

SMSAPI

Interfejs HTTP/S API
wersja 2.20



SPIS TREŚCI

Historia zmian

1. Wprowadzenie

1.1. Rozpoczęcie współpracy

1.2. Filtr IP dla interfejsu API

1.3. Hasło do interfejsu API

I. WYSYŁKA SMS

2. Wysyłanie pojedynczych SMS'ów

2.1. Wysyłanie wiadomości Eco

2.2. Wysyłanie wiadomości Fast

2.3. Wysyłanie SMSa o zaplanowanej godzinie/dacie

2.4. Usuwanie zaplanowanej wiadomości

2.5. Wysyłanie SMS do grupy z książki telefonicznej

3. Wysyłanie masowe SMS'ów

3.1. Wysyłanie masowe spersonalizowanych wiadomości z wykorzystaniem parametrów oraz z wykorzystaniem parametru IDX

4. Wysyłanie wiadomości z wykorzystaniem szablonów

5. Wysyłanie WAP PUSH

6. Wysyłanie vCard

7. mail2SMS – Wysyłanie SMS za pomocą e-mail

8. Sprawdzenie ilości punktów na koncie

II. WYSYŁKA MMS

9. Wysyłanie MMS

III. WYSYŁKA VMS

10. Wysyłanie VMS

IV. SPRAWDZANIE RAPORTÓW DORĘCZENIA WIADOMOŚCI

11. Odbieranie raportu doręczenia wiadomości SMS - CALLBACK

12. Odbieranie raportów doręczenia wiadomości MMS - CALLBACK

13. Odbieranie raportów doręczenia wiadomości VMS - CALLBACK

V. ODBIÓR WIADOMOŚCI SMS ORAZ MMS

14. Odbiór SMS/MMS

14.1. Odbiór SMS

14.2. Odbiór MMS

VI. HLR – System sprawdzania numeru

15. System sprawdzania numeru

VII. PODSUMOWANIE

16. Uwagi końcowe

Dodatek nr 1 - Lista raportów doręczeń wiadomości

Dodatek nr 2 - Kody błędów

Dodatek nr 3 - Kodowanie

Dodatek nr 4 - Przykładowe skrypty

Historia zmian

Wersja	Data	Zmiany
Wersja 2.20	27.03.2012	1. Dodano parametr <i>check_idx</i> do wysyłki SMS oraz błąd 53 związany z tym parametrem. 2. Dodano parametr <i>skip_gsm</i> do wysyłki VMS oraz błąd 55 związany z tym parametrem.
Wersja 2.19	22.03.2012	1. Opis odwołań HLR 2. Wprowadzenie nowego błędu 52. System zabezpieczający przed przypadkowym zapętleniem się skryptu.
Wersja 2.18	02.02.2012	1. Nowe przykłady wysyłek wiadomości
Wersja 2.17	13.12.2011	1. Dodane nowe kody błędów 2. Aktualizacja opisu wysyłki VMS (usunięte efekty specjalne)
Wersja 2.16	31.10.2011	1. Opis parametrów <i>try</i> oraz <i>interval</i> dla wysyłek VMS
Wersja 2.15	14.10.2011	1. Dodano możliwość wysyłania wiadomości do grup z książki telefonicznej poprzez API 2. Dodano opis odbioru raportów doręczenia wiadomości VMS poprzez CALLBACK
Wersja 2.14	12.09.2011	1. Dodano opis API dla wiadomości VMS
Wersja 2.13	29.08.2011	1. Aktualizacja listy kodów błędów 2. Aktualizacja listy statusów wiadomości
Wersja 2.12	05.08.2011	1. Aktualizacja specyfikacji pod względem nowej wersji serwisu SMSAPI
Wersja 2.11	24.04.2011	1. Dodano adresy URL backup
Wersja 2.10	28.09.2010	1. Oddzielona specyfikacja SOAP od http.
Wersja 2.9	08.07.2010	1. Rozszerzenie funkcji sprawdzania ilości punktów w serwisie SMSAPI.pl o pokazywanie ilości dostępnych SMS.
Wersja 2.8	05.05.2010	1. Zmiana w zwracaniu wyniku podczas wysyłania wiadomości masowych korzystając z metody SOAP
Wersja 2.7	18.03.2010	1. Opis parametru fast=1
Wersja 2.6	24.02.2010	1. Filtr IP dla interfejsu API
Wersja 2.5	22.01.2010	1. Wysyłanie wiadomości MMS
Wersja 2.4	11.11.2009	1. Odbieranie wiadomości MMS
Wersja 2.3	28.08.2009	1. Opis korzystania z serwisu SMSAPI.pl za pomocą WebServices
Wersja 2.2	10.07.2009	1. Parametr nunicode 2. Parametr single=1 3. Sprawdzanie statusu wiadomości – kilka statusów za pomocą jednego zapytania
Wersja 2.1		1. Wysyłanie wiadomości Eco poprzez MailtoSMS 2. Zmiana odbioru raportów doręczenia poprzez CALLBACK
Wersja 2.0		1. Wysyłanie wiadomości Eco
Wersja 1.9		1. Wysyłanie wiadomości WAP PUSH 2. Wysyłka z wykorzystaniem parametru IDX
Wersja 1.8		1. Wysyłanie z wykorzystaniem szablonów 2. Wysyłka masowa ze spersonalizowaną treścią
Wersja 1.7		1. Wysyłka o określonej dacie/godzinie

UWAGA! Najnowsza specyfikacja techniczna SMSAPI znajduje się w zakładce POMOC na stronie <http://www.smsapi.pl/>

1. Wprowadzenie

Platforma SMSAPI została skierowana do użytkowników chcących rozbudować swoje aplikacje o system wysyłania i odbierania SMS-ów oraz MMS-ów oraz wysyłki wiadomości głosowych VMS. Aplikacja ta w prosty sposób umożliwi integrację dowolnego serwisu z bramką SMS, MMS i/lub VMS. Głównymi atutami naszego serwisu oprócz prostej implementacji jest możliwość nadawania wiadomości z własnej nazwy (maksymalnie 11 znaków). Każdy wysłany SMS, MMS oraz VMS za pośrednictwem systemu posiada unikalny numer identyfikacyjny pozwalający na sprawdzenie raportu doręczenia wiadomości.

1.1 Rozpoczęcie współpracy

W celu rozpoczęcia współpracy należy utworzyć konto w serwisie SMSAPI: <http://www.smsapi.pl/>

Utworzone konto jest gotowe do użytku, jednak zalecamy ustawienia własnej nazwy. Konta bez dodanej własnej nazwy posiadają jako nadawcę wiadomości nazwę serwisu „SMSAPI”. Ustawienie nazwy nadawcy jest usługą całkowicie darmową.

1.2 Filtr IP dla interfejsu API

W celu dodatkowego zabezpieczenia interfejsu API w menu „USTAWIENIA” → „API” zakładka „Filtr adresów IP” w polu **Filtr adresów IP** ustawić można adresy IP z których możliwa będzie wysyłka wiadomości (w przypadku próby dokonania wysyłki z innego IP system zwróci błąd: ERROR:105). Adresy należy oddzielić średnikami.

1.3 Hasło do interfejsu API

Hasło do interfejsu API po rejestracji jest takie same jak hasło do panelu WWW. Istnieje możliwość zmiany hasła do interfejsu API w zakładce Ustawienia → API. Zmiana hasła do panelu WWW nie wpływa na hasło API.

