

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 SOAP API

Wersja 1.3 [01 stycznia 2011]

SPIS TREŚCI:

| | |
|---|-----------|
| 1. Historia zmian w dokumentacji..... | 3 |
| 2. Wprowadzenie..... | 3 |
| 2.1. Krok 1 – Rejestracja konta w GSMService.pl..... | 3 |
| 2.2. Krok 2 – Aktywacja subkonta API | 4 |
| 3. Dostęp do SOAP API | 4 |
| 3.1. Opis metod udostępniane w ramach Webservices SOAP | 5 |
| 3.1.1. Wysyłanie wiadomości SMS (Metoda SendSMS)..... | 5 |
| 3.1.2. Masowa wysyłka wiadomości SMS – Bulk SMS (Metoda SendBulk)..... | 6 |
| 3.1.3. Wysyłanie wiadomości FAX (Metoda SendFAX) | 6 |
| 3.1.4. Sprawdzanie kosztu wiadomości (Metoda GetMsgPrice) | 7 |
| 3.1.5. Sprawdzenie statusu wiadomości (Metoda GetMsgStatus) | 7 |
| 3.1.6. Sprawdzenie stanu konta (Metoda GetAccountBalance)..... | 8 |
| 3.2. Opis struktur obiektów wykorzystanych przy wywołaniu metod | 8 |
| 3.2.1. Struktura GenericReturnObject | 8 |
| 3.2.2. Struktura SendMsgReturnObject..... | 8 |
| 3.2.3. Struktura ArrayOfSendMsgReturnObjects..... | 9 |
| 3.2.4. Struktura SmsMessage | 9 |
| 3.2.5. Struktura ArrayOfSmsMessages | 9 |
| 3.2.6. Struktura SendMsgReturnObject..... | 10 |
| 3.2.7. 'Struktura MsgStatusReturnObject | 10 |
| 4. Definiowanie zawartości Pola Nadawcy..... | 10 |
| 5. Kody statusów API | 11 |
| 6. Odbieranie wiadomości SMS ^{nowość} | 12 |
| 7. Przykłady prostej implementacji SOAP API w języku PHP5 | 12 |
| 7.1. Wysyłanie wiadomości SMS | 12 |
| 7.2. Masowa wysyłka wiadomości BULK SMS..... | 13 |
| 7.3. Wysyłanie wiadomości WAP PUSH | 13 |
| 7.4. Wysyłanie wiadomości FAX..... | 13 |
| 7.5. Sprawdzenie statusu wiadomości | 13 |
| 7.6. Sprawdzenie kosztu wiadomości SMS..... | 14 |
| 7.7. Sprawdzenie stanu konta..... | 14 |
| 7.8. Skrypt odbierający wiadomość SMS po wywołaniu adresu Callback..... | 14 |
| 8. Dodatkowe informacje i kontakt | 15 |

SPIS TABEL:

| | |
|--|----|
| Tab. 1. Objasnienie zwracanych kodów statusów..... | 11 |
|--|----|

1. Historia zmian w dokumentacji

| Wersja | Data | Wprowadzone zmiany |
|--------|------------|--|
| 1.3 | 01.01.2011 | Koszt wiadomości jest teraz wyrażony w zł netto (poprzednio w punktach). Poprawiono przykład implementacji skryptu odbierającego wiadomości SMS. |
| 1.2 | 09.08.2010 | Dodano metodę SendBulk do masowej wysyłki wiadomości poprzez API oraz właściwość <i>recipient</i> do struktury <i>SendMsgReturnObject</i> |
| 1.1 | 21.04.2010 | Dodano możliwość wysyłki SMSów Ekonomicznych oraz opisano usługę odbioru wiadomości SMS na podany adres Callback (wraz z przykładową implementacją w PHP). |
| 1.0 | 04.03.2010 | Pierwsza wersja SOAP API opartego na Webservices |

2. Wprowadzenie

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

- Bezpośrednio ze strony <http://bramka.gsmervice.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 SOAP API. W chwili obecnej poprzez SOAP API możliwa jest wysyłka wiadomości **SMS, Flash SMS, WAP PUSH** oraz **FAX**.

