawcy) <i>nowość</i></b>
<b>sender</b>	Podpis <sup>1</sup> , który znajdzie się w polu nadawcy zamiast numeru nadawcy. Jeżeli pominięto ten parametr, SMS zostanie wysłany z polem nadawcy „Bramka SMS”. W przypadku, gdy parametr <i>type</i> =3, parametr <i>sender</i> nie jest brany pod uwagę. <b>UWAGA – prosimy przeczytać przypis na końcu tej strony!</b>
<b>deliv_time</b>	Parametr zawiera czas (wyrażony w minutach od chwili wywołania funkcji wysyłającej), kiedy wiadomość ma zostać doręczona do odbiorcy. Wiadomość zostanie doręczona o zdefiniowanej porze (+ - 5 minut). Maksymalna wartość: 10080 (7 dni). Brak tego parametru powoduje, że wiadomość jest doręczana od razu.

Po wywołaniu adresu z prawidłowo przekazanymi parametrami, w treści wyświetlonej strony otrzymamy kod statusu SMSa w formie:

XXX  
MSGID

gdzie:

❖ **XXX** – oznacza odpowiedni kod trzycyfrowy statusu,

<sup>1</sup> Każda zawartość pola nadawcy musi zostać uprzednio zweryfikowana. Oznacza to, że konieczne jest uprzednie zdefiniowanie wszystkich wyrażeń, numerów itd. wykorzystywanych w polu nadawcy. Każdy użytkownik może bezpłatnie zdefiniować nieograniczoną ilość pól nadawcy. Uprzednie zdefiniowanie zawartości oraz weryfikacja ma na celu ograniczenie potencjalnych nadużyć i podnieść bezpieczeństwo wszystkich użytkowników bramki. Opis, w jaki sposób zdefiniować zawartość pola nadawcy oraz pozostałe informacje znajdują się w Rozdziale 4. niniejszej dokumentacji.

- ❖ **MSGID** – unikatowy identyfikator wysłanej wiadomości. Jeżeli kod statusu oznacza błąd, wówczas wartość parametru MSGID nie zostanie zwrócona.

*Posiadając identyfikator wiadomości możliwe jest sprawdzenie aktualnego statusu wysłanej wiadomości. Więcej w rozdziale 3.4 – Status wysłanych wiadomości.*

Objaśnienie poszczególnych kodów statusu przedstawia **Tab. 7**.

#### **Przykład:**

Wywołanie adresu w postaci:

[https://api.gsm.service.pl/send.php?login=user&pass=password&recipient=48601234567&text=To+jest+zakodowana+przez+funkcje+urlencode+tresc+wiadomosci+sms&type=1&sender=Ala&deliv\\_time=30](https://api.gsm.service.pl/send.php?login=user&pass=password&recipient=48601234567&text=To+jest+zakodowana+przez+funkcje+urlencode+tresc+wiadomosci+sms&type=1&sender=Ala&deliv_time=30)

spowoduje wysłanie wiadomości (SMS tradycyjny) o treści „To jest zakodowana przez funkcję urlencode tresc wiadomości sms” pod numer 48601234567, której nadawcą jest Ala. Wiadomość zostanie doręczona za 30 minut. W rezultacie zwrócone zostanie:

**003  
141567**

Zgodnie z **Tab. 7** kod statusu 003 oznacza, że SMS został wysłany, natomiast jego identyfikator (msgid) to 141567.

## 3.2. Wysyłanie wiadomości WAP PUSH

W celu wysłania wiadomości WAP PUSH poprzez interfejs API należy wywołać następujący adres URL z parametrami w postaci:

[https://api.gsm.service.pl/send.php?login=xxxxx&pass=xxxxx&recipient=xxxxx&url=xxxxxx&sender=x&deliv\\_time=x](https://api.gsm.service.pl/send.php?login=xxxxx&pass=xxxxx&recipient=xxxxx&url=xxxxxx&sender=x&deliv_time=x)

gdzie xxx to odpowiednie wartości parametrów. Parametry można przekazywać zarówno metodą POST, jak i GET. Zostały one objaśnione w **Tab. 2**.

