

Dokumentacja SOAP



Wprowadzenie	2
Komunikaty ogólne	3
Wysyłanie wiadomości SMS o jednakowej treści	7
Wysyłanie spersonalizowanych wiadomości SMS	13
Przykład integracji w C#	16

Wprowadzenie

Usługa SOAP polega na skonfigurowaniu połączenia HTTP w którym struktura danych określona jest w formacie XML. Wiadomość ta zawiera następujące elementy (tagi):

- envelope, zawiera treść wiadomości,
- header, element opcjonalny, zawiera informacje nagłówkowe,
- body, zawiera zapytanie i odpowiedź.

Aktualnie ta usługa nie jest już wspierana przez jej twórców, dlatego polecamy Naszym Klientom aby do połączenia z Naszą Platformą wykorzystywali inne dostępne interfejsy.

Komunikaty ogólne

W przypadku braku określenia akcji, zablokowanego konta klienta lub podaniu błędnych danych klienta, SerwerSMS.pl wygeneruje dokument w formacie JSON lub XML z informacją na temat błędu, który wystąpił. Przykładowo, w przypadku podania niewłaściwego loginu lub hasła klient zobaczy następujący komunikat:

```
{
  "error":
  {
    "code":1001,
    "type":"InvalidUser",
    "message":"Nieprawidłowy login lub hasło"
  }
}
```

Jeśli nie zostanie określona żadna akcja do wykonania, system wyświetli komunikat:

```
{
  "error":
  {
    "code":1000,
    "type":"InvalidAction",
    "message":"Nie określono akcji do wykonania"
  }
}
```

Podgląd komunikatu błędu można sprawdzić za pomocą akcji:

error/[code]

przykład:

<https://api2.serwersms.pl/error/1000>

odpowieź:

```
{
  "error":
  {
    "code":1000,
    "type":"InvalidAction",
    "message":"Nie określono akcji do wykonania"
  }
}
```

Lista kodów błędów zwracanych przez API

Poniżej znajduje się tabela z kodami oraz przyporządkowanymi do nich opisami tekstowymi.

Kod	Typ	Opis
Ogólne		
1000	InvalidAction	Nie określono akcji do wykonania
1001	InvalidUser	Nieprawidłowy login lub hasło
1002	InvalidRole	Użytkownik nie ma uprawnień do korzystania z API
1003	InvalidIP	Nieautoryzowany adres IP

1004	InvalidID	Nieprawidłowy parametr ID
1005	InvalidRole	Brak uprawnień
1006	InvalidData	Niepoprawny format danych
1007	InvalidRole	Konto nie jest w pełni aktywne
1008	InvalidRole	Użytkownik nie ma uprawnień do podglądu numerów
2000	ValidationRequiredError	Brak wymaganych parametrów
2001	ValidationPhoneError	Nieprawidłowy numer telefonu
2002	ValidationCodeError	Nieprawidłowy kod błędu
Wiadomości		
3000	SendError	Wyczerpano limit dostępnych wiadomości SMS ECO+
3001	SendError	Brak uprawnień do wysyłania wiadomości SMS ECO+
3002	SendError	Wyczerpano limit dostępnych wiadomości SMS FULL
3003	SendError	Brak uprawnień do wysyłania wiadomości SMS FULL
3004	SendError	Wyczerpano limit dostępnych wiadomości MMS
3005	SendError	Brak uprawnień do wysyłania wiadomości MMS
3006	SendError	Wyczerpano limit dostępnych wiadomości VOICE
3007	SendError	Brak uprawnień do wysyłania wiadomości VOICE
3008	SendError	Brak uprawnień do wysyłania wiadomości typu Flash SMS
3009	SendError	Brak uprawnień do wysyłania wiadomości SMS
3100	SendError	Wysyłanie wiadomości SMS zostało zablokowane
3101	SendError	Wiadomość jest pusta
3102	SendError	Wiadomość przekroczyła dozwoloną ilość znaków
3103	SendError	Błędny numer odbiorcy
3104	SendError	Nie podano numerów telefonów
3105	SendError	Niepoprawna treść wiadomości
3106	SendError	Nie można użyć tej nazwy nadawcy
3107	SendError	Nie posiadasz uprawnień do wysyłania wiadomości spersonalizowanych
3108	SendError	Nieprawidłowa godzina wysyłki
3109	SendError	Nieprawidłowy plik
3110	SendError	Zbyt duża ilość wiadomości do przesłania, maksymalna ilość to 100 000 SMS w jednym zgłoszeniu
3111	SendError	Błąd unikalności