Wysłanie wiadomości SMS/FAX, sprawdzenie jej statusu czy stanu konta poprzez SOAP API polega na wywołaniu zdalnej funkcji Webservices z jednoczesnym przekazaniem pewnych parametrów.

Komunikacja z SOAP API może odbywać się zarówno na porcie **80** (w przypadku nieszyfrowanych połączeń) lub na porcie **443** (z szyfrowaniem **SSL**).

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 krok po kroku jak aktywować dostęp do interfejsu SOAP 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.

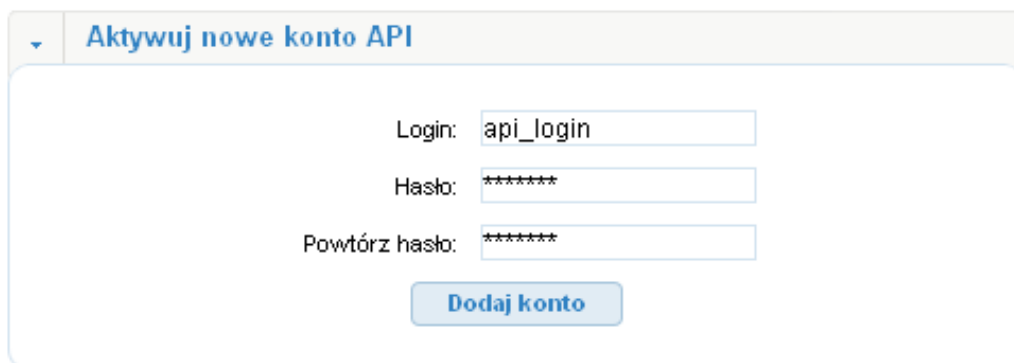
Po pomyślnej rejestracji zostaniesz automatycznie zalogowany i poproszony o akceptację Regulaminu Bramki SMS. Będziesz mógł również wówczas wprowadzić dane do **Faktur VAT**, jeżeli

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.gsm-service.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 SOAP API, dlatego też prosimy o staranne dobranie i bezpieczne przechowywanie loginu oraz hasła, aby nikt niepowołany nie uzyskał dostępu do Państwa subkonta API).



Aktywuj nowe konto API

Login:

Hasło:

Powtórz hasło:

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.gsm-service.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. Dostęp do SOAP API

Udostępniamy możliwość integracji dowolnej aplikacji za pośrednictwem SOAP API.

WebService SOAP dostępny jest pod adresami:

- <http://api.gsm-service.pl/soap/gateway.php>
- <https://api.gsm-service.pl/soap/gateway.php> (połączenie szyfrowane - SSL)

WSDL dostępny jest pod adresami:

- <http://api.gsm-service.pl/soap/gateway.php?wsdl>
- <https://api.gsm-service.pl/soap/gateway.php?wsdl> (połączenie szyfrowane - SSL)

3.1. Opis metod udostępniane w ramach Webservices SOAP

3.1.1. Wysyłanie wiadomości SMS (Metoda SendSMS)

Metoda **SendSMS** umożliwia wysłanie pojedynczej wiadomości SMS. Metoda przyjmuje następujące parametry:

| Metoda | Parametr | Typ parametru | Opis |
|----------------|-----------|---------------|--|
| SendSMS | login | xsd:string | Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API] |
| | password | xsd:string | Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API] |
| | recipient | xsd:string | Numer telefonu odbiorcy, pod który SMS ma zostać wysłany. UWAGA! Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555 |
| | msgText | xsd:string | Treść SMSa, która ma zostać wysłana. 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 UTF-8 . <i>W przypadku wysyłania wiadomości WAP PUSH, parametr ten powinien zawierać adres WWW obiektu, który ma zostać pobrany po otrzymaniu wiadomości.</i> |
| | sender | xsd:string | Podpis ¹ , który znajdzie się w polu nadawcy zamiast numeru nadawcy. Jeżeli pominięto ten parametr, SMS zostanie wysłany z polem nadawcy „Bramka SMS”. UWAGA – prosimy przeczytać przypis na końcu tej strony! |
| | type | xsd:int | Rodzaj wysyłanego SMSa: 1 – SMS tradycyjny 2 – Flash SMS 3 – SMS Ekonomiczny <i>nowość</i> 8 – WAP PUSH |
| | delivTime | xsd:int | 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. |

¹ 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.