**Tab. 2. Objasnienie parametrów modułu do wysyłania wiadomości WAP PUSH.**

Parametr	Opis parametrów
<b>login</b>	Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API]
<b>pass</b>	Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API]
<b>recipient</b>	Numer telefonu odbiorcy, pod który WAP PUSH ma zostać wysłany. UWAGA!!! Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555
<b>url</b>	Parametr ten powinien zawierać adres WWW obiektu, który ma zostać pobrany po otrzymaniu wiadomości. Wartość powinna być zakodowana funkcją <b>urlencode</b> lub podobną.
<b>sender</b>	Podpis <sup>1</sup> , który znajdzie się w polu nadawcy zamiast numeru nadawcy. Jeżeli pominięto ten parametr, WAP PUSH zostanie wysłany z polem nadawcy „Bramka SMS”. <b>UWAGA – prosimy przeczytać przypis na końcu poprzedniej strony!</b>
<b>deliv_time</b>	Parametr zawiera czas (wyrażony w minutach od chwili wywołania funkcji wysyłającej), kiedy wiadomość ma zostać doręczona do odbiorcy. Wiadomość zostanie doręczona o zdefiniowanej porze (+ - 5 minut). Maksymalna wartość: 10080 (7 dni). Brak tego parametru powoduje, że wiadomość jest doręczana od razu.



Po wywołaniu adresu z prawidłowo przekazanymi parametrami, w treści wyświetlonej strony otrzymamy kod statusu wiadomości w następującym formacie:

XXX  
MSGID

gdzie:

- ❖ **XXX** – oznacza odpowiedni kod trzycyfrowy statusu,
- ❖ **MSGID** – identyfikator wysłanej wiadomości. Jeżeli kod statusu oznacza błąd, wówczas wartość parametru MSGID nie zostanie zwrócona.

Objaśnienie poszczególnych kodów statusu przedstawia **Tab. 7**.

#### Przykład:

Wywołanie adresu w postaci:

[https://api.gsmservice.pl/send.php?login=user&pass=password&recipient=48601234567&url=http%3A%2F%2Fwww.gsmservice.pl&sender=Ala&deliv\\_time=15](https://api.gsmservice.pl/send.php?login=user&pass=password&recipient=48601234567&url=http%3A%2F%2Fwww.gsmservice.pl&sender=Ala&deliv_time=15)

spowoduje wysłanie wiadomości WAP PUSH z adresem URL „<http://www.gsmservice.pl>” pod numer 48601234567, której nadawcą jest *Ala*. Wiadomość zostanie doręczona za 15 minut. W rezultacie zwrócone zostanie:

**003**  
**141568**

Zgodnie z **Tab. 7** kod statusu 003 oznacza, że SMS został wysłany, natomiast jego identyfikator (msgid) to 141568.

### 3.3. Wysyłanie wiadomości FAX

Istnieje możliwość wysłania wiadomości faksowych za pośrednictwem interfejsu HTTP API. Wiadomość, która ma zostać wysłana powinna znajdować się w pliku **PDF**.

W celu wysłania wiadomości FAX poprzez interfejs API należy przekazać metodą POST parametry przedstawione w **Tab. 3** na następujący adres URL:

<https://api.gsmservice.pl/sendfax.php>

Parametry w tym przypadku przekazywać można **JEDYNIĘ** metodą **POST**. Zostały one objaśnione w **Tab. 3**.

**Tab. 3. Objaśnienie parametrów modułu do wysyłania wiadomości FAX.**

Parametr	Opis parametrów
<b>login</b>	Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API]
<b>pass</b>	Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API]
<b>recipient</b>	Numer telefonu odbiorcy, pod który FAX ma zostać wysłany. UWAGA!!! Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555. <b>Obecnie obsługiwane są jedynie polskie numery.</b>
<b>file</b>	Zawartość pliku PDF z wiadomością FAX, która ma zostać wysłana. Zawartość pliku musi zostać odpowiednio zakodowana i przesłana jako wartość parametru <b>file</b> metodą <b>POST</b> z <b>Content-type</b> ustawionym na <b>multipart/form-data</b> .