3200	SendError	Niepoprawne znaki w unique_id
3201	SendError	Przestane unique_id nie są unikalne
3202	SendError	Niezgodna ilość spersonalizowanych i unique_id
3330	MessageError	Brak wiadomości
Pliki		
4000	FileError	Plik jest zbyt duży
4001	FileError	Plik nie istnieje
4002	FileInvalid	Nieprawidłowy plik
Numer telefonu/HLR		
4100	PhoneError	Błąd połączenia
4101	PhoneError	Niepoprawny numer
4102	PhoneError	Brak uprawnień do HLR
4103	PhoneError	Wyczerpano limit dostępnych zapytań HLR
4104	PhoneError	Blokada konta
4105	PhoneError	Trwa oczekiwanie na odpowiedź
4106	PhoneError	Nieprawidłowy identyfikator
Premium		
4200	PremiumError	Błąd wysyłki
4201	PremiumError	Nieprawidłowe dane
Konto		
4300	AccountError	Brak loginu
4301	AccountError	Brak hasła
4302	AccountError	Brak numeru telefonu
4303	AccountError	Brak adresu e-mail
4304	AccountError	Brak imienia
4305	AccountError	Brak nazwiska
4306	AccountError	Brak nazwy firmy
4307	AccountError	Nieprawidłowy numer
4308	AccountError	Nieprawidłowy adres e-mail
4309	AccountError	Zbyt krótki login
4310	AccountError	Nieprawidłowy login
4311	AccountError	Nieprawidłowa firma

4312	AccountError	Login istnieje
4313	AccountError	Błąd rejestracji
4320	AccountError	Nie podano aktualnego hasła
4321	AccountError	Aktualne hasło jest nieprawidłowe
4322	AccountError	Nie podano nowego hasła
4323	AccountError	Wprowadzone nowe hasła nie są identyczne
4324	AccountError	Nie podano loginu
4325	AccountError	Nieprawidłowy login
Nazwy nadawcy		
4400	SenderExists	Nazwa nadawcy już istnieje
4401	SenderError	Brak możliwości dodania numeru jako nazwę nadawcy
4402	SenderInvalid	Nieprawidłowa nazwa nadawcy
4403	SenderError	Nieprawidłowy numer
4404	SenderError	Nieprawidłowy kod lub numer
4405	SenderError	Numer był już aktywny
Subkonta		
4500	SubaccountError	Brak uprawnień do działu użytkownicy
4501	SubaccountError	Subkonto nie może tworzyć subkont
4502	SubaccountError	Login zajęty
4503	SubaccountError	Brak szablonu uprawnień
4504	SubaccountError	Nieprawidłowy login lub hasło
Czarna lista		
4600	BlacklistError	Brak operacji do wykonania
4601	BlacklistPhoneExists	Numer istnieje już na liście
4602	BlacklistError	Nie znaleziono podanego numeru

Wysyłanie wiadomości SMS o jednakowej treści

Wywołanie adresu

Aby przy pomocy Zdalnej obsługi wysłać wiadomość SMS należy wykorzystać poniższy kod i wysłać odpowiednie parametry.

```
{  
  
require_once('lib/nusoap.php');  
  
try {  
  
    $wsdl_url = 'https://soap.serwersms.pl/server.php?wsdl';  
  
    $client = new nusoap_client($wsdl_url, 'wsdl');  
    $client->setCredentials("demo", "demo", "basic");  
    $result = $client->call('Messages.sendSms',  
        array( ' ', 'test', 'test', array( 'test' => 1, 'details' => 1 ) ));  
  
    var_dump($client->responseData,$result,$client->getError());  
    exit;  
  
} catch(Exception $e) { var_dump($e);exit; } }  
}
```

UWAGA! Aby wysłać wiadomość SMS FULL należy wypełnić pole nadawca (nazwa alfanumeryczna lub numer). Jeśli parametr nadawca zostanie pominięty, lub jego wartość będzie pusta, zostanie wysłany SMS typu ECO. Wypełnienie parametru nadawca decyduje o tym czy wysłany zostanie SMS ECO+ czy FULL.