2. Wysyłanie pojedynczych SMS'ów

Adresy URL do połączenia z aplikacją – zwane dalej „Adresem połączenia”:

- <http://api.smsapi.pl/sms.do> - dla połączeń standardowych
- <http://api2.smsapi.pl/sms.do> - backup dla połączeń standardowych
- <https://ssl.smsapi.pl/sms.do> - dla połączeń szyfrowanych SSL 256bit v3.0
- <https://ssl2.smsapi.pl/sms.do> - backup dla połączeń szyfrowanych SSL 256bit v3.0

Wysyłanie SMS-ów odbywa się przez wysłanie metodą **GET** lub **POST** danych do adresu połączenia:

Parametr	Opis
<i>username</i> *	Nazwa użytkownika lub główny adres e-mail przypisany do konta w serwisie SMSAPI
<i>password</i> *	Hasło do Twojego konta w serwisie SMSAPI zaszyfrowane w MD5
<i>to</i> *	Numer odbiorcy wiadomości w formacie 48xxxxxxxx lub xxxxxxxxx. Np. 48505602702 lub 505602702.
<i>group</i> *	Nazwa grupy kontaktów z książki telefonicznej, do których ma zostać wysłana wiadomość
<i>message</i> *	Treść wiadomości. Standardowo do 160 znaków lub 70 znaków w przypadku wystąpienia chociaż jednego znaku specjalnego (polskie znaki uważane są za specjalne). Maksymalna długość wiadomości wynosi 459 znaków (lub 201 ze znakami specjalnymi) i jest wysłana jako 3 połączone SMS-y, obciążając konto zgodnie z aktualnym cennikiem. Więcej szczegółów odnośnie znaków specjalnych znajduje się w pkt 13.
<i>from</i>	Nazwa nadawcy wiadomości. Pozostawienie pola pustego powoduje wysłanie wiadomości od „SMSAPI”. Przyjmowane są tylko nazwy zweryfikowane. (&from=aktywna_nazwa). Pole nadawcy należy dodać po zalogowaniu na stronie SMSAPI, w zakładce USTAWIENIA → POLA NADAWCY.
<i>encoding</i>	Parametr określa kodowanie polskich znaków w SMS-ie. Domyślne kodowanie jest windows-1250. Jeżeli występuje konieczność zmiany kodowania, należy użyć parametru encoding z danymi: <ul style="list-style-type: none"> - dla iso-8859-2 (latin2) – należy podać wartość „iso-8859-2”, - dla utf-8 – należy podać wartość „utf-8”.

<i>flash</i>	Wysyłanie wiadomości trybem „flash”, odbywa się poprzez podanie parametru flash o wartości „1”. SMS-y flash są automatycznie wyświetlane na ekranie głównym telefonu komórkowego i nie są przechowywane w skrzynce odbiorczej (jeśli nie zostaną zapisane). (&flash=1)
<i>test</i>	Wiadomość nie jest wysyłana, wyświetlana jest jedynie odpowiedź (w celach testowych). (&test=1)
<i>details</i>	W odpowiedzi zawarte jest więcej szczegółów. (Treść wiadomości, długość wiadomości, ilość części z jakich składa się wiadomość). (&details=1)
<i>date</i>	Data w formacie unix unixtime. Określa kiedy wiadomość ma być wysłana. (&date=1287734110). W przypadku wstawienia daty przeszłej wiadomość zostanie wysłana od razu.
<i>datacoding</i>	Parametr pozwalający na wysyłanie wiadomości WAP PUSH. (&datacoding=bin)
<i>idx</i>	Opcjonalny parametr użytkownika wysyłany z wiadomością a następnie zwracany przy wywołaniu zwrotnym CALLBACK. (&idx=123)
<i>check_idx</i>	Pozwala zabezpieczyć przed wysłaniem dwóch wiadomości z identyczną wartością parametru <i>idx</i> . W przypadku ustawienia parametru (&check_idx=1) system sprawdza czy wiadomość z takim idx już została przyjęta, jeśli tak zwracany jest błąd 53.
<i>single</i>	Ustawienie 1 zabezpiecza przed wysłaniem wiadomości składających się z kilku części. (ERROR:12)
<i>eco</i>	Ustawienie parametru &eco=1 spowoduje wysłanie wiadomości Eco (brak możliwości wyboru pola nadawcy, wiadomość wysyłana z losowego numeru dziewięciocyfrowego) szczegóły dotyczące wiadomości Eco znajdują się na naszej stronie: http://www.smsapi.pl/
<i>nounicode</i>	Ustawienie 1 zabezpiecza przed wysłaniem wiadomości ze znakami specjalnymi (w tym polskimi) (ERROR:11).
<i>fast</i>	Ustawienie 1 spowoduje wysłanie wiadomości przy wykorzystaniu osobnego kanału zapewniającego szybkie doręczenie wiadomości Fast. Z parametru korzystać można podczas wysyłania wiadomości Pro oraz Eco, ilość punktów za wysyłkę pomnożona będzie przez 1.5 Uwaga! Dla tego parametru zabronione jest prowadzenie wysyłek masowych i marketingowych.

* - pole obowiązkowe

Uwaga! Parametry *group* oraz *to* są zamienne, w odwołaniu musi się pojawić jeden z tych dwóch parametrów. Brak jednego z tych dwóch parametrów lub wystąpienie ich obu spowoduje niewysłanie wiadomości oraz zwrócenie błędu ERROR: 13.

Zapytanie: **[http://api.smsapi.pl/sms.do?](http://api.smsapi.pl/sms.do?username=uzytkownik&password=hasloMD5&from=nazwa&to=48500500500&message=test%20wiadomosci)
username=uzytkownik&password=hasloMD5&from=nazwa
&to=48500500500&message=test wiadomosci**

Odpowiedź: **OK:<ID>:<POINTS>**

lub w razie błędu

ERROR:<ERR>

<ID> numer ID wysłanego SMS-a. Numer należy zachować w celu sprawdzenia statusu wiadomości.

<POINTS> ilość pobranych punktów (np. SMS składający się z 3 części zwróci wartość 3 x (stawka SMS)).

<ERR> kod błędu (zobacz dodatek – kody błędów)

Przykład: **OK:17101000090360359:0.14**
lub
ERROR:102

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **<http://api2.smsapi.pl/>**

2.1 Wysyłanie wiadomości Eco

Wysyłanie wiadomości Eco odbywa się z losowego numeru (brak możliwości wyboru pola nadawcy). W celu wysłania wiadomości Eco należy użyć parametr `&eco=1` (również przy wysyłaniu wiadomości masowych).

Zapytanie: **`http://api.smsapi.pl/sms.do?username=uzytkownik&password=hasloMD5&to=48500500500&eco=1&message=tes t wiadomosci Eco`**

Odpowiedź: **`OK:<ID>:<POINTS>`**

lub w razie błędu

`ERROR:<ERR>`

<ID> numer ID wysłanego SMS-a. Numer należy zachować w celu sprawdzenia statusu wiadomości.

<POINTS> ilość pobranych punktów (zobacz tabelę poniżej).

<ERR> kod błędu (zobacz dodatek – kody błędów)

Przykład: **`OK:17101000090360359:0.06`**

lub

`ERROR:102`

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **`http://api2.smsapi.pl/`**.

