

DOKUMENTACJA PROTOKOŁU WEBSERVICES/SOAP

Platforma SMeSKom - instrukcja korzystania z interfejsu
Webservices v1.0

Autor smeskom@smeskom.pl
Data 06.09.2011
Wersja 1.0 (rev. 1)

SOFTIKA
ROZWIĄZANIA NA MIARĘ POTRZEB



Spis treści

Dokumentacja protokołu Webservices/Soap.....	2
1 Zawartość dokumentu.....	3
2 Ogólne informacje o interfejsie	3
2.1 Zastosowanie.....	3
2.2 Wersja protokołu	3
2.3 Połączenie	3
2.3.1 WSDL.....	3
2.3.2 Autoryzacja.....	4
3 Specyfikacja protokołu Soap.....	4
3.1 Informacje ogólne.....	4
3.1.1 Parametry typu data/czas.....	4
3.1.2 ApiResult.....	4
3.2 Komendy	4
3.2.1 Wysłanie SMS-a (SendSms).....	4
3.2.2 Sprawdzanie stanu wysłanego SMS-a (SmsStatus).....	6
3.2.3 Odwołanie zaplanowanego SMS-a RevokeSms.....	7
3.2.4 Odebranie SMS -a (ReceiveSms)	7
3.2.5 Zaznaczanie odebranych SMS-ów jako przeczytane (MarkRead).....	9

1 Zawartość dokumentu

Niniejszy dokument zawiera informacje na temat sposobu komunikowania się z platformą SMeSKom korzystając z interfejsu zbudowanego w architekturze Webservices/Soap.

2 Ogólne informacje o interfejsie

2.1 Zastosowanie

Opisany w dokumencie interfejs służy do wysyłania i odbierania SMS-ów za pomocą platformy SMeSKom. Protokół ten ma zastosowanie jedynie dla platformy SMeSKom (<http://www.smeskom.pl>) i stanowi jej część składową.

2.2 Wersja protokołu

Zestaw komend opisanych w niniejszym dokumencie jest stały dla danej wersji protokołu. Następne wersje protokołu nie będą musiały wspierać poprzedniej wersji protokołu. Rozszerzania protokołu o nowe wersje nie będą miały wpływu na wersje poprzednie o ile ich wsparcie nie wygaśnie, o czym odpowiednio wcześniej klient zostanie poinformowany.

Niniejszy dokument opisuje wersję 1.0 protokołu. Wersja ta została zaprojektowana zbudowana na bazie interfejsu SmeSx (implementuje jej wszystkie funkcje)

2.3 Połączenie

Połączenie za pomocą SOAP może nastąpić poprzez port 80 (HTTP) lub 443 (HTTPS).

2.3.1 WSDL

Połączenie zwykłe na porcie 80:

<http://www.smeskom.pl/soap/wSDL>

lub połączenie szyfrowane na porcie 443:

<https://www.smeskom.pl/soap/wSDL>

2.3.2 Autoryzacja

Do autoryzacji wywołań należy użyć parametrów połączeniowych interfejsu HTTPS (do sprawdzenia w panelu administracyjnym - użytkownik `htguserNNNN`). Interfejs HTTPS musi być aktywny.

3 Specyfikacja protokołu Soap

3.1 Informacje ogólne

Wszystkie zaimplementowane funkcje mają podobny schemat parametrów wejściowych oraz wyjściowych.

Pierwsze 3 parametry: *login,password* oraz *version* identyfikują parametry połączenia. Następne parametry są potrzebne dla poprawnego wywołania danej funkcji (o ile występują).