Wynik:

| Struktura | Opis |
|----------------------------|---|
| SendMsgReturnObject | Opis struktury znajduje się w Rozdziale 3.2.2 |

3.1.2. Masowa wysyłka wiadomości SMS – Bulk SMS (Metoda SendBulk)

Metoda **SendBulk** umożliwia masowe wysłanie wiadomości SMS. Jednocześnie wysłać można do 2.000 wiadomości SMS. Metoda przyjmuje następujące parametry:

| Metoda | Parametr | Typ parametru | Opis |
|-----------------|-------------|------------------------|---|
| SendBulk | login | xsd:string | Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API] |
| | password | xsd:string | Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API] |
| | smsmessages | tns:ArrayOfSmsMessages | Tablica zawierająca tablice z numerami odbiorców, treścią wiadomości, polami nadawcy, typami wiadomości itd. Przetwarzane będzie pierwsze 2.000 elementów z tablicy. Struktura parametru opisana została w Rozdziale 3.2.5. |

Wynik:

| Struktura | Opis |
|------------------------------------|--|
| ArrayOfSendMsgReturnObjects | Opis struktury znajduje się w Rozdziale 3.2.3. |

3.1.3. Wysyłanie wiadomości FAX (Metoda SendFAX)

Metoda **SendFAX** umożliwia wysłanie pojedynczej wiadomości FAX. Istnieje możliwość wysłania wiadomości faksowych za pośrednictwem interfejsu SOAP API. Wiadomość, która ma zostać wysłana powinna znajdować się w pliku PDF. Metoda przyjmuje następujące parametry:

| Metoda | Parametr | Typ parametru | Opis |
|----------------|-----------|---------------|---|
| SendFAX | login | xsd:string | Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API] |
| | password | xsd:string | Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API] |
| | recipient | xsd:string | Numer telefonu odbiorcy, pod który FAX ma zostać wysłany. UWAGA!!! Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555. Obecnie obsługiwane są jedynie polskie numery. |
| | file | xsd:base64 | Zawartość pliku PDF z wiadomością FAX, która ma zostać wysłana. Zawartość pliku musi zostać odpowiednio zakodowana w standardzie base64 |

Wynik:

| Struktura | Opis |
|----------------------------|--|
| SendMsgReturnObject | Opis struktury znajduje się w Rozdziale 3.2.2. |

3.1.4. Sprawdzanie kosztu wiadomości (Metoda GetMsgPrice)

Metoda **GetMsgPrice** umożliwia sprawdzenie kosztu wiadomości przed jej wysłaniem. Koszt wiadomości uzależniony jest od sieci docelowej odbiorcy i może być okresowo zmieniany przez poszczególnych operatorów GSM. Metoda przyjmuje następujące parametry:

| Metoda | Parametr | Typ parametru | Opis |
|--------------------|-----------|---------------|---|
| GetMsgPrice | login | xsd:string | Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API] |
| | password | xsd:string | Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API] |
| | recipient | xsd:string | Numer telefonu odbiorcy, pod który SMS ma zostać wysłany. UWAGA! Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555 |
| | msgText | xsd:string | Treść SMSa, która ma zostać wysłana. 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 UTF-8. |
| | type | xsd:int | Rodzaj wysyłanego SMSa |

Wynik:

| Struktura | Opis |
|----------------------------|---|
| GenericReturnObject | Opis struktury znajduje się w Rozdziale 3.2.1 |

3.1.5. Sprawdzenie statusu wiadomości (Metoda GetMsgStatus)

Metoda **GetMsgStatus** umożliwia sprawdzenie aktualnego statusu wysyłanych wiadomości SMS, WAP PUSH oraz FAX. Statusy wiadomości zmieniają się np. w momencie doręczenia wiadomości do odbiorcy, itp. Metoda przyjmuje następujące parametry:

| Metoda | Parametr | Typ parametru | Opis |
|---------------------|----------|---------------|--|
| GetMsgStatus | login | xsd:string | Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API] |
| | password | xsd:string | Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API] |

| | | | | |
|--|-------|------------|------------------------------------|--------------|
| | msgId | xsd:string | Identyfikator wiadomości wysyłanej | otrzymany po |
|--|-------|------------|------------------------------------|--------------|