2.2 Wysyłanie wiadomości Fast

Ustawienie parametru `fast=1` spowoduje wysłanie wiadomości przy wykorzystaniu osobnego kanału zapewniającego najwyższy priorytet wysyłki. Z parametru korzystać można podczas wysyłania wiadomości Pro oraz Eco, ilość punktów za wysyłkę pomnożona będzie przez 1.5

Zapytanie: **`http://api.smsapi.pl/sms.do?username=uzytkownik&password=hasloMD5&to=48500500500&fast=1&message=tes t wiadomosci Fast`**

Odpowiedź: **`OK:<ID>:<POINTS>`**

lub w razie błędu

`ERROR:<ERR>`

<ID> numer ID wysłanego SMS-a. Numer należy zachować w celu sprawdzenia statusu wiadomości.

<POINTS> ilość pobranych punktów (zobacz tabelę poniżej).

<ERR> kod błędu (zobacz dodatek – kody błędów)

Przykład: **`OK:17101000090360359:0.21`**

lub

`ERROR:102`

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **`http://api2.smsapi.pl/`**.

UWAGA! Dla tego parametru zabronione jest prowadzenie wysyłek masowych i marketingowych. W przypadku wysyłki do więcej niż jednego numeru w jednym odwołaniu wiadomości zostaną wysłane jako zwykłe wiadomości.

2.3 Wysłanie SMS-a o zaplanowanej godzinie/dacie

W celu wysłania wiadomości o określonej godzinie/dacie musi zostać użyty parametr **&date=** wskazujący datę którą należy skonwertować do postaci unixtime. Do konwersji wykorzystać można narzędzie znajdujące w zakładce POMOC na stronie <https://www.smsapi.pl>.

Zapytanie: **http://api.smsapi.pl/sms.do?username=uzytownik&password=hasloMD5&to=48500500500&date=1577878200&message=wiadomosc_zaplanowana**

Odpowiedź: **OK:<ID>:<POINTS>**

lub w razie błędu

ERROR:<ERR>

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **<http://api2.smsapi.pl/>**.

2.4. Usuwanie zaplanowanej wiadomości

Parametr	Opis
<i>username</i> *	Nazwa użytkownika lub główny adres e-mail przypisany do konta w serwisie SMSAPI
<i>password</i> *	Hasło do Twojego konta w serwisie SMSAPI zaszyfrowane w MD5
<i>sch_del</i> *	ID zaplanowanej wiadomości (zwrócony podczas wysyłki).

* - pole obowiązkowe

Zapytanie: **http://api.smsapi.pl/sms.do?username=uzytownik&password=haslo&sch_del=09040616088106874**

Odpowiedź: **OK**
lub gdy ID wiadomości nie istnieje
ERROR:301

Przykład ID: 09040616088106874

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **<http://api2.smsapi.pl/>**.

2.5 Wysłanie SMS do grupy z książki telefonicznej

W celu wysłania wiadomości do określonej grupy należy w pierwszej kolejności stworzyć ją w panelu WWW w zakładce KSIĄŻKA TELEFONICZNA. Poniżej przedstawiamy przykład wysyłki do grupy:

Zapytanie: **http://api.smsapi.pl/sms.do?username=uzytownik&password=hasloMD5&group=Testowa_grupa&message=wiadomosc_wyslana_do_grupy_z_ksiazki_telefoniczne**

Odpowiedź: **OK:<ID>:<POINTS>**

lub w razie błędu

ERROR:<ERR>

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **<http://api2.smsapi.pl/>**.

3. Wysłanie masowe SMS-ów

Wysyłanie SMS-ów do grupy odbiorców odbywa się tak jak do pojedynczego odbiorcy (opis w punkcie 2) lecz z podaniem wielu numerów telefonów w parametrze „to”.

Dodatkowo cała wysyłka powinna odbywać się metodą **POST** do adresu połączenia. W przypadku wysyłki metodą **GET** przy większej ilości numerów, część parametrów może zostać ucięta i w efekcie niedostarczona w całości do serwisu SMSAPI.

Jeżeli cena za wysłanie wszystkich SMS-ów jest większa od ilości punktów dostępnych w serwisie, zostanie zwrócony komunikat o błędzie (**103**) i SMS-y nie zostaną wysłane. Jeżeli zostały podane błędne numery (nierozpoznane przez serwis SMSAPI ze względu na błędny prefiks), wszystkie SMS-y zostaną wysłane z pominięciem tych numerów, wówczas ilość otrzymanych raportów będzie różnić się od ilości wysłanych SMS-ów. Sytuacja ta nie dotyczy numerów z poprawnym prefiksem (np. 500000000), gdzie SMS zostanie wysłany, a następnie odrzucony gdyż numer nie istnieje (punkty za wysłanie SMS-a zostaną pobrane). Numery, które się powtarzają zostaną wysłane tylko 1 raz.

Zapytanie: **http://api.smsapi.pl/sms.do?
username=uzytownik&password=hasloMD5&from=nazwa
&to=48500500500,48501501501,48502502502&message=test wiadomosci**

Odpowiedź: **OK:<ID>:<POINTS>:<PHONE>;...;...;...**

<ID> numer ID wysłanego SMS-a. Numer należy zachować w celu sprawdzenia statusu wiadomości.

<POINTS> ilość pobranych punktów (np. sms wysłany w 2 wiadomościach zwróci wartość 2 x (stawka za SMS)).

<PHONE> numer telefonu odbiorcy.

Przykład: **OK:17101000090567759:0.14:500500500;OK:171010000903455357:0.14:501500501;
OK:17101000096577326:0.14:502502502;**

Warto zauważyć, że przy wysyłce SMS-ów do grupy odbiorców dodatkowo zwracany jest numer telefonu oraz każdy SMS jest zakańczany średnikiem (również ostatni).

Zalecana maksymalna ilość jednorazowej wysyłki (jedno wywołanie) wynosi **10 000** wiadomości metodą **POST** oraz do **200** wiadomości metodą **GET**.

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **http://api2.smsapi.pl/**

3.1. Wysłanie masowe spersonalizowanych wiadomości z wykorzystaniem parametrów oraz z wykorzystaniem parametru IDX

Istnieje możliwość wysłania do 100 spersonalizowanych wiadomości przy pomocy jednego wywołania wykorzystując parametry. W razie potrzeby wysłania większej ilości wiadomości można wywołać równoległe kilka odwołań.

Parametry powinny być zdefiniowane w wywołaniu jako **param1**, **param2**, **param3**, **param4**, które zastępowały będą [%1%], [%2%], [%3%] oraz [%4%] w treści wiadomości. Wartości parametrów muszą być oddzielone znakiem pipe „|” według wzoru:

param1=Ania|Michal|Andrzej¶m2=Nowak|Kowalski|Nowakowski

Liczba parametrów musi być identyczna z liczbą numerów do których wysłane mają być wiadomości w innym przypadku zwrócony będzie błąd: ERROR:18 i wiadomość nie zostanie wysłana.

WAŻNE! Długość wiadomości może być różna w zależności od długości parametru!

Jeżeli jeden z numerów będzie błędny zostanie on pominięty i wiadomości zostaną wysłane z pominięciem tego numeru.