Każde wywołanie funkcji SOAP kończy się zwrotem struktury, która składa się z elementu typu *ApiResponse*, zawierający informacje o poprawności wykonania funkcji oraz elementów

Parametry *login,password* oraz *version*:

- **login, password** – parametry autoryzacyjne dla interfejsu HTTPS (login typu: `htguserNNNN`)
- **version** - należy podać wartość **1.0**

3.1.1 Parametry typu data/czas

Wszystkie parametry typu daty zapisywane są i przesyłane poprzez pole typu string w formacie 'YYYY-MM-DD HH24:MI:SS'

3.1.2 ApiResponse

Wartości przekazywane poprzez ApiResponse:

- **execution_status** – success/failed, przy wystąpieniu błędu następne parametry zawierają więcej szczegółów
- **fail_code** - kod błędu
- **fail_description** – opis błędu

3.2 Komendy

3.2.1 Wysłanie SMS-a (SendSms)

Komenda *SendSms* służy do zainicjowania wysłania SMSa. Funkcja nie czeka na wysłanie SMS-a. Odpowiedź z serwera przychodzi w najkrótszym możliwym

czasie. Informacja o tym, czy i kiedy SMS został wysłany lub odebrany przez telefon na jaki SMS został nadany, dostępna jest między innymi za pomocą funkcji *SmsStatus*. Funkcja zwraca wewnętrzny, unikalny dla platformy identyfikator SMS-a. Identyfikator ten używany jest przy wykonywaniu komendy *SmsStatus*.

- **msisdn** - numer MSISDN. Platforma SmeSKom powinna zaakceptować większość możliwych formatów o ile numer jest napisany jednoznacznie. Przykładowe formaty, które zostaną zaakceptowane:
 - 601 234 567
 - +48 601-234-567
 - (0) 601234567
- **body** - treść SMS-a. Maksymalna liczba znaków jaką można przesłać jest równa 160 (dla wiadomości jednoczęściowych) lub 459 (dla wiadomości wieloczęściowych - wysłane zostaną 3 SMS-y) . Jeśli w tekście znajdują się polskie znaki i chcecie Państwo je przesłać, nie wolno przekroczyć 70 znaków (201 - dla wieloczęściowych). Jeśli długość tekstu przekracza te wartości, polskie znaki zostaną automatycznie zamienione na ich odpowiedniki bez ogonków. Więcej na ten temat dowiedzieć się można na naszych stronach („Reguły wysyłania SMS-ów”). Domyślne traktowanie zbyt długich wiadomości można zmienić za pomocą parametrów usługi, które są opisane w panelu administracyjnym.
- **expire_at** - data wygaśnięcia ważności. Platforma nie pozwoli na wysłanie SMS po tym terminie. Jeśli parametr nie zostanie podany czas wygaśnięcia zostanie ustawiony na 3 dni do przodu.
- **sender** - Ustawia inne niż domyślne pole nadawcy dla SMS-ów w usłudze typu "Powiadomienia". Pominięcie pola spowoduje użycie domyślnego pola nadawcy.
- **sms_type** - typ SMS-a (zwykajny, lub wyskakujący SMS, tzw. „flash sms”). Kody typów:
 - n - zwykły SMS - wartość domyślna
 - f - wyskakujący SMS („flash sms”)
- **send_after** - data planowanej wysyłki SMS. Platforma wyśle SMS-a o wyznaczonej godzinie niezależnie od ustawień limitów w Panelu administratora. Jeśli parametr nie zostanie podany SMS zostanie wysłany od razu o ile nie są ustawione w panelu limity na okres wysyłki
- **user_id** - dowolna wartość numeryczna (np. unikalny numer w bazie klienta). Pole to może służyć do połączenia instancji SMS-a w bazie klienta z instancją w bazie platformy SMeSKom

Odpowiedź.

W odpowiedzi funkcja zwraca tabelę z jednym elementem typu **SentSms**. Struktura ta zawiera dwie wartości: **expire_at** oraz **id**.

- **id** – unikalny identyfikator SMS-a w bazie serwisu SMeSKom
- **expire_at** – data wygaśnięcia ważności SMS-a

3.2.2 Sprawdzanie stanu wysłanego SMS-a (SmsStatus)

Funkcja *SmsStatus* pozwala sprawdzić aktualny stan wysyłki jednego lub kilku SMS-ów. Za pomocą jednego wywołania komendy można sprawdzić do 10-u SMS-ów jednocześnie.