Wynik:

| Struktura | Opis |
|------------------------------|--|
| MsgStatusReturnObject | Opis struktury znajduje się w Rozdziale 3.2.7. |

3.1.6. Sprawdzenie stanu konta (Metoda GetAccountBalance)

Metoda **GetAccountBalance** umożliwia sprawdzenie kosztu wiadomości przed jej wysłaniem. Metoda przyjmuje następujące parametry:

| Metoda | Parametr | Typ parametru | Opis |
|--------------------------|----------|---------------|--|
| GetAccountBalance | login | xsd:string | Login subkonta API w Bramce SMS [zdefiniowany w czasie aktywacji API] |
| | password | xsd:string | Hasło do subkonta API w Bramce SMS [zdefiniowane w czasie aktywacji API] |

Wynik:

| Struktura | Opis |
|----------------------------|---|
| GenericReturnObject | Opis struktury znajduje się w Rozdziale 3.2.1 |

3.2. Opis struktur obiektów wykorzystanych przy wywołaniu metod

3.2.1. Struktura GenericReturnObject

| Pole | Typ | Opis |
|-------------|------------|--|
| status | xsd:string | Status operacji: ERR w przypadku błędu, OK w przypadku powodzenia. |
| statusCode | xsd:string | Trzycyfrowy kod statusu operacji zgodnie z Tab. 1 |
| description | xsd:string | Tekstowy opis statusu operacji |
| value | xsd:double | Zwrócona wartość wywołanej funkcji (np. koszt wiadomości, stan konta). |

3.2.2. Struktura SendMsgReturnObject

| Pole | Typ | Opis |
|------------|------------|--|
| status | xsd:string | Status operacji: ERR w przypadku błędu, OK w przypadku powodzenia. |
| statusCode | xsd:string | Trzycyfrowy kod statusu operacji zgodnie z Tab. 1 |

| | | |
|-------------|------------|--|
| description | xsd:string | Tekstowy opis statusu operacji |
| recipient | xsd:string | Numer telefonu odbiorcy wiadomości |
| msgId | xsd:string | Unikalny identyfikator wysłanej wiadomości |

3.2.3. Struktura ArrayOfSendMsgReturnObjects

| Pole | Typ | Opis |
|-------|---------------------------|--|
| Array | tns:SendMsgReturnObject[] | Tablica zawierająca obiekty struktury SendMsgReturnObject (przy wysyłce masowej Bulk SMS każdy z obiektów SendMsgReturnObject zawiera informacje o wysłanej pojedynczej wiadomości SMS). |

3.2.4. Struktura SmsMessage

| Pole | Typ | Opis |
|-----------|------------|--|
| recipient | xsd:string | Numer telefonu odbiorcy, pod który SMS ma zostać wysłany. UWAGA! Koniecznie w postaci międzynarodowej z kodem kraju na początku (bez znaku +) np. 48601444555 |
| msgText | xsd:string | Treść SMSa, która ma zostać wysłana. 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 UTF-8 . <i>W przypadku wysyłania wiadomości WAP PUSH, parametr ten powinien zawierać adres WWW obiektu, który ma zostać pobrany po otrzymaniu wiadomości.</i> |
| sender | xsd:string | Podpis ¹ , który znajdzie się w polu nadawcy zamiast numeru nadawcy. UWAGA – prosimy przeczytać przypis na końcu strony nr 5! |
| type | xsd:int | Rodzaj wysłanego SMSa: 1 – SMS tradycyjny 2 – Flash SMS 3 – SMS Ekonomiczny nowość 8 – WAP PUSH |
| delivTime | xsd:int | 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). Wartość 0 powoduje, że wiadomość jest doręczana od razu. |

3.2.5. Struktura ArrayOfSmsMessages

| Pole | Typ | Opis |
|-------|------------------|--|
| Array | tns:SmsMessage[] | Tablica zawierająca obiekty struktury SmsMessage (przy wysyłce masowej Bulk SMS każdy z obiektów SmsMessage zawiera parametry pojedynczej wiadomości SMS). |