Parametry

Parametry wykorzystane mogą być po zdefiniowaniu ich miejsca w wiadomości:

Parametr	Wartość
[%1%]	Tekst parametru 1 (param1)
[%2%]	Tekst parametru 2 (param2)
[%3%]	Tekst parametru 3 (param3)
[%4%]	Tekst parametru 4 (param4)

Przykład:

```
http://api.smsapi.pl/sms.do?sername=login&password=hasloMD5&from=nadawca
&to=48600111222,48500111222&message=Test wiadomosci, parametr1: [%1%] parametr2: [%2%]
&param1=Jan|Ania&param2=30|40
```

Wiadomości będą miały następującą treść:

Wiadomość 1: *Test wiadomosci, parametr1: Jan parametr2: 30*

Wiadomość 2: *Test wiadomosci, parametr1: Ania parametr2: 40*

Istnieje możliwość wysyłania masowych wiadomości z parametrem użytkownika **IDX** różnym dla każdej wiadomości. Parametr ten zwracany jest w wywołaniu zwrotnym CALLBACK. W celu przypisania różnych parametrów różnym wiadomością należy oddzielić je znakiem "pipe" co przedstawia poniższy przykład:

&idx=idx1|idx2|idx3|idx4

Liczba parametrów **IDX** musi być identyczna z liczbą wysyłanych wiadomości.

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **http://api2.smsapi.pl/**.

4. Wysyłanie wiadomości z wykorzystaniem szablonów

Przy wykorzystaniu szablonów w bardzo prosty sposób zmienić można treść SMS-ów powiadamiających (w sklepach, serwisach internetowych, klinikach itp.) bez ingerencji w kod skryptu odpowiedzialnego za wysyłanie wiadomości SMS.

W celu skorzystania z szablonów należy:

- Po zalogowaniu na stronie <http://www.smsapi.pl> dodać szablon w menu „Ustawienia” → „Szablony”
- W miejsce, w którym ma występować parametr należy wpisać [%N%] gdzie N to liczba od 1 do 4
- W celu wykorzystania szablonu w interfejsie API należy wybrać jeden z szablonów poprzez wpisanie parametru &templates=nazwa_szablonu
- Oprócz podstawowych parametrów w trakcie używania szablonów dostępne są parametry:

Parametr	Opis
<i>template</i>	Nazwa szablonu
<i>paramN</i>	Parametr wstawiany w miejsce [%N%] w szablonie gdzie N to liczba od 1 do 4
<i>single</i>	Jeżeli wiadomość będzie składała się z więcej niż 160 znaków nie zostanie wysłana i zwrócony zostanie ERROR:12 (&single=1)

Przykład:

Nazwa szablonu: **Powiadom**
 Treść szablonu: Witaj [%1%], Twoje zamówienie zostało zrealizowane. Numer listu przewozowego to [%2%] sledzenie przesyłek na stronie.

<http://api.smsapi.pl/sms.do?sename=login&password=hasloMD5&from=nadawca&to=48600111222&template=powiadom¶m1=Marcin¶m2=BG12344423>

Treść wiadomości: Witaj **Marcin**, Twoje zamówienie zostało zrealizowane. Numer listu przewozowego to **BG12344423** sledzenie przesyłek na stronie.

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **<http://api2.smsapi.pl/>**.

5. Wysyłanie WAP PUSH

W celu wysłania wiadomości WAP PUSH link oraz treść wiadomości muszą być przekonwertowane do postaci binarnej. Do tego celu służy konwerter który można znaleźć w zakładce POMOC na stronie <https://www.smsapi.pl>.

Zwrócony wynik konwersji należy wkleić jako treść wiadomości. W wywołaniu dodatkowo należy użyć parametrów **&datacoding=bin** oraz **&udh=0605040b8423f0**. Poniżej przedstawiony został przykład wysłania wiadomości WAP Push.

Przykład:

<http://api.smsapi.pl/sms.do?username=uzytownik&password=hasloMD5&from=nazwa&to=48501502503&datacoding=bin&udh=0605040b8423f0&message=0605040b8423f0860601ae02056a0045c60c03687474703a2f2f777772e736d736170692e706c000701034e61737a61207374726f6e613a000101>

W razie problemów podczas odwoływania się do standardowego adresu (jak w przykładzie powyżej) można wykorzystać adres URL backup.

6. Wysyłanie vCard

W celu wysłania wiadomości vCard dane kontaktu muszą być przekonwertowane do postaci binarnej. Do tego celu służy konwerter który można znaleźć w zakładce POMOC na stronie **<http://www.smsapi.pl/>**

Zwrócony wynik konwersji należy wkleić jako treść wiadomości. W wywołaniu dodatkowo należy użyć parametrów **&datacoding=bin** oraz **&udh=06050423F40000**. Poniżej przedstawiony został przykład wysłania wiadomości vCard.

Przykład:

<http://api.smsapi.pl/sms.do?username=uzytownik&password=hasloMD5&from=nazwa&to=48501502503&datacoding=bin&udh=06050423F40000&message=>

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **<http://api2.smsapi.pl/>**.

7. mail2SMS – Wysyłanie SMS za pomocą e-mail

Aby wysłać smsa za pomocą maila należy wysłać maila według schematu:

ADRES: **send.do@smsapi.pl**
 TEMAT: **login@haslo_32znaki_w_md5**
 TREŚĆ: **from=nadawca&to=numer&message=tresc_wiadomosci**

UWAGA! Hasło należy podać w formie zaszyfrowanej algorytmem MD5!

Przykład:

ADRES: **send.do@smsapi.pl**
 TEMAT: **login@8456fky567gb3bg37b357b3457b3457**
 TREŚĆ: **from=nadawca&to=numer&message=tresc_wiadomosci**

Dodanie parametru **raport=1** spowoduje odsyłanie maila z raportem (potwierdzenie wysłania wiadomości lub błąd – jest to przydatne w trakcie testowania usługi):

ADRES: **send.do@smsapi.pl**
 TEMAT: **login@haslo_32znaki_w_md5**
 TREŚĆ: **from=nadawca&to=numer&raport=1&message=tresc wiadomosci**

Wiadomości mogą być wysyłane w kodowaniu plain / quotedprintable / base64.

Ważne! Numer telefonu nie może zawierać znaku plus "+" na początku. Parametr message musi znajdować się na końcu treści wiadomości.

Nazwa nadawcy (zmienna &from=) musi być aktywna.

W celu wysłanie wiadomości Eco przez mail2SMS należy dodać parametr eco=1.

8. Sprawdzenie ilości punktów na koncie

Parametr	Opis
<i>username</i> *	Nazwa użytkownika lub główny adres e-mail przypisany do konta w serwisie SMSAPI
<i>password</i> *	Hasło do Twojego konta w serwisie SMSAPI zaszyfrowane w MD5
<i>points</i> *	Należy podać wartość „1”
<i>details</i>	Dodatkowo wyświetlana jest ilość wiadomości Pro oraz Eco

* - pole obowiązkowe

Zapytanie: **[http://api.smsapi.pl/sms.do?username=uzytkownik&password=hasloMD5&points=1&details=1](http://api.smsapi.pl/sms.do?username=<u>uzytkownik</u>&password=<u>hasloMD5</u>&points=1&details=1)**

Odpowiedź: **Points:<PKT>;<Pro>;<Eco>;<MMS>;<VMS_GSM>;<VMS_STAC>**
<PKT> ilość punktów dostępnych dla tego użytkownika w serwisie SMSAPI
<Pro> ilość wiadomości Pro w serwisie SMSAPI
<Eco> ilość wiadomości Eco w serwisie SMSAPI
<MMS> ilość wiadomości MMS w serwisie SMSAPI
<VMS_GSM> ilość jednostek VMS do sieci GSM w serwisie SMSAPI
<VMS_STAC> ilość jednostek VMS do sieci stacjonarnych w serwisie SMSAPI

Przykład: **Points: 100.0000;606;1428;333;476;714**

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **<http://api2.smsapi.pl/>**.