Dostępne parametry

Parametr	Typ	Przykładowa wartość lub format	Opis
username	String	login	Login konta używanego do wysyłki.
password	String	hasło	Hasło do konta.
phone	String Array	+48500600700	Numer lub tablica numerów telefonów.
text	String	tresc	Treść wiadomości.
flash	Boolean	true, false lub brak	Parametr opcjonalny.

test	Boolean	true, false lub brak	Parametr opcjonalny, pozwalający na sprawdzenie zapytania wysyłającego wiadomość SMS. W odpowiedzi generowany jest dokument JSON/XML identyczny jak w przypadku standardowego zapytania jednak wiadomość nie jest wysyłana. Przydatne do testowania aplikacji (debug).
sender	String	Alfanumeryczna nazwa nadawcy np. „INFORMACJA”, własna nazwa lub numer 4 lub 9-cio cyfrowy.	Parametr opcjonalny, umożliwia zmianę pola nadawcy wiadomości SMS na niemal dowolną nazwę lub numer. Każda nazwa nadawcy musi zostać najpierw dodana w Panelu lub przez API, a następnie zatwierdzona przez administratora SerwerSMS.pl.
wap_push	String	Adres URL lub brak	Parametr opcjonalny.
utf	Boolean	true, false lub brak	Parametr opcjonalny, umożliwia wysłanie wiadomości FULL SMS zawierającej np. polskie znaki diakrytyczne, cyrylicę itp. Opcja dostępna wyłącznie w SMS FULL. Przy użyciu tej opcji zmienia się maksymalna długość pojedynczej wiadomości SMS z 160 do 70 znaków.
details	Boolean	true, false lub brak	Parametr wyświetlający w odpowiedzi zwrotnej szczegóły wysłanych wiadomości.
vcard	Boolean	true, false lub brak	Parametr opcjonalny, umożliwiający wysłanie wiadomości VCARD.
speed	Boolean	1, 0 lub brak	Parametr opcjonalny, pozwalający na wysłanie wiadomości niezależnym kanałem o najwyższej jakości (dodatkowa opłata).
date	DateTime	ISO np. „2015-02-22 12:25:55”	Parametr opcjonalny, pozwalający na określenie terminu wysyłki wiadomości.

unique_id	String Array	np. 6asTD3fif98gj	Parametr opcjonalny, pozwalający na zdefiniowanie własnego identyfikatora wysyłanej wiadomości. Identyfikator może mieć minimalnie 3 znaki i maksymalnie 50 znaków alfanumerycznych (a-z, A-Z, 0-9). Dla grupowych wysyłek, kolejne unique_id muszą być unikalne oraz ilość unique_id musi być zgodna z ilością numerów.
group_id	String Array	np. 123456789	Identyfikator lub identyfikatory grup w Panelu Klienta. Identyfikatory te można pobrać korzystając z akcji groups/index lub kopiując je z poziomu edycji grupy w Panelu Klienta.
contact_id	String Array	np. 123456789	Identyfikator lub identyfikatory kontaktów w Panelu Klienta. Identyfikatory te można pobrać korzystając z akcji contacts/index.
dlr_url	String	http://www.twojadres.pl/skrypt.php?smsid=#SMSID#&stan=#STAN#&data=#DATA#	Parametr pozwala na przekazywanie metodą Push raportów doręczenia na URL Klienta. Ustawienie tej opcji w zapytaniu API nadpisuje ew. ustawienia w Panelu Klienta. Adres powinien być przekazany w formie zakodowanej przez urlencode(). Adres do przekazania raportów obowiązuje dla wszystkich wiadomości przekazanych do realizacji w ramach jednego zapytania. Dostępne parametry: #SMSID# - identyfikator wiadomości #STAN# - stan doręczenia #DATA# - data zmiany statusu #PRZYCZYNA# - ew. przyczyna niedoręczenia wiadomości

Parametry oznaczone pogrubieniem są obowiązkowe. Pozostałe są opcjonalne.