Po wywołaniu adresu z prawidłowo przekazanymi parametrami, w treści wyświetlonej strony otrzymamy kod statusu wysłanego FAKSU w formie:

XXX  
MSGID

gdzie:

- ❖ **XXX** – oznacza odpowiedni trzycyfrowy kod statusu,
- ❖ **MSGID** – identyfikator wysłanej wiadomości. Jeżeli kod statusu oznacza błąd, wówczas wartość parametru MSGID nie zostanie zwrócona.

Objaśnienie poszczególnych kodów statusu przedstawia **Tab. 7**.

#### Przykład:

Ze względu na specyfikę wiadomości FAX oraz konieczność przekazywania parametrów wyłącznie metodą POST, przykład gotowej implementacji w PHP znajduje się w rozdziale 7.3.

### 3.4. Status wysłanych wiadomości

Istnieje możliwość sprawdzenia aktualnego statusu wysyłanych wiadomości SMS, WAP PUSH oraz FAX. Statusy wiadomości zmieniają się np. w momencie doręczenia wiadomości do odbiorcy, itp.

W celu sprawdzenia aktualnego statusu wiadomości należy wywołać adres w następującej formie (parametry można przekazać metodą POST lub GET):

<https://api.gsmservice.pl/getstatus.php?login=xxxxx&pass=xxxxxx&msgid=xxxx>

gdzie xxx to odpowiednie wartości parametrów, które zostały objaśnione w **Tab. 4**.

**Tab. 4. Objasnienie parametrów modułu do odczytu statusów wysłanych wiadomości.**

Parametr	Opis parametrów
<b>login</b>	Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API]
<b>pass</b>	Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API]
<b>msgid</b>	Identyfikator wiadomości otrzymany po wysłaniu wiadomości

Po wywołaniu odpowiednio spreparowanego adresu w treści wyświetlonej strony otrzymamy wynik w następującej postaci:

XXX  
MSGID|NR\_DOCELOWY|DATA\_GODZINA\_1|DATA\_GODZINA\_2|STATUS|RODZAJ|KOSZT

gdzie:

- ❖ **XXX** – kod statusu żądania, zgodny z **Tab. 7**.
- ❖ **MSGID** – identyfikator wiadomości w naszym systemie
- ❖ **NR\_DOCELOWY** – numer telefonu odbiorcy wiadomości
- ❖ **DATA\_GODZINA\_1** – data i godzina wysłania wiadomości
- ❖ **DATA\_GODZINA\_2** – data i godzina ostatniej zmiany statusu wiadomości
- ❖ **STATUS** – aktualny kod statusu wiadomości w formie trzycyfrowej zgodnie z **Tab. 7**
- ❖ **RODZAJ** – rodzaj wiadomości (1 – SMS tradycyjny, 2 – Flash SMS, 3 – SMS ekonomiczny,



4 – Logo czarno-białe, 5 – Tapeta, 6 – Logo kolorowe, 7 – wiadomość FAKS, 8 – WAP PUSH, 99 – SMS aktywacyjny)

❖ **KOSZT** – koszt wysyłki wiadomości (wyrażony w zł netto)

W przypadku błędnego zapytania zostanie wyświetlony jedynie kod statusu żądania.

#### Przykład:

Wywołanie adresu w postaci:

<https://api.gsmservice.pl/getstatus.php?login=user&pass=password&msgid=141568>

spowoduje wysłanie żądania sprawdzenia statusu wiadomości, której identyfikator (msgid) to 141568. W rezultacie zwrócone zostanie:

**300**

**141568|48601234567|2008-04-01 22:34:23|2008-04-01 22:35:54|013|1|24**

Zgodnie z **Tab. 7** kod statusu 300 oznacza, że żądanie było prawidłowe, dlatego w kolejnej linii system zwrócił, że wiadomość z msgid równym 141568, wysłana pod numer **48601234567** w dniu 2008-04-01 o godz. 22:34:23 została doręczona do odbiorcy (kod statusu równy 013) w dniu 2008-04-01 o godz. 22:35:54. Była to wiadomość typu SMS tradycyjny (rodzaj 1), a jej koszt to 24 pkt.

### 3.5. Sprawdzanie stanu konta

Aby sprawdzić aktualny stan konta w Bramce SMS za pośrednictwem interfejsu API należy wywołać adres o następującej składni:

<https://api.gsmservice.pl/getbalance.php?login=xxxxx&pass=xxxxxx>

gdzie xxx to odpowiednie wartości parametrów. Znaczenie każdego parametru przedstawia Tab. 5.