9. Wysyłanie wiadomości MMS

Wysyłanie wiadomości MMS odbywa się za pomocą odwołania do adresu **<http://api.smsapi.pl/mms.do>** (dla połączeń ssl: **<https://ssl.smsapi.pl/mms.do>**) i podanie odpowiednich parametrów.

Poniższa tabela przedstawia parametry niezbędne do wysłania wiadomości MMS.

Parametr	Opis
<i>username</i> *	Nazwa użytkownika lub adres e-mail przypisany do konta w serwisie SMSAPI
<i>password</i> *	Hasło do Twojego konta w serwisie SMSAPI zaszyfrowane w MD5
<i>to</i> *	Numer odbiorcy wiadomości w formacie 48xxxxxxxx lub xxxxxxxx. Np. 48505602702 lub 505602702.
<i>group</i> *	Nazwa grupy kontaktów z książki telefonicznej, do których ma zostać wysłana

	wiadomość
<i>subject</i> *	Temat wiadomości MMS
<i>date</i>	Data w formacie unixtime kiedy wiadomość ma być wysłana
<i>smil</i> *	Wiadomość MMS w formacie SMIL (przykłady w punkcie 3)
<i>check_idx</i>	Pozwala zabezpieczyć przed wysłaniem dwóch wiadomości z identyczną wartością parametru <i>idx</i> . W przypadku ustawienia parametru (&check_idx=1) system sprawdza czy wiadomość z takim <i>idx</i> już została przyjęta, jeśli tak zwracany jest błąd 53.

* - pole obowiązkowe

Uwaga! Parametry *group* oraz *to* są zamienne, w odwołaniu musi się pojawić jeden z tych dwóch parametrów. Brak jednego z tych dwóch parametrów lub wystąpienie ich obu spowoduje niewysłanie wiadomości oraz zwrócenie błędu ERROR: 13.

Zapytanie: **http://api.smsapi.pl/mms.do?username=uzytownik
&password=hasloMD5&to=48500500500&subject=Testowy MMS&smil=[smil]**

Odpowiedź: **OK:<ID>:<POINTS>**

lub w razie błędu

ERROR:<ERR>

<ID> numer ID wysłanego MMSa. Numer należy zachować w celu sprawdzenia statusu wiadomości.

<POINTS> ilość pobranych punktów

<ERR> kod błędu (zobacz dodatek – kody błędów)

Przykład: **OK:17101000090360359:0.4**
lub
ERROR:102

W razie problemów podczas odwoływania się do podstawowego adresu URL (jak w przykładzie powyżej) można wykorzystać adres URL backup **http://api2.smsapi.pl/**.

10. Wysłanie wiadomości VMS

Wysyłanie wiadomości VMS odbywa się za pomocą odwołania do adresu <http://api.smsapi.pl/vms.do> (dla połączeń ssl: <https://ssl.smsapi.pl/vms.do>) i podanie odpowiednich parametrów.

Poniższa tabela przedstawia parametry niezbędne do wysłania wiadomości VMS.

Parametr	Opis
<i>username</i> *	Nazwa użytkownika lub adres e-mail przypisany do konta w serwisie SMSAPI
<i>password</i> *	Hasło do Twojego konta w serwisie SMSAPI zaszyfrowane w MD5
<i>to</i> *	Numer odbiorcy wiadomości w formacie 48xxxxxxxx lub xxxxxxxx. Np. 48505602702 lub 505602702.
<i>group</i> *	Nazwa grupy kontaktów z książki telefonicznej, do których ma zostać wysłana wiadomość
<i>tts</i> *	Treść wiadomości głosowej w postaci tekstu

<i>file *</i>	Treść wiadomości w postaci pliku wave, akceptowalny link do pliku umieszczonego w sieci lub plik wysłany postem metodą multipart from-data
<i>date</i>	Data w formacie unixtime kiedy wiadomość ma być wysłana, parametr ten jest opcjonalny, jego brak spowoduje natychmiastową wysyłkę
<i>try</i>	Ilość prób połączenia (dopuszczalne wartości od 1 do 6)
<i>interval</i>	Czas w sekundach po jakim powtórzone ma być połączenie w przypadku jego nieodebrania lub odrzucenia (dopuszczalne wartości od 1800 do 7200s)
<i>skip_gsm</i>	Ustawienie tego parametru (&skip_gsm=1) spowoduje pominięcie telefonów komórkowych podczas wysyłki i wysłanie wiadomości tylko do numerów stacjonarnych
<i>check_idx</i>	Pozwala zabezpieczyć przed wysłaniem dwóch wiadomości z identyczną wartością parametru <i>idx</i> . W przypadku ustawienia parametru (&check_idx=1) system sprawdza czy wiadomość z takim <i>idx</i> już została przyjęta, jeśli tak zwracany jest błąd 53.

* - pole obowiązkowe

Uwaga! Parametry *tts* oraz *file* lub *wystąpienie* są zamienne, w odwołaniu musi się pojawić jeden z tych dwóch parametrów. Brak jednego z tych dwóch parametrów lub wystąpienie ich obu spowoduje niewysłanie wiadomości oraz zwrócenie błędu ERROR: 11.

Uwaga! Parametry *group* oraz *to* są zamienne, w odwołaniu musi się pojawić jeden z tych dwóch parametrów. Brak jednego z tych dwóch parametrów lub wystąpienie ich obu spowoduje niewysłanie wiadomości oraz zwrócenie błędu ERROR: 13.

Zapytanie: **http://api.smsapi.pl/vms.do?username=uzytownik
&password=hasloMD5&to=48500500500&tts=Wiadomość testowa**

lub: **http://api.smsapi.pl/vms.do?username=uzytownik
&password=hasloMD5&to=48500500500&file=http://hiper.lacze.do/pliku.wav**

Odpowiedź: **OK:<ID>:<POINTS>**

lub w razie błędu

ERROR:<ERR>

<ID> numer ID wysłanego VMSa. Numer należy zachować w celu sprawdzenia statusu wiadomości.

<POINTS> ilość pobranych punktów

<ERR> kod błędu (zobacz dodatek – kody błędów)

Przykład: **OK:17101000090360359:0.4**
lub
ERROR:102

Uwaga! Z szacunku dla abonentów wszystkich sieci, usługi głosowe dostępne są wyłącznie w godzinach od 8:00 do 22:00. Zlecenia wysyłki w pozostałych godzinach będą rozpatrywane indywidualnie.
W przypadku gdy wysłanie wiadomości VMS o zadanej porze nie jest możliwe wiadomość jest przyjęta i data wysyłki zostaje ustawiona na pierwszą możliwą.

11. Odbieranie raportu doręczenia wiadomości SMS - CALLBACK

W celu sprawdzenia statusu wiadomości należy w panelu klienta z zakładce „Ustawienia” → „API” ustawić adres do skryptu do którego przekazywany będzie raport doręczenia wiadomości. Aby móc wprowadzić adres callback, skrypt musi być umieszczony w podanej lokalizacji.

Np.: http://www.moja_strona.pl/status_update.php

Ważne, aby podany adres był poprawnym adresem, tzn. aby wskazany skrypt istniał na serwerze.

W momencie dodawania adresu do skryptu callback w panelu SMSAPI, weryfikowany jest on poprzez odwołanie się do niego z pustą tablicą GET.