W przypadku chęci wysłania wiadomości jako wizytówka Vcard (parametr „vcard=true”), struktura wiadomości musi mieć ściśle określony format. Przykład formatu Vcard można znaleźć w ogólnie dostępnej dokumentacji w internecie lub korzystając z Panelu Klienta i formularza wysyłki SMS. Kolejne tagi w wiadomości Vcard powinny być rozdzielone znakiem nowej linii.

Wiadomość SMS FULL, która zawiera polskie znaki specjalne powinna być przed wysłaniem odpowiednio zakodowana w UTF-8. Aby wysłać wiadomość z polskimi znakami musi być ustawiony parametr „utf=true”, abonent musi posiadać uprawnienia do wysyłania wiadomości typu SMS FULL oraz musi być wypełnione pole nadawcy (to oznacza, iż wysłany będzie SMS FULL).

Zwrot odpowiedzi

W zależności od przesłanych danych SerwerSMS.pl wygeneruje w odpowiedni dokument w formacie JSON/XML z informacją na temat wykonanych akcji. I tak w przypadku prawidłowego wysłania wiadomości SMS klient otrzyma przykładowo następującą informację:

```
{
  "success":true,
  "queued":1,
  "unsent":0
}
```

W przypadku podania dodatkowego parametru details=true, odpowiedź zwrotna zostanie uzupełniona o szczegóły wysyłanych wiadomości, które można zapisać w bazie danych po stronie oprogramowania klienta:

```
{
  "success":true,
  "queued":1,
  "unsent":0,
  "items":[{
    "id":"1c142d81c7",
    "phone":"+48500600700",
    "status":"queued",
    "queued":"2014-10-16 16:49:05",
    "parts":1,
    "text":"Test SerwerSMS.pl"
  }]
}
```

Parametr "success" zawiera informację o powodzeniu przeprowadzonej operacji. W atrybutach "queued" oraz "unset" znajdują się liczby skolejkowanych oraz niewysłanych wiadomości. Sekcja "items" zawiera numery telefonów i ID wiadomości przekazanych do wysłania (oraz wiadomości, których nie skolejkowano z określonego powodu). Unikalny znacznik wiadomości SMS może być wykorzystany później do sprawdzenia w sposób zdalny stanu wysyłki konkretnej wiadomości SMS. W parametrze "text" widnieje treść wysyłanej wiadomości SMS. Numer telefonu jest automatycznie poprawiany i wyświetlany w pełnym formacie wymaganym przez SerwerSMS.pl czyli z numerem kierunkowym kraju (np. +48) na początku.

W przypadku podania np. złego numeru telefonu zostanie wygenerowana informacja o błędzie jak również ID wiadomości (aby można było później sprawdzić kiedy i dlaczego nie została wysłana). I tak np. w przypadku podania dwóch numerów prawidłowych i dwóch numerów błędnych system wygeneruje następujący dokument JSON:

```
{
  "success":true,
  "queued":2,
  "unsent":2,
  "items":[{
    "id":"32039da8e9",
    "phone":"+48500600700",
    "status":"queued",
    "queued":"2014-10-20 12:32:52",
    "parts":1,
    "text":"Test platformy SerwerSMS.pl"
  }],
}
```

```
{
  "id":"844c2b4af0",
  "phone":"+48500600500",
  "status":"queued",
  "queued":"2014-10-20 12:32:52",
  "parts":1,
  "text":"Test platformy SerwerSMS.pl"
},
{
  "id":"6e373d7856",
  "phone":"45616",
  "status":"unsent",
  "queued":"2014-10-20 12:32:52",
  "error_code":3103,
  "error_message":"Błędny numer odbiorcy",
  "text":"Test platformy SerwerSMS.pl"
},
{
  "id":"bf069fe2c1",
  "phone":"7799123",
  "status":"unsent",
  "queued":"2014-10-20 12:32:52",
  "error_code":3103,
  "error_message":"Błędny numer odbiorcy",
  "text":"Test platformy SerwerSMS.pl"
}
}]
}
```

Oprócz tego może zostać wygenerowany błąd ogólny gdzie nie ma rozgraniczenia na skolejkowane i błędne. Może to nastąpić np. w sytuacji gdy klient nie zdefiniuje treści wiadomości, nie poda numerów telefonów, jego konto nie jest aktywne lub wystąpił inny problem opisany w komunikatach błędów. W przypadku niewpisania treści wiadomości zostanie wygenerowany następujący komunikat:

```
{
  "error":{
    "code":3101,
    "type":"SendError",
    "message":"Wiadomość jest pusta"
  }
}
```

oraz analogicznie w przypadku niepodania numerów telefonów:

```
{
  "error":{
    "code":3104,
    "type":"SendError",
    "message":"Nie podano numerów telefonów"
  }
}
```