**Tab. 5. Objasnienie parametrów modułu do sprawdzania stanu konta.**

Parametr	Opis parametrów
<b>login</b>	Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API]
<b>pass</b>	Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API]

Po wywołaniu prawidłowego adresu z przekazanymi parametrami, w treści wyświetlonej strony otrzymamy aktualny stan konta w postaci:

**XXX  
LOGIN|YYY**

gdzie:

- ❖ **XXX** – kod statusu żądania, zgodny z **Tab. 7**
- ❖ **LOGIN** – Login konta, którego sprawdzamy stan konta
- ❖ **YYY** – oznacza stan konta wyrażony w zł netto.

W przypadku błędnego żądania wyświetlony zostanie jedynie kod statusu żądania.

**Przykład:**

Wywołanie adresu w postaci:

<https://api.gsmservice.pl/getbalance.php?login=user&pass=password>

spowoduje wysłanie żądania sprawdzenia stanu konta, którego subkonto API to „user”. W rezultacie zwrócone zostanie:

**300**  
**user|123**

Zgodnie z **Tab. 7** kod statusu 300 oznacza, że żądanie było prawidłowe, dlatego w kolejnej linii system zwrócił informację, że na koncie „user” znajduje się jeszcze 123 pkt.

### 3.6. Sprawdzenie kosztu SMS'a

Koszt wiadomości uzależniony jest od sieci docelowej odbiorcy i może być okresowo zmieniany przez poszczególnych operatorów GSM. Aby sprawdzić **aktualny koszt** wiadomości, wysyłanej pod konkretny numer, należy wywołać adres o następującej postaci:

<https://api.gsmservice.pl/getprice.php?recipient=xxxxxxxxxxx&text=xxxxxxxxx&type=x>

gdzie xxx to odpowiednie wartości parametrów. Parametry można przekazać za pomocą metody POST lub GET. Objaśnienia parametrów przedstawia **Tab. 6**.

**Tab. 6. Objaśnienie parametrów modułu do sprawdzenia kosztu wysyłki wiadomości.**

Parametr	Opis parametrów
<b>recipient</b>	Numer telefonu odbiorcy, pod który ma zostać wysłana wiadomość. UWAGA!!! Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555.
<b>text</b>	Treść wiadomości, która ma zostać wysłana (treść powinna zostać odpowiednio zakodowana np. funkcją <b>urlencode</b> (w PHP). Obowiązuje kodowanie ISO-8859-2. W przypadku wiadomości WAP PUSH należy wpisać adres URL.
<b>type</b>	Rodzaj wysyłanego SMSa: 1 – SMS tradycyjny 2 – Flash SMS 8 – WAP PUSH

Po wywołaniu adresu z prawidłowymi parametrami, w treści wyświetlonej strony otrzymamy:

**XXX**  
**NR\_DOCELOWY|YY**

gdzie:

- ❖ **XXX** – status żądania. Znaczenie każdego z kodów znaleźć można w **Tab. 7**
- ❖ **NR\_DOCELOWY** – numer odbiorcy, którego koszt wysyłki wiadomości jest sprawdzany
- ❖ **YY** – oznacza koszt wiadomości (wyrażony w zł netto).

W przypadku błędnego żądania wyświetlony zostanie jedynie status żądania.

**Przykład:**

Wywołanie adresu w postaci:

<https://api.gsmservice.pl/getprice.php?login=user&pass=password&recipient=48601234567&text=To+jest+zakodowana+przez+funkcje+urlencode+tresc+wiadomosci+sms&type=2>

spowoduje wysłanie żądania sprawdzenia kosztu wiadomości *Flash SMS* (type=2) o treści „*To jest zakodowana przez funkcje urlencode tresc wiadomości sms*” wysyłanej na numer 48601234567.

W rezultacie zwrócone zostanie:

300  
48601234567|26

Zgodnie z **Tab. 7** kod statusu 300 oznacza, że żądanie było prawidłowe. Wiadomość wysłana pod numer 48601234567 kosztować będzie 26 pkt.