Do skryptu wysyłane będą statusy wiadomości (od 1 do 5 w zależności od tego dla ilu wiadomości zmienił się status). Parametry będą podane metodą GET i będą oddzielone przecinkami np. :

```
$_GET['MsgId']=09062414383994024,09062414383994025
$_GET['status']=403,404
```

Poniższa tabela przedstawia wysyłane parametry:

Parametr	Opis
MsgId*	ID wysłanej wiadomości
status*	Kod zwrotu wg tabeli kodów zwrotów z 'Dodatek nr 1'
idx*	Opcjonalny parametr użytkownika wysłany z SMS'em
doneDate*	Czas dostarczenia wiadomości w formacie unixtime
username*	Nazwa użytkownika wysyłającego wiadomość
points*	Ilość punktów pobranych za wysyłkę wiadomości
to*	Numer na jaki wysłana była wiadomość

* - należy zachować wielkość znaków!

Skrypt powinien zwracać OK, w innym przypadku nastąpią ponowne odwołania z interwałem 300 sekund.

12. Odbieranie raportu doręczenia wiadomości MMS - CALLBACK

W celu sprawdzenia statusu wiadomości należy w panelu klienta „Ustawienia” → „API” ustawić adres do skryptu do którego przekazywany będzie raport doręczenia wiadomości.

Przykład: http://www.moja_strona.pl/status_update.php

Ważne, aby podany adres był poprawnym adresem, tzn. aby wskazany skrypt istniał na serwerze.

W momencie dodawania adresu do skryptu callback w panelu SMSAPI, weryfikowany jest on poprzez odwołanie się do niego z pustą tablicą GET.

Do skryptu wysyłane będą statusy wiadomości. Parametry będą podane metodą GET np. :

```
MsgId=09062414383994024
status=404
```

Poniższa tabela przedstawia wysyłane parametry:

Parametr	Opis
MsgId*	ID wysłanej wiadomości
status*	Kod doręczenia wiadomości
doneDate*	Czas dostarczenia wiadomości
username*	Nazwa użytkownika z którego została wysłana wiadomość

* - należy zachować wielkość znaków!

Skrypt powinien zwracać OK (np. echo 'OK');, w innym przypadku nastąpią ponowne odwołania z interwalem 300 sekund.

13. Odbieranie raportu doręczenia wiadomości VMS- CALLBACK

W celu sprawdzenia statusu wiadomości należy w panelu klienta „Ustawienia” → „API” ustawić adres do skryptu do którego przekazywany będzie raport doręczenia wiadomości.

Przykład: http://www.moja_strona.pl/status_update.php

Ważne, aby podany adres był poprawnym adresem, tzn. aby wskazany skrypt istniał na serwerze. W momencie dodawania adresu do skryptu callback w panelu SMSAPI, weryfikowany jest on poprzez odwołanie się do niego z pustą tablicą GET.

Do skryptu wysyłane będą statusy wiadomości. Parametry będą podane metodą GET np. :

MsgId=09062414383994024
status=404

Poniższa tabela przedstawia wysyłane parametry:

Parametr	Opis
MsgId*	ID wysłanej wiadomości
status*	Kod doręczenia wiadomości
donedate*	Czas dostarczenia wiadomości
username*	Nazwa użytkownika z którego została wysłana wiadomość

* - należy zachować wielkość znaków!

Skrypt powinien zwracać OK (np. echo 'OK');, w innym przypadku nastąpią ponowne odwołania z interwalem 300 sekund.

14. Odbiór SMS/MMS

Serwis **SMSAPI** oferuje również odbiór wiadomości SMS oraz MMS. Odbierać można odpowiedzi na wysłane wiadomości Eco (do 24h od momentu wysłania wiadomości) lub wiadomości SMS i/lub MMS wysłane na wykupiony numer dedykowany.

14.1. Odbiór SMS

Odbiór wiadomości odbywać się będzie za pomocą skryptu, do którego adres należy ustawić w zakładce „Ustawienia” → „API”. Działanie skryptu opisane jest poniżej:

Po odebraniu wiadomości odwoływać będziemy się do skryptu stworzonego przez Państwa, który powinien obsługiwać 4 parametry z tablicy POST:

Parametr	Opis
sms_to	numer telefonu odbiorcy
sms_from	numer telefonu nadawcy
sms_text	treść wiadomości
sms_date	czas w postaci unixtime pobrany z SMS'a
MsgId	ID wiadomości ECO na którą jest to odpowiedź, dla odpowiedzi na numer dedykowany pole to będzie puste
username	Nazwa użytkownika (login) do którego wiadomość została przydzielona

smsc

Centrum SMS z którego przyszła wiadomość

Dane wysyłane są w kodowaniu UTF8.

Skrypt powinien zwracać OK (np. echo 'OK');, w innym przypadku następują ponowne odwołania z interwałem 300 sekund.

Odwołanie może działać z szyfrowaniem SSL (adres <https://...>).

W momencie dodawania adresu do skryptu callback w panelu SMSAPI, weryfikowany jest on poprzez odwołanie się do niego z pustą tablicą GET.

14.2. Odbiór MMS

Odbiór wiadomości odbywać się będzie za pomocą skryptu, do którego adres należy ustawić w zakładce „Ustawienia” → „API”. Działanie skryptu opisane jest poniżej:

Po odebraniu wiadomości odwoływać będziemy się do skryptu stworzonego przez Państwa, który powinien obsługiwać parametry z tablicy POST oraz FILES podane poniżej:

Tablica POST:

Parametr	Opis
mms_to	numer telefonu odbiorcy
mms_from	numer telefonu nadawcy
mms_subject	temat wiadomości
mms_date	czas w postaci unixtime pobrany z MMS'a

Tablica FILES:

Parametr	Opis
name	oryginalna nazwa wysyłanego pliku
type	typ MIME wysyłanego pliku (JPEG, GIF, ...)
tmp_name	tymczasowa nazwa pliku, który został wysłany na serwer
error	numer błędu (0 oznacza prawidłowe wysłanie)
size	rozmiar wysyłanego pliku (w bajtach)

Dane wysyłane są w kodowaniu UTF8.

Skrypt powinien zwracać OK (np. echo 'OK');, w innym przypadku następują ponowne odwołania z interwałem 300 sekund. PRZYKŁAD

Odwołanie może działać z szyfrowaniem SSL (adres <https://...>).

W momencie dodawania adresu do skryptu callback w panelu SMSAPI, weryfikowany jest on poprzez odwołanie się do niego z pustą tablicą GET.

15. HLR - System sprawdzania numeru

HLR (ang. Home Location Register) – Rejestr Abonentów Macierzystych – element infrastruktury telekomunikacyjnej zawierający informacje o każdym aktywnym numerze GSM. W celu skorzystania z tej opcji należy wysłać żądanie pod adres: <https://ssl.smsapi.pl/hlr.do> z parametrami opisanymi w tabeli poniżej. Informacje o numerach będą wysłane do skryptu do którego adres należy podać w panelu klienta na stronie <https://ssl.smsapi.pl> w menu „Ustawienia” → „API” zakładka „Adresy callback” w polu „Adres callback - odwołania HLR”.

Ważne aby wprowadzony adres URL prowadził do istniejącego, działającego skryptu.

Po sprawdzeniu numeru w systemie HLR informacja o nim zostanie przesłana do podanego adresu URL w postaci tablicy POST. W jednym odwołaniu może znajdować się do 20 numerów.