3.2.6. Struktura SendMsgReturnObject

| Pole | Typ | Opis |
|-------------|------------|--|
| status | xsd:string | Status operacji: ERR w przypadku błędu, OK w przypadku powodzenia. |
| statusCode | xsd:string | Trzycyfrowy kod statusu operacji zgodnie z Tab. 1 |
| description | xsd:string | Tekstowy opis statusu operacji |
| recipient | xsd:string | Numer telefonu odbiorcy wiadomości |
| msgId | xsd:string | Unikalny identyfikator wysłanej wiadomości |

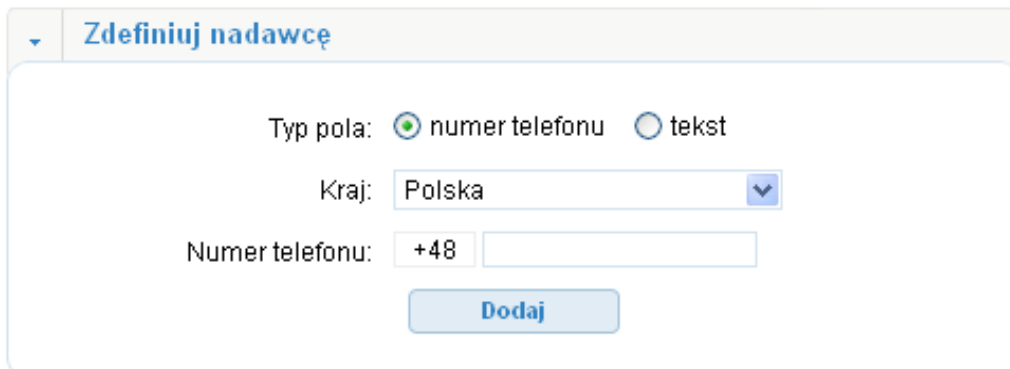
3.2.7. 'Struktura MsgStatusReturnObject

| Pole | Typ | Opis |
|----------------------|------------|--|
| status | xsd:string | Status operacji: ERR w przypadku błędu, OK w przypadku powodzenia. |
| statusCode | xsd:string | Trzycyfrowy kod statusu operacji zgodnie z Tab. 1 |
| description | xsd:string | Tekstowy opis statusu operacji |
| msgId | xsd:string | Unikalny identyfikator wiadomości |
| msgRecipient | xsd:string | Numer telefonu odbiorcy wiadomości |
| msgDate | xsd:string | Data wysłania wiadomości |
| msgStatusDate | xsd:string | Data ostatniej zmiany statusu wiadomości |
| msgStatusCode | xsd:string | Trzycyfrowy kod statusu wysłanej wiadomości zgodnie z Tab. 1 |
| msgStatusDescription | xsd:string | Tekstowy opis aktualnego statusu wiadomości |
| msgType | xsd:int | 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) |
| msgPrice | xsd:double | Koszt wysłanej wiadomości (wyrażony w zł netto) |

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**”.



Zdefiniuj nadawcę

Typ pola: numer telefonu tekst

Kraj: Polska

Numer telefonu: +48

Dodaj

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. 1** znajduje się ich spis wraz z niezbędnymi objaśnieniami.

Tab. 1. Objaśnienie zwracanych kodów statusów.

| Parametr | Opis parametru |
|----------|---|
| 000 | Brak wystarczających środków na koncie w Bramce SMS |
| 001 | Błąd techniczny w składni SMSa |
| 002 | SMS nie został wysłany |
| 003 | SMS został wysłany |
| 004 | Sieć, do której próbowano wysłać SMSa jest obecnie niedostępna |
| 005 | Numer odbiorcy jest nieprawidłowy |
| 006 | Brak treści SMSa |
| 007 | Problem z doręczeniem SMSa (abonent ma wyłączony telefon, jest poza zasięgiem sieci, jego numer został przeniesiony lub jego skrzynka odbiorcza jest przepelniona itp.) |
| 008 | Nieprawidłowy Login lub/i hasło do subkonta API |
| 009 | Nieprawidłowy rodzaj SMSa |
| 010 | SMS oczekuje w kolejce na wysłanie |
| 011 | Trwa oczekiwanie na zwrócenie statusu SMSa przez sieć GSM |
| 012 | Wiadomość anulowana przez użytkownika |
| 013 | 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 |
| 200 | FAX został przekazany do operatora. |
| 201 | FAX został wysłany do odbiorcy. |
| 202 | Linia odbiorcy była zajęta. FAX nie został wysłany. |
| 203 | Brak sygnału faksu. Prawdopodobnie pod podanym numerem nie pracuje fax. |
| 204 | Nie udało się odebrać informacji o rezultacie wysyłania faksu. |
| 205 | Połączenie nie zostało odebrane. |
| 206 | Zbyt niska jakość połączenia ze strony odbiorcy nie pozwala na wysyłkę faksu pod ten numer. Faks wysyłający nie rozpoczął transmisji obrazu. |
| 207 | Faks został częściowo wysłany. Przetransmitowano tylko część stron gdyż połączenie zostało przerwane. |
| 208 | Błąd protokołu. FAX nie został wysłany. |
| 209 | Faks wysyłający i odbierający nie uzgodniły opcji transmisji. |

