

Ahmes Sp. z o.o.

ul. Lewicka 13/15

02-547 Warszawa

tel: (22) 113 10 00, fax: (22) 203 63 21, e-mail: biuro@ahmes.pl, <http://www.ahmes.pl>



**DOKUMENTACJA
IMPLEMENTACJI MECHANIZMÓW OBSŁUGI
AHMES SMS (soap) 2013-03-24**



SPIIS TREŚCI

WPROWADZENIE	3
WYSYŁANIE SMS'ów	4
Parametry obowiązkowe do wstawienia	4
Parametry opcjonalne	4
KSIĄŻKA ADRESOWA	5
Parametry	5
Dodanie grupy numerów	5
Usunięcie grupy numerów	5
Dodanie numeru do grupy numerów	6
Usunięcie numeru z grupy numerów	6
Pobranie listy grup numerów	6
Pobranie listy numerów przypisanych do danej grupy	7
RAPORTY SMS	7
Parametry obowiązkowe	8
Parametry do filtrowania	8
Parametry dodatkowe	8
USUWANIE ZAPLANOWANYCH SMSÓW	8
Parametry obowiązkowe	9
Parametr opcjonalny	9



WPROWADZENIE

Dokumentacja pokazuje dostęp do usługi przy pomocy webserwisu SOAP. Do każdego wywołania usługi są potrzebne parametry *userName* i *userPassword* odzwierciedlające dane potrzebne do uwierzytelnienia na witrynie usług:

<http://partners.ahmes.pl>

Adres webserwisu:

<https://partners.ahmes.pl/AhmesSmsService.svc>

WYSYŁANIE SMS'ów

Do wysyłania sms'ów służy metoda SendSms

Parametry obowiązkowe do wstawienia

userName - nazwa klienta

userPassword – hasło zakodowane MD5

message – treść wiadomości

odbiorca:

to – numer telefonu odbiorcy w formacie 48XXXXXXXXXX lub XXXXXXXXXXXX(może być podana tablica numerów w formacie XXXXXXXXXXX,XXXXXXXXX,XXXXXXXXX itd. dla większej ilości odbiorców)
lub

groupName – grupa kontaktów z książki adresowej(opis edycji książki adresowej w punkcie II. Książka adresowa)

Przykład użycia metody w języku C#:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
SendSmsStatus status = new SendSmsStatus();  
client.SendSms("LOGIN", "PASSWORD",  
null, "test", null, "123456789", null, DateTime.Now, null, out status);
```

Parametry opcjonalne

messageType – 1 –wiadomość sms, 0- powiadomienie flash(domyślnie 1)

sendDate - data wysłania wiadomości. Jeżeli termin będzie późniejszy od obecnej daty to zostanie wysłana wiadomość w terminie nie wcześniejszym niż podany w parametrze, jeżeli nie, zostanie wysłana natychmiast.

testMode - pozwala na testowanie komunikacji z interfejsem. W tym trybie testowana jest poprawność przekazanych parametrów oraz zwracane są losowo komunikaty błędów lub poprawności przetworzenia wiadomości, ale nie są one wysyłane do bramy SMS.

KSIĄŻKA ADRESOWA

Do edycji książki adresowej służy metoda `AddressBook`.

Parametry

userName - nazwa klienta

userPassword – hasło zakodowane MD5

groupName – nazwa grupy numerów

operationId – id operacji na książce adresowej

number – numer telefonu(opcjonalnie, w zależności od operacji)

Dodanie grupy numerów

Do dodania grupy kontaktów należy użyć parametru **operationId** z wartością 1 i podać nazwę danej grupy w parametrze **groupName**.

Przykład użycia w języku C#:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<AddressBook> addressBookCollection = new List<AddressBook>();  
List<AddressBookNumber> AddressBookNumberCollection = new List<AddressBookNumber>();  
client.AddressBook("LOGIN", "PASSWORD",  
1, "test", null, out addressBookCollection, out AddressBookNumberCollection);
```

Usunięcie grupy numerów

Do usunięcia grupy numerów należy użyć parametru **operationId** z wartością 2 i podać nazwę danej grupy do usunięcia w parametrze **groupName**.

Przykład użycia w języku C#:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<AddressBook> addressBookCollection = new List<AddressBook>();  
List<AddressBookNumber> AddressBookNumberCollection = new List<AddressBookNumber>();  
client.AddressBook("LOGIN", "PASSWORD",  
2, "test", null, out addressBookCollection, out AddressBookNumberCollection);
```



Dodanie numeru do grupy numerów

Do dodania numeru do grupy numerów należy użyć parametru **operationId** z wartością 3, podać nazwę grupy, do jakiej chcemy dodać numer w parametrze **groupName** i podać numer, jaki chcemy dodać w parametrze **number**.

Przykład użycia w języku C#:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<AddressBook> addressBookCollection = new List<AddressBook>();  
List<AddressBookNumber> AddressBookNumberCollection = new List<AddressBookNumber>();  
client.AddressBook("LOGIN", "PASSWORD",  
3, "test", "