Zapytanie: **[http://ssl.smsapi.pl/hlr.do?
username=uzytownik&password=hasloMD5&number=48600600600,48500600700](http://ssl.smsapi.pl/hlr.do?username=uzytownik&password=hasloMD5&number=48600600600,48500600700)**

Odpowiedź: **OK:<NUMER>:<ID>:<POINTS>;OK<NUMER>:<ID>:<POINTS>**

lub w razie błędu

ERROR:<NUMER>:<ERR>

<NUMER> sprawdzany numer
<ID> unikalny numer ID sprawdzania danego numeru
<POINTS> ilość pobranych punktów (zobacz tabelę poniżej).
<ERR> kod błędu (zobacz dodatek – kody błędów)

Przykład: **OK:48600600600:80625:0.05;OK:48500600700:80627:0.05**
 lub
OK:48600600600:80625:0.05;ERROR:48500600700:13

Poniższa tabela przedstawia parametry wysyłane do skryptu:

Parametr	Opis
id*	id zwrócony podczas sprawdzania numeru
number*	sprawdzany numer
mcc*	numer identyfikujący kraj (ang. Mobile Country Code)
mnc*	numer sieci w danym kraju (ang. Mobile Network Code)
info*	nazwa sieci, do której należy numer, lub opis błędu
status*	OK kiedy numer jest poprawny, FAIL kiedy numer jest błędny (np. wyłączony, nieaktywny itp.)
date*	Data, w formacie UNIX timestamp, kiedy numer był sprawdzany
ported*	0 jeżeli numer jest nieprzeniesiony, 1 jeżeli numer jest przeniesiony
ported_from*	null kiedy numer jest nieprzeniesiony lub nazwa sieci z której numer został przeniesiony

* - należy zachować wielkość znaków!

Dane wysyłane są w kodowaniu UTF8.

Przykład danych przesyłanych do skryptu CALLBACK:

```
POST[0]
--POST[id]=80625
--POST[number]=48600600600
--POST[mcc]=260
--POST[mnc]=2
--POST[info]=T-Mobile
--POST[status]=OK
--POST[date]=1302703609
--POST[ported]=0
--POST[ported_from]=null
```

```
POST[1]
--POST[id]=80627
--POST[number]=48500600700
--POST[mcc]=260
--POST[mnc]=3
--POST[info]=Orange
--POST[status]=OK
--POST[date]=1302703609
--POST[ported]=0
--POST[ported_from]=null
```

Skrypt powinien zwracać OK (np. echo 'OK!');, w innym przypadku następują ponowne odwołania z interwałem 300 sekund.

Odwołanie może działać z szyfrowaniem SSL (adres <https://..>).

16. Uwagi końcowe

Numery telefonu powinny być podawane w formacie 11 cyfrowym (np. 48502602702). W przypadku gdy zostanie podany numer bez prefiksu 48, zostanie on automatycznie dopisany.

Za znaki specjalne uważa się wszystkie znaki nie spełniające wyrażenia regularnego: @£\$¥èùìòçøå_^[~]|ÆæßÉ!"#%&'()*+,-./0-9:;<=>?A-ZÄÖÑÛŞ¿a-zäöñüà <enter>

Przykładowa implementacja w skryptach PHP:

```
// Funkcja zwraca wynik 1 lub 0 w zależności czy w zmiennej $_message występują
// znaki specjalne.

function is_spec_char($_message) {
    mb_regex_encoding('UTF-8');
    if(mb_eregi("[^\@£$¥èùìòçøå_^[~]|ÆæßÉ!"#%&'()*+,-./0123456789:;<=>?A-ZÄÖÑÛŞ¿a-zäöñüà \r\n]", $_message)) return 1;
    else return 0;
}
```

Uwaga! Znaki: ^ { } [] ~ \ | <enter> liczone są **podwójnie** (w przypadku wysyłania wiadomości bez znaków specjalnych) ze względu na wymogi specyfikacji GSM.

Tabela ilości wysłanych SMS w zależności od długość wiadomości:

Bez znaków specjalnych		Ze znakami specjalnymi (w tym PL)	
Ilość znaków	Ilość punktów	Ilość znaków	Ilość punktów
160	1 SMS	70	1 SMS
306	2 SMSy	134	2 SMSy
459	3 SMSy	201	3 SMSy

Uwaga! Najnowsza specyfikacja techniczna SMSAPI znajduje się w zakładce POMOC na stronie <http://www.smsapi.pl/>

Adresy IP z których wykonywane są odwołania callback to: **46.4.31.8** oraz **62.181.2.52**

Dodatek nr 1 – Lista raportów doręczeń wiadomości

Tabela kodów zwrotu:

Numer	Status	Opis
401	Nieznaleziona	Błędny numer ID lub raport wygasł
402	Przedawniona	Wiadomość niedostarczona z powodu zbyt długiego czasu niedostępności numeru
403	Wysłana	Wiadomość została wysłana ale operator nie zwrócił jeszcze raportu doręczenia
404	Dostarczona	Wiadomość dotarła do odbiorcy
405	Niedostarczona	Wiadomość niedostarczona (np.: błędny numer, numer niedostępny)
406	Nieudana	Błąd podczas wysyłki wiadomości - prosimy zgłosić
408	Nieznany	Brak raportu doręczenia dla wiadomości (wiadomość doręczona lub brak możliwości doręczenia)
409	Kolejka	Wiadomość czeka w kolejce na wysyłkę
410	Zaakceptowana	Wiadomość przyjęta przez operatora
411	Ponawianie	Wykonana była próba połączenia która nie została odebrana, połączenie zostanie ponowione.

Dodatek nr 2 - Kody błędów

Tabela kodów błędów:

ERROR	Opis
8	Błąd w odwołaniu (Prosimy zgłosić)
11	Zbyt długa lub brak wiadomości lub ustawiono parametr nounicode i pojawiły się znaki specjalne w wiadomości. Dla wysyłki VMS błąd oznacz brak pliku WAV lub treści TTS.
12	Wiadomość zawiera ponad 160 znaków (gdy użyty parametr &single=1)
13	Nieprawidłowy numer odbiorcy
14	Nieprawidłowe pole nadawcy
17	Nie można wysłać FLASH ze znakami specjalnymi
18	Nieprawidłowa liczba parametrów
19	Za dużo wiadomości w jednym odwołaniu
20	Nieprawidłowa liczba parametrów IDX
21	Wiadomość MMS ma za duży rozmiar (maksymalnie 100kB)
22	Błędny format SMIL
23	Błąd pobierania pliku dla wiadomości MMS lub VMS
24	Błędny format pobieranego pliku
30	Brak parametru UDH jak podany jest datacoding=bin
31	Błąd konwersji TTS
32	Nie można wysłać wiadomości Eco, MMS i VMS na zagraniczne numery.
33	Brak poprawnych numerów
40	Brak grupy o podanej nazwie
41	Wybrana grupa jest pusta (brak kontaktów w grupie)
50	Nie można zaplanować wysyłki na więcej niż 3 miesiące w przyszłość
51	Ustawiono błędną godzinę wysyłki, wiadomość VMS mogą być wysyłane tylko pomiędzy godzinami 8 a 22
52	Za dużo prób wysyłki wiadomości do jednego numeru (maksymalnie 10 prób w przeciągu 60sek do jednego numeru)
53	Nieunikalny parametr idx. Została już przyjęta wiadomość z identyczną wartością parametru idx przy wykorzystaniu parametru &check_idx=1.
55	Brak numerów stacjonarnych w wysyłce i ustawiony parametr skip_gsm
101	Niepoprawne lub brak danych autoryzacji
102	Nieprawidłowy login lub hasło
103	Brak punktów dla tego użytkownika
104	Brak szablonu
105	Błędny adres IP (włączony filtr IP dla interfejsu API)
200	Nieudana próba wysłania wiadomości, prosimy ponowić odwołanie
201	Wewnętrzny błąd systemu (prosimy zgłosić)
202	Zbyt duża ilość jednoczesnych odwołań do serwisu, wiadomość nie została wysłana (prosimy odwołać się ponownie)
300	Nieprawidłowa wartość pola <i>points</i> (przy użyciu pola <i>points</i> jest wymagana wartość 1)
301	ID wiadomości nie istnieje
400	Nieprawidłowy ID statusu wiadomości
999	Wewnętrzny błąd systemu (prosimy zgłosić)

Dodatek nr 3 - Kodowanie

Domyślnie kodowanie znaków ustawione jest na windows-1250. Jednak do wysyłania wiadomości można użyć jednego z poniżej przedstawionych rodzajów kodowania. W tym celu wykorzystać należy parametr **&encoding**.