| | |
|-----|---|
| 210 | Wiadomość jest w trakcie wysłania. |
| 211 | Wystąpił nieoczekiwany błąd. Wiadomość nie została wysłana. |
| 212 | Wiadomość jest w trakcie wysłania. |
| 213 | Nieprawidłowy plik PDF |
| 300 | Żądanie wykonane poprawnie |
| 301 | 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.

Przykładowy skrypt PHP odbierający wiadomość SMS z adresu Callback umieszczony został w rozdziale 7.8.

7. Przykłady prostej implementacji SOAP API w języku PHP5

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;

$client = new SoapClient("https://api.gsmervice.pl/soap/gateway.php?wsdl");

$out = $client->SendSMS($login, $pass, $recipient, $text, $sender, $type, $deliv_time);

print_r($out);
?>
```

7.2. Masowa wysyłka wiadomości BULK SMS

```
<?php
$login = "user";
$pass = "password";
$messages = array(
    array("recipient" => "48012345678", "msgText" => "tresc 1", "sender" => "Ania",
"type" => "1", "delivTime" => "0"),
    array("recipient" => "48987654321", "msgText" => "tresc 2", "sender" => "Test",
"type" => "1", "delivTime" => "0"),
    array("recipient" => "48990099009", "msgText" => "tresc 3", "sender" => "Bramka",
"type" => "1", "delivTime" => "0")
);

$client = new SoapClient("https://api.gsm.service.pl/soap/gateway.php?wsdl");
$out = $client->SendBulk($login, $pass, $messages);
print_r($out);
?>
```

7.3. Wysyłanie wiadomości WAP PUSH

```
<?php
$login = "user";
$pass = "password";
$recipient = "48601234567";
$url = urlencode("http://www.gsm.service.pl");
$type = 8;
$sender = urlencode("Ala");
$deliv_time = 15;

$client = new SoapClient("https://api.gsm.service.pl/soap/gateway.php?wsdl");

$out = $client->SendSMS($login, $pass, $recipient, $url, $sender, $type, $deliv_time);

print_r($out);
?>
```

7.4. Wysyłanie wiadomości FAX

```
<?php
$login = "user";
$pass = "password";
$recipient = "48123456789";
$pdf_file = file_get_contents("plik.pdf"); // Ścieżka do pliku PDF

$client = new SoapClient("https://api.gsm.service.pl/soap/gateway.php?wsdl");
$out = $client->SendFAX($login, $pass, $recipient, base64_encode($pdf_file));

print_r($out);
?>
```

7.5. Sprawdzenie statusu wiadomości

```
<?php
$login = "user";
$pass = "password";
$msgid = 141568;

$client = new SoapClient("https://api.gsm.service.pl/soap/gateway.php?wsdl");
$out = $client->GetMsgStatus($login, $pass, $msgid);

print_r($out);
?>
```

7.6. Sprawdzenie kosztu wiadomości SMS

```
<?php

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

$client = new SoapClient("https://api.gsm.service.pl/soap/gateway.php?wsdl");
$out = $client->GetMsgPrice($login, $pass, $recipient, $text, $type);

print_r($out);
?>
```

7.7. Sprawdzenie stanu konta

```
<?php

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

$client = new SoapClient("https://api.gsm.service.pl/soap/gateway.php?wsdl");
$out = $client->GetAccountBalance($login, $pass);

print_r($out);
?>
```

7.8. 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.gsmervice.pl>
e-mail: bramka@gsmervice.pl
fax.: +48 13 49 27 060



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