Wywołanie funkcji *SmsStatus* bez identyfikatorów spowoduje odczytanie zmian w statusach SMS-ów, ale tylko tych, których przy pomocy tej komendy jeszcze nie odczytaliśmy. Zasięg wyszukiwania zmian w statusach SMS-ów to ostatnie 24 godziny. W wyniku otrzymamy listę SMS-ów (maksymalnie 10), których statusy zmieniły się od czasu ostatniego wysłania tej komendy jako pustej. Tak więc wysyłając cyklicznie tą komendę mamy pewność, że wszystkie zmiany w statusach zostaną zauważone i przeczytane przez nas.

Elementy opcjonalne:

- **id** – lista wewnętrznych identyfikatorów SMS-ów nadanych przy wysyłce (funkcji *SendSms*). Maksymalnie można zawrzeć 10 elementów. Jeśli nie podamy żadnego elementu zostaną zwrócone nieodczytane statusy od czasu ostatniego wywołania *SmsStatus* z pustą listą identyfikatorów.

Odpowiedź

Opis parametrów zawartych w strukturze *SmsStatus*

- **id** – wewnętrzny identyfikator SMS-a wychodzącego
- **status** – status wiadomości. Wartość elementu może być jedna z:
 - **new** – nowy SMS, status **new** zostaje ustawiony przy wysłaniu komendy `<send_sms>` (nadaniu wiadomości SMS)
 - **processed** – system rozpoczął przetwarzanie wiadomości, wiadomość została przeznaczona do wysyłki
 - **sent** – wiadomość została poprawnie wysłana przez system
 - **delivered** – wiadomość została odebrana przez adresata (przyszedł tzw. „raport doręczenia”)
 - **failed** – nie udało się wysłać wiadomości. Informacja o błędzie została zawarta w elemencie `<fail_code>`
- **sent_at** – czas wysłania wiadomości w formacie „yyyy-mm-dd hh:mi:ss” (np.: „2007-11-10 14:20:00”). Element ten nie zawiera żadnych znaków, jeśli wiadomość nie została jeszcze wysłana.
- **delivered_at** – czas odebrania wiadomości przez adresata w formacie „yyyy-mm-dd hh:mi:ss” (np.: „2007-11-10 14:20:00”). Element ten nie zawiera żadnych znaków, jeśli wiadomość nie została jeszcze odebrana (SMS nie dotarł lub nie mógł być odebrany przez

odbiorcę).

- **fail_code** – kod błędu, wartość całkowita, numeryczna. Wartość 0 jest wpisana, jeśli SMS został wysłany poprawnie lub, gdy jeszcze nie doszło do wysłania.

3.2.2.1 Optymalne sprawdzanie zmian statusów SMS-ów

Zalecana metoda odczytywania/synchronizacji statusów z własnym systemem to wykorzystanie funkcji `SmsStatus` z pustą listą identyfikatorów jako parametr wejściowy. Metoda ta polega na sprawdzeniu SMS-ów, których status zmienił się od czasu ostatniego odpytywania o zmiany w statusach. Najlepiej, aby wywołania funkcji nie były zbyt częste (z uwagi na niepotrzebne obciążenie serwisu). Optymalna częstotliwość to wywołania zależne od okresu od ostatniego wysłania SMSa (jednym słowem odpytujemy o statusy częściej, jeśli jest szansa, że takie statusy się pojawią).

3.2.3 Odwołanie zaplanowanego SMS-a `RevokeSms`

Do odwołania SMS-a, którego zaplanowaliśmy na czas przyszły użyć należy komendy `RevokeSms`. Składnia komendy jest analogiczna do komendy sprawdzającej status. Możliwe jest jednoczesne odwołanie do 10 SMS-ów, a zwrot z komendy pokazuje, czy udało się rozpocząć procedurę odwołania, czy nie.