Zalecane ustawienia

W przypadku średnich i dużych ilości wysyłanych wiadomości rzędu kilku tysięcy lub więcej, zalecane jest przekazywanie wiadomości w „paczkach” po ok 50-200 numerów w jednym zapytaniu. Przyspieszy to znacznie proces przekazywania danych

do SerwerSMS.pl i zmniejszy ilość koniecznych do wysłania zapytań.

W przypadku próby wysyłki wiadomości na dwa lub więcej takie same numery telefonów w jednym zapytaniu, system wyśle wiadomość tylko raz na ten numer.

Wysyłanie spersonalizowanych wiadomości SMS

Wywołanie adresu

Aby przy pomocy zdalnej obsługi wysłać spersonalizowaną wiadomość SMS należy wywołać określony adres URL metodą POST. W wiadomościach spersonalizowanych nie ma dostępnych parametrów takich jak „phone” czy „text”. Zamiast tego jest dodatkowy parametr „messages” który zawiera zbiór numerów oraz przyporządkowanych do nich indywidualnych treści wiadomości. Wartość parametru „messages” składa się z tablicy dwuwymiarowej, w której zawarte są klucze "phone" oraz "text". Limit pojedynczego wywołania takiego zapytania to 10 000 numerów i odpowiadającym im treściom wiadomości (zalecane są znacznie mniejsze porcje danych, po ok. 50-200 numerów w jednym zapytaniu). Przykładowe wywołanie wysyłki może wyglądać następująco:

```
{  
  
require_once('lib/nusoap.php');  
  
try {  
  
    $wsdl_url = 'https://soap.serwersms.pl/server.php?wsdl';  
  
    $client = new nusoap_client($wsdl_url, 'wsdl');  
    $client->setCredentials("demo", "demo", "basic");  
    $result = $client->call('Messages.sendPersonalized',  
        array( array(  
            array('phone' => '5000000001','text'=>'test3'),  
            array('phone' => '5000000002','text'=>'test4') ), 'sobek',  
            array( 'test' => 1, 'details' => 1, 'group_id' => '17324179' ) ));  
  
    var_dump($client->responseData,$result,$client->getError());  
    exit;  
  
    } catch(Exception $e) { var_dump($e); exit; } }  
}
```

Wywołanie powyższej akcji wraz z podanymi parametrami (w przypadku podania prawidłowych danych do zalogowania) spowodowałoby wysłanie wiadomości SMS o treści „wiadomosc1” na numer 500600700 oraz drugą wiadomość SMS o treści „wiadomosc2” na numer 500600500.

Dostępne parametry

Parametr	Typ	Przykładowa wartość lub format	Opis
username	String	login	Login konta używanego do wysyłki.
password	String	hasło	Hasło do konta.
messages	Array	[array('phone' => numer, 'text' => wiadomosc), array('phone' => numer, 'text' => wiadomosc)]	Numery telefonów i przypisane do nich treści wiadomości SMS.