## 4. Definiowanie zawartości Pola Nadawcy

Pole nadawcy jest to numer lub nazwa nadawcy SMSa, która wyświetla się na telefonie odbiorcy w polu „od”. Z uwagi na konieczność zachowania wysokich standardów bezpieczeństwa oraz konieczność ograniczenia potencjalnych nadużyć niezbędne jest uprzednie zdefiniowanie oraz zweryfikowanie zawartości pola nadawcy, które można będzie wykorzystać podczas wysyłania wiadomości SMS.

W tym celu należy zalogować się na konto bramki w serwisie GSMService.pl, a następnie wybrać z lewego menu (sekcja „MENU GŁÓWNE”) opcję „**Edycja pola nadawcy**”.

W formularzu można wpisać zarówno **numer telefonu**, jak i **tekst alfanumeryczny**, który chcesz wykorzystać w polu nadawcy. Tekst alfanumeryczny może zawierać maksymalnie 11 znaków, w tym małe i duże litery od A do Z, cyfry od 1 do 9 oraz znak spacji. Pozostałe znaki nie są akceptowane.

Tekst alfanumeryczny po zweryfikowaniu przez operatora Bramki SMS zostanie przypisany do Twojego konta i będziesz mógł go używać **w polu nadawcy** podczas wysyłania wiadomości SMS. W przypadku, gdy wprowadzisz do formularza numer telefonu, otrzymasz na ekranie wskazówki, w jaki sposób go zweryfikować.

Każdy użytkownik Bramki SMS może zdefiniować **dowolną ilość** pól nadawcy. W przypadku, gdy wysyłając wiadomość SMS użyjesz niezdefiniowane uprzednio pole nadawcy, wiadomość zostanie wysłana z polem nadawcy ustawionym domyślnie na „**Bramka SMS**”.

## 5. Kody statusów API

Opisane w niniejszej dokumentacji funkcje zwracają w rezultacie swojej pracy trzycyfrowe kody statusu, które świadczą o przebiegu danej operacji. W **Tab. 7** znajduje się ich spis wraz z niezbędnymi objaśnieniami.

**Tab. 7. Objaśnienie zwracanych kodów statusów.**

Parametr	Opis parametrów
<b>000</b>	Brak wystarczających środków na koncie w Bramce SMS
<b>001</b>	Błąd techniczny w składni SMSa
<b>002</b>	SMS nie został wysłany
<b>003</b>	SMS został wysłany
<b>004</b>	Sieć, do której próbowano wysłać SMSa jest obecnie niedostępna
<b>005</b>	Numer odbiorcy jest nieprawidłowy
<b>006</b>	Brak treści SMSa
<b>007</b>	Problem z doręczeniem SMSa (abonent ma wyłączony telefon, jest poza zasięgiem sieci, jego numer został przeniesiony lub jego skrzynka odbiorcza jest przepełniona itp.)
<b>008</b>	Nieprawidłowy Login lub/i hasło do subkonta API
<b>009</b>	Nieprawidłowy rodzaj SMSa
<b>010</b>	SMS oczekuje w kolejce na wysłanie
<b>011</b>	Trwa oczekiwanie na zwrócenie statusu SMSa przez sieć GSM
<b>012</b>	Wiadomość anulowana przez użytkownika
<b>013</b>	SMS został doręczony do odbiorcy (otrzymano raport doręczenia). UWAGA! Status ten aktywny jest w nie wszystkich docelowych sieciach. Sieci, które nie obsługują raportów doręczeń zwrócą status 003
<b>200</b>	FAX został przekazany do operatora.
<b>201</b>	FAX został wysłany do odbiorcy.
<b>202</b>	Linia odbiorcy była zajęta. FAX nie został wysłany.
<b>203</b>	Brak sygnału faksu. Prawdopodobnie pod podanym numerem nie pracuje fax.
<b>204</b>	Nie udało się odebrać informacji o rezultacie wysyłania faksu.
<b>205</b>	Połączenie nie zostało odebrane.
<b>206</b>	Zbyt niska jakość połączenia ze strony odbiorcy nie pozwala na wysyłkę faksu pod ten numer. Faks wysyłający nie rozpoczął transmisji obrazu.
<b>207</b>	Faks został częściowo wysłany. Przetransmitowano tylko część stron gdyż połączenie zostało przerwane.
<b>208</b>	Błąd protokołu. FAX nie został wysłany.
<b>209</b>	Faks wysyłający i odbierający nie uzgodniły opcji transmisji.
<b>210</b>	Wiadomość jest w trakcie wysłania.
<b>211</b>	Wystąpił nieoczekiwany błąd. Wiadomość nie została wysłana.
<b>212</b>	Wiadomość jest w trakcie wysłania.
<b>213</b>	Nieprawidłowy plik PDF
<b>300</b>	Żądanie wykonane poprawnie
<b>301</b>	Błąd w żądaniu