Parametry w strukturze odpowiedzi:

- **id** – wewnętrzny identyfikator SMS-a wychodzącego
- **revoke_status** – status polecenia odwołania SMS-a. Wartość elementu może być jedna z:
 - `revoke requested` – odwołanie SMS-a zostanie przeprowadzone w ciągu najbliższych kilku sekund. Do sprawdzenia czy odwołanie powiodło się, należy użyć komendy „`sms_status`”. Zmiana statusu na „`failed`” i ustawienie „`fail_code`” na 410 oznacza poprawne zakończenie procedury odwołania. Inny `fail_code` może oznaczać niemożność odwołania. Patrz kody błędów.
 - `revoke impossible` – SMS został już przekazany dalej do wysyłki – odwołanie nie jest możliwe.

3.2.4 Odebranie SMS -a (`ReceiveSms`)

Funkcja próbuje odczytać SMS-a przychodzącego (SMS-a wysłanego do serwisu z telefonu komórkowego). Jeśli zostały spełnione warunki wskazane w wywołaniu komendy zwracany jest SMS, jeśli nie odpowiedź nie zawiera SMS-a.

Komenda pracuje w dwóch trybach (tryb ustawia się za pomocą elementu `type`).

Tryb 1. (*type = unread*) Odbieranie wiadomości, która nie została jeszcze oznaczona jako przeczytana.

Każda nowa wiadomość, która przyjdzie do serwisu jest oznaczona jako „nieprzeczytana”. Można ją zaznaczyć jako przeczytaną za pomocą:

- *funkcji MarkRead*. Patrz punkt 3.2.5, aby dowiedzieć się więcej.
- ustawiając parametr *mark* w poleceniu *ReceiveSms* wartością „*true*” (jest to również domyślna wartość dla tego parametru)

W tym trybie system próbuje dostarczyć SMS, który nie został jeszcze oznaczony jako „przeczytany”

Tryb 2. (*type = time*) Odbieranie dowolnych wiadomości w zadanym czasie.

W tym trybie system nie zwraca uwagi na oznaczenie SMS-a (czy został oznaczony jako przeczytany, czy nie). Zwraca najstarszy SMS w zadanym przez parametry czasie.

Uwaga: W obu trybach bardzo istotny jest parametr *after_id* w wywołaniu polecenia. Wartość parametru *after_id* może zawierać identyfikator SMS-a, który został odebrany przez system tuż przed SMS-em, który chcemy odczytać. Np.: jeśli próbujemy odczytać SMSy odebrane przez system pomiędzy godziną 13-ą, a 15-ą danego dnia i przyszło ich kilka w tym okresie należy:

- przy pierwszym wywołaniu *ReceiveSms* pominąć parametr *after_id*
- wywoływać *ReceiveSms* dotąd, aż nie zwróci żadnego SMS-a. Za każdym razem parametr *after_id* ustawiamy wartością identyfikatora SMS-a odebranego w poprzednim wykonaniu polecenia *ReceiveSms*

Elementy obowiązkowe:

- **type** – wybór trybu odczytu SMS-ów przychodzących. Ten parametr może mieć dwie wartości: „*time*”, „*unread*”. Znaczenie tych wartości i działania poszczególnych trybów zostało opisane w punkcie 3.2.4.
- **start_time** - parametr obowiązkowy jedynie w trybie „*time*”. Wartość przekazana w tym parametrze oznacza początek odpytywanego okresu. Format parametru „*yyyy-mm-dd hh:mi:ss*”.

Elementy opcjonalne:

- **stop_time** – parametr mający sens jedynie dla trybu „*time*”. Oznacza koniec odpytywanego okresu czasu. Format parametru „*yyyy-mm-dd hh:mi:ss*”. Jeśli nie podamy tego parametru – wartość zostanie ustawiona na aktualny czas.
- **mark** – domyślna wartość: „*true*”. Jeśli wartość ta jest ustawiona na „*true*”, odczytany za pomocą wywołania SMS zostanie oznaczony jako „przeczytany”. Jeśli wartość ustawimy na „*false*”, SMS pozostanie „nieprzeczytany”.
- **after_id** – identyfikator SMS, który odczytany został w poprzednim wykonaniu. Parametr ten ma znaczenie jeśli odbieramy wiele SMS-ów w zadanym czasie. Jeśli nie podamy tego parametru, zostanie zwrócony

najstarszy SMS w zadanym czasie.