- 'iso-8859-1'
- 'iso-8859-2'
- 'iso-8859-3'
- 'iso-8859-4'
- 'iso-8859-5'
- 'iso-8859-7'
- 'windows-1250'
- 'windows-1251'
- 'utf-8'

Przykład: `http://api.smsapi.pl/sms.do?username=uzytkownik&password=haslo&to=4850000000&encoding=utf-8&message=wiadomosc`

Dodatek nr 4 – Przykładowe skrypty

Wysłanie Wiadomości SMS metodą GET

```
<?php

$params = array(
    'username' => 'uzytkownik', //login z konta SMSAPI
    'password' => md5('haslo'), //lub $password="ciąg md5"
    'to' => '48500000000', //numer odbiorcy
    'from' => 'nadawca', //nazwa nadawcy musi być aktywna
    'message' => 'moja wiadomosc', //treść wiadomości
);
if ($username && $password && $to && $message) {
    $data = '?' . http_build_query($params);
    $plik = fopen('http://api.smsapi.pl/sms.do' . $data, 'r');
    $wynik = fread($plik, 1024);
    fclose($plik);
    echo $wynik;
}

?>
```

Wysłanie Wiadomości SMS metodą POST

```
<?php

$username = 'login'; //login z konta SMSAPI
$password = md5('haslo'); //lub $password="ciąg md5"
$to = '5000000000'; //numer odbiorcy
$from = 'SMSAPI'; //nazwa nadawcy musi być aktywna
$message = urlencode("Moja wiadomosc"); //treść wiadomości
$url = 'http://api.smsapi.pl/sms.do';
$c = curl_init();
curl_setopt($c, CURLOPT_URL, $url);
curl_setopt($c, CURLOPT_POST, true);
curl_setopt($c, CURLOPT_POSTFIELDS, 'username=' . $username . '&password=' . $password . '&from=' .
$from . '&to=' . $to . '&message=' . $message);
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
$content = curl_exec($c);
curl_close($c);
echo $content;

?>
```

Przykład skryptu wysyłającego wiadomość MMS

```
<?php

$username = 'login';
$password = md5('haslo');
$to = '48500000000';
$subject = urlencode('SMSAPI');
$smil = urlencode('<smil><head><layout><root-layout height="600" width="425"/> <region
id="Image" top="0" left="0" height="100%" width="100%"
fit="meet"/></layout></head><body><par dur="5000ms"></img></par></body></smil>');

if($username && $password) {
    $data = '?username=' . $username . '&password=' . $password . '&to=' . $to . '&subject=' .
$subject .
        '&smil=' . $smil;
    $plik = fopen('http://api.smsapi.pl/mms.do' . $data, 'r');
    $wynik = fread($plik, 1024);
    fclose($plik);
    echo $wynik;
}

?>
```

Przykładowa wiadomość MMS w formacie SMIL (Wysyłka MMS)

```

<smil>
  <head>
    <layout>
      <root-layout width="160" height="135" />
      <region id="Image" width="160" height="114" left="0" top="0" /> //Rozmiar obrazka
      <region id="Text" width="160" height="20" left="0" top="120" /> //Rozmiar pola tekst
      <region id="Audio" width="106" height="1" left="0" top="120" /> //Rozmiar pola audio
    </layout>
  </head>
  <body>
    <par dur="0s">
       //Adres do obrazka
      <text src="http://adres_do_pliku_txt" region="Text" /> //Adres do pliku tekstowego
      <audio src="http://adres_do_pliku_wav" region="Audio" /> //Adres do pliku audio
    </par>
  </body>
</smil>

```

Przykładowy skrypt wysyłki VMS z wykorzystaniem biblioteki cURL

```

<?php

$path_to_file = dirname(__FILE__).'/file.wav';
$ch = curl_init('http://api.smsapi.pl/vms.do');
curl_setopt_array($ch, array(
    CURLOPT_RETURNTRANSFER => 1,
    CURLOPT_POST => 1,
    CURLOPT_POSTFIELDS => array(
        'username' => 'username',
        'password' => md5('password'),
        'to' => '48500500500',
        'file' => '@'.$path_to_file
    )
));
$response = curl_exec($ch);
echo $response;

?>

```

Odebranie raportu doręczenia (CALLBACK SMS DLR)

```

<?php

// UWAGA! Poniższy przykład jest tylko poglądowy. Nie sprawdza on danych
// wejściowych co jest zalecane dla zmiennych GET: MsgId i status, oraz nie
// sprawdza czy zmienne nie są podane przez osoby trzecie.
//
// W przypadku wysłania z wiadomością parametru 'idx' zostanie on zwrócony $_GET['idx']

if($_GET['MsgId'] && $_GET['status']) {
    mysql_select_db('nazwa_bazy',mysql_connect('localhost','login','haslo'));
    $arIds = explode(',',$_GET['MsgId']);
    $arStatus = explode(',',$_GET['status']);
    $arIdx = explode(',',$_GET['idx']);

    if($arIds) foreach($arIds as $k => $v){
        mysql_query("UPDATE sms SET sms_status = '".$_.$arStatus[$k]."', sms_index = '".$_
            .$arIdx[$k]."' WHERE sms_id='".$_.$v."' LIMIT 1");
    }

    mysql_close();
    echo "OK";
}

?>

```

Przykładowy skrypt odbioru MMS

```
<?php
$IP = $_SERVER['REMOTE_ADDR'];
$received=0;

foreach($_FILES as $plik) {
    if(is_uploaded_file($plik['tmp_name'])) {
        if(move_uploaded_file($plik['tmp_name'],'mms/'.$plik['name'])) {
            $received=1;
        }
    }
}

if($received == 1) {
    echo "OK";
}

?>
```

Przykładowa wiadomość MMS wysłana do skryptu (odbior MMS)

```
POST: Array
(
    [mms_subject] => Testowy MMS
    [mms_from] => 48600000000
    [mms_to] => 48500000000
    [mms_date_recv] => 1256031673
    [mms_date] => 1256031643
)

FILES: Array
(
    [0] => Array
        (
            [name] => 0.JPG
            [type] => image/jpeg
            [tmp_name] => /tmp/phpGXutNv
            [error] => 0
            [size] => 25097
        )

    [1] => Array
        (
            [name] => 1.txt
            [type] => text/plain
            [tmp_name] => /tmp/phpaoM6dU
            [error] => 0
            [size] => 14
        )

    [2] => Array
        (
            [name] => 3.amr
            [type] => audio/AMR
            [tmp_name] => /tmp/php6I0KEi
            [error] => 0
            [size] => 27078
        )
)
```