## 6. Odbieranie wiadomości SMS **nowość**

Udostępniamy usługę umożliwiającą odbieranie wiadomości SMS na przydzielony Państwu numer dedykowany lub numer z dedykowanym prefiksem. Szczegółowy opis poszczególnych wariantów usługi zawarty został na stronie <http://bramka.gsmervice.pl/mo.php>. Odebrane wiadomości mogą zostać przekazane na podany w czasie aktywacji usługi adres **Callback**. Adres ten można także skonfigurować w dowolnej chwili na stronie konfiguracji usługi odbioru wiadomości SMS. Po odebraniu wiadomości SMS kierowanej do Państwa nasza platforma wywoła adres Callback przekazując do niego metodą GET następujące parametry:

- **msg\_id** - Unikalny identyfikator odebranej wiadomości
- **sender** - Numer telefonu nadawcy wiadomości
- **recipient** - Numer telefonu odbiorcy wiadomości
- **text** - Treść odebranej wiadomości SMS (zakodowanej funkcją urlencode)
- **date** - Data odbioru wiadomości SMS w formacie RRRR-MM-DD GG:MM:SS (zakodowana funkcją urlencode)

Umożliwi to Państwu odebranie wiadomości SMS bezpośrednio w Państwa serwisach, systemach i aplikacjach.

### **Przykład:**

Użytkownik usługi odbioru wiadomości SMS opartej na numerze dedykowanym wskazał adres <http://jakas-domena.pl/skrypt.php> jako adres **Callback**. W momencie aktywacji usługi został przydzielony numer dedykowany +48601234567. W chwili odbierania przez Bramkę wiadomości SMS o treści „To jest wiadomosc testowa” wysłanej spod numeru 48509876543 zostanie wywołany następujący adres: [http://jakas-domena.pl/skrypt.php?msg\\_id=455&sender=48509876543&recipient=48601234567&text=To+jest+wiadomosc+testowa&date=2010-04-21+13%3A22%3A14](http://jakas-domena.pl/skrypt.php?msg_id=455&sender=48509876543&recipient=48601234567&text=To+jest+wiadomosc+testowa&date=2010-04-21+13%3A22%3A14).

Przykładowy skrypt PHP odbierający wiadomość SMS z adresu Callback umieszczony został w rozdziale 7.7.

## 7. Przykłady prostej implementacji HTTP API w języku PHP

### 7.1. Wysyłanie wiadomości SMS

```
<?php

$login = "user";
$pass = "password";
$recipient = "48601234567";
$text = urlencode("to jest tresc wiadomosci sms, ktora zostanie wyslana.");
$type = 1;
$sender = urlencode("Ala");
$deliv_time = 15;

$parametry = "?login=$login&pass=$pass&recipient=$recipient&text=$text";
$parametry .= "&type=$type&sender=$sender&deliv_time=$deliv_time";
$plik = fopen('http://api.gsmervice.pl/send.php' . $parametry, 'r');
$wynik = fread($plik, 1024);
fclose($plik);
echo $wynik;

?>
```

## 7.2. Wysyłanie wiadomości WAP PUSH

```
<?php

$login = "user";
$pass = "password";
$recipient = "48601234567";
$url = urlencode("http://www.gsmervice.pl");
$sender = urlencode("Ala");
$deliv_time = 15;

$parametry = "?login=$login&pass=$pass&recipient=$recipient&url=$url";
$parametry .= "&sender=$sender&deliv_time=$deliv_time";
$plik = fopen('http://api.gsmervice.pl/send.php' . $parametry, 'r');
$wynik = fread($plik, 1024);
fclose($plik);
echo $wynik;
?>
```