Parametry struktury w odpowiedzi:

- **contain_sms** – jedna z wartości: „true”, „false”. „true” oznacza, że udało się odczytać SMS-a, a wiadomość zawiera wypełnioną listę *ReceiveSms* ze szczegółami odczytanego SMS-a. Wartość „false” oznacza, że dla zadanych parametrów nie ma odebranych SMS-ów.
- **has_more** – jedna z wartości: „true”, „false”. „true” oznacza, że dla zadanych parametrów są jeszcze SMS-y, które można odczytać. Wartość „false” oznacza, że dla zadanych parametrów nie ma więcej odebranych SMS-ów w systemie.

Struktura *ReceiveSms*

- **Id** – wewnętrzny identyfikator SMS-a w systemie.
- **msisdn** – numer MSISDN telefonu, który przysłał wiadomość do serwisu.
- **body** – treść odebranej wiadomości
- **inserted_at** – czas nadejścia SMS-a do systemu. Czas w formacie „yyyy-mm-dd hh:mi:ss” (np.: „2007-11-10 14:20:00”).
- **received_at** – czas odebrania wiadomości przez modem. Zazwyczaj ten czas jest od kilku do kilkudziesięciu sekund starszy od elementu *inserted_at*. Element w formacie „yyyy-mm-dd hh:mi:ss”.
- **marked_at** – czas oznaczenia wiadomości jako „przeczytanej”. Jeśli pole to jest puste, oznacza to, że wiadomość nie została jeszcze oznaczona jako „przeczytana”.

3.2.4.1 Opis procedury odbierania SMSów

Ciągłe odczytywanie wiadomości przychodzących może odbywać się za pomocą cyklicznie wykonywania funkcji *ReceiveSms* w trybie **unread** (*type=unread*).

Jednoczesne ustawienie parametru *type* na wartość „unread” oraz pozostawienie bez zmian parametru *mark* (domyślna wartość „true”) powodują, że jeśli w systemie oczekuje na odczytanie SMS przychodzący, zostanie on zwrócony i jednocześnie zaznaczony jako „przeczytany”. Dzięki temu ponowne wykonanie tej samej komendy spowoduje zwrócenie następnego nieprzeczytanego SMS-a.

3.2.5 Zaznaczanie odebranych SMS-ów jako przeczytane (MarkRead)

MarkRead jest pomocniczym poleceniem w procesie odbierania SMS-ów. Komenda ta pozwala na zaznaczenie wiadomości odebranej jako przeczytana przez użytkownika.

Dwuetapowe odczytywanie SMS-ów:

- etap 1. odczytanie SMS-a z parametrem `mark = „false”` (Funkcja `ReceiveSms`)
- etap 2. zaznaczenie SMS-a jako „przeczytany” (Funkcja `MarkRead`)

pozwała na zwiększenie bezpieczeństwa odczytu SMS-ów.

Takie odczytywanie SMS-ów nie pozwoli na hipotetyczną sytuację, gdy do aplikacji klienckiej nie dojdzie wiadomość zwrotna (np. z powodu problemów z siecią)

Elementy opcjonalne: brak

- `id` – wewnętrzny identyfikator SMS-a, nadany w momencie odebrania SMS-a i przekazywany przez polecenie `ReceiveSms`. Elementów `id` można załączyć jeden lub kilka. Maksymalnie można zawrzeć 10 elementów `id` w jednej wiadomości.

Elementy opcjonalne: brak

Parametry w strukturze odpowiedzi:

- `id` – wewnętrzny identyfikator SMS-a przychodzącego
- `marked_at` – data oznaczenia SMS-a. Jeśli SMS został oznaczony wcześniej, SMS nie zostanie ponownie zaznaczony. Data w tym polu jest zawsze datą pierwszego oznaczenia.