details	Boolean	true, false lub brak	Parametr wyświetlający w odpowiedzi zwrotnej szczegóły wysłanych wiadomości.
flash	Boolean	true, false lub brak parametru	Parametr opcjonalny.
test	Boolean	true, false lub brak parametru	Parametr opcjonalny, pozwalający na sprawdzenie zapytania wysyłającego wiadomość SMS. W odpowiedzi generowany jest dokument JSON/XML identyczny jak w przypadku standardowego zapytania jednak wiadomość nie jest wysyłana. Przydatne do testowania aplikacji (debug).
sender	String	Numer w formacie +48500600700 lub nazwa własna (max 11 znaków, duże i małe litery oraz liczby i znak spacji).	Parametr opcjonalny, umożliwia zmianę pola nadawcy wiadomości SMS na dowolny numer lub nazwę. Użyć można wyłącznie nazwy zatwierdzone przez administratorów lub nazwy predefiniowane.
utf	Boolean	true, false lub brak parametru	Parametr opcjonalny, umożliwia wysłanie wiadomości FULL SMS zawierającej m.in. polskie znaki diakrytyczne.
voice	Boolean	true, false lub brak parametru	Parametr opcjonalny, umożliwia wysłanie wiadomości SMS VOICE. Parametr nadawca musi pozostać pusty.
date	DateTime	ISO np: „2010-11-09 15:23”	Parametr opcjonalny pozwalający na określenie terminu wysyłki wiadomości SMS.
unique_id	String Array	np. 6asTD3fif98gj	Parametr opcjonalny pozwalający na zdefiniowanie własnego identyfikatora wysyłanej wiadomości. Identyfikator może mieć minimalnie 3 znaki i maksymalnie 50 znaków alfanumerycznych (a-z, A-Z, 0-9). Dla grupowych wysyłek, kolejne unique_id muszą być unikalne, podane w formie tablicy oraz ilość unique_id musi być zgodna z ilością wiadomości spersonalizowanych.

Parametry oznaczone pogrubieniem są obowiązkowe. Pozostałe są opcjonalne.

Zwrot wygenerowany po przesłaniu jest taki sam jak w przypadku zwykłego wysłania wiadomości SMS.

```
{
  "success":true,
  "queued":2,
  "unsent":2,
  "items":[{"
    "id":"98c93cc5f2",
    "phone":"+48500600700",
    "status":"queued",

```

```

    "queued":"2014-10-20 13:26:50",
    "parts":1,
    "text":"wiadomosc1"
  },
  {
    "id":"a20ff834c8",
    "phone":"+48500600500",
    "status":"queued",
    "queued":"2014-10-20 13:26:50",
    "parts":1,"text":"wiadomosc2"
  },
  {
    "id":"b25462142e",
    "phone":"45648",
    "status":"unsent",
    "queued":"2014-10-20 13:26:50",
    "error_code":3103,
    "error_message":"Błędny numer odbiorcy",
    "text":"wiadomosc3"
  },
  {
    "id":"d862cc6575",
    "phone":"11999885544",
    "status":"unsent",
    "queued":"2014-10-20 13:26:50",
    "error_code":3103,
    "error_message":"Błędny numer odbiorcy",
    "text":"wiadomosc4"
  }
}

```

Zalecane ustawienia

W przypadku średnich i dużych ilości wysyłanych wiadomości rzędu kilku tysięcy lub więcej, zalecane jest przekazywanie wiadomości w „paczkach” po ok 50-200 numerów w jednym zapytaniu. Przyspieszy to znacznie proces przekazywania danych do SerwerSMS.pl i zmniejszy ilość koniecznych do wysłania zapytań.

Zdefiniowanie parametru „messages” nadpisuje wartości parametrów „phone” oraz „text”.

Przykład integracji w C#

Aby móc skorzystać z usługi SOAP należy dodać w swoim projekcie Service References podając jako parametr adres url: <https://soap.serwersms.pl/server.php?wsdl>

Poniżej przedstawiono przykład jak wysłać wiadomość SMS dzięki tej usłudze:

```
{

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web;
using System.ServiceModel;
using System.ServiceModel.Channels;

var service = new namespace.Service.SerwerSMSPlPortTypeClient();
service.ClientCredentials.UserName.UserName= "demo";
service.ClientCredentials.UserName.Password = "demo";

try {

String encoded = System.Convert.ToBase64String(System.Text.Encoding.GetEncoding("UTF-8")
    .GetBytes(service.ClientCredentials.UserName.UserName + ":" + service.ClientCredentials.UserName.Password));

HttpRequestMessageProperty request = new HttpRequestMessageProperty();
request.Headers[System.Net.HttpRequestHeader.Authorization] = "Basic " + encoded;
OperationContextScope clientScope = new OperationContextScope(service.InnerChannel);
OperationContext.Current.OutgoingMessageProperties.Add(HttpRequestMessageProperty.Name, request);

soap.Service.paramsMessageSendSms param = new namespace.Service.paramsMessageSendSms { };
param.details = true;
var response = service.messages_sendSms("5000000002", "text", "sender", param);
Console.WriteLine(response);

} catch(Exception e) { Console.WriteLine("{0} Exception caught.", e); } }

}
```