## 7.3. Wysyłanie wiadomości FAX

```
<?php

$login = "user";
$pass = "password";
$recipient = "48123456789";
$pdf_file = "plik.pdf"; // Ścieżka do pliku PDF

$hCon = @fsockopen("ssl://api.gsmervice.pl", 443, $errno, $errstr, 15);
if (!$hCon)
    echo "Błąd połączenia";
else
{
    $sBoundary = "-----".substr(md5(rand(0,32000)),0,10);
    $sData .= "--$sBoundary\r\n";
    $sData .= "Content-Disposition: form-data; name=\"file\"; filename=\"\". $pdf_file.\"\"";
    $sData .= "\r\nContent-Type: application/pdf\r\n\r\n";
    $sData .= join("", file($pdf_file))."\r\n";
    $sData .= "--$sBoundary--\r\n";
    $sHeader = "POST /sendfax.php?login=$login&pass=$pass&recipient=$recipient ";
    $sHeader .= "HTTP/1.1\r\nHost: api.gsmervice.pl\r\n";
    $sHeader .= "User-Agent: PHP \r\n";
    $sHeader .= "Content-type: multipart/form-data, boundary=$sBoundary\r\n";
    $sHeader .= "Content-Length: ".strlen($sData)."\r\n";
    $sHeader .= "Connection: Close\r\n\r\n";
    $sHeader .= $sData;
    fputs($hCon, $sHeader);
    fread($hCon, 1400);
    $status = trim(fgets($hCon));
    $msgid = trim(fgets($hCon));
    fclose($hCon);
    echo "Kod statusu: ".$status." msgid: ".$msgid;
}
?>
```

## 7.4. Sprawdzenie statusu wiadomości

```
<?php

$login = "user";
$pass = "password";
$msgid = 141568;

$parametry = "?login=$login&pass=$pass&msgid=$msgid";
$plik = fopen('http://api.gsmervice.pl/getstatus.php' . $parametry, 'r');
$wynik = fread($plik, 1024);
fclose($plik);
echo $wynik;
?>
```



## 7.5. Sprawdzenie kosztu wiadomości SMS

```
<?php

$login = "user";
$pass = "password";
$recipient = "48601234567";
$text = urlencode("to jest tresc wiadomosci sms, ktora zostanie wyslana.");
$type = 1;

$parametry = "?login=$login&pass=$pass&recipient=$recipient&text=$text&type=$type";
$plik = fopen('http://api.gsm.service.pl/getprice.php'.$parametry,'r');
$wynik = fread($plik,1024);
fclose($plik);
echo $wynik;
?>
```

## 7.6. Sprawdzenie stanu konta

```
<?php

$login = "user";
$pass = "password";

$parametry = "?login=$login&pass=$pass ";
$plik = fopen('http://api.gsm.service.pl/getbalance.php'.$parametry,'r');
$wynik = fread($plik,1024);
fclose($plik);
echo $wynik;
?>
```

## 7.7. Skrypt odbierający wiadomość SMS po wywołaniu adresu Callback

```
<?php

$msgId = $_GET["msg_id"];
$sender = $_GET["sender"];
$recipient = $_GET["recipient"];
$text = $_GET["text"];
$date = $_GET["date"];

echo "Odebrano wiadomość SMS z numeru ".$sender." wysłaną pod numer ".$recipient." o treści ".$text.". ID wiadomości: ".$msgId.". Data odebrania wiadomości: ".$date;
?>
```

## 8. Dodatkowe informacje i kontakt

Na specjalne życzenie możemy dostarczyć dodatkowe, gotowe funkcje w **PHP** o znacznie większych możliwościach. **Oferujemy również usługi wdrożeniowe dla użytkowników naszej bramki w atrakcyjnych cenach.**

Odpowiedzi na wszelkie pytania oraz dodatkowe informacje uzyskać można kontaktując się z nami:

### **INTERMEDIA**

www: <http://www.gsmservice.pl>  
e-mail: [bramka@gsmservice.pl](mailto:bramka@gsmservice.pl)  
fax.: +48 13 49 27 060



Aktualne informacje znaleźć można zawsze pod adresem <http://bramka.gsmservice.pl/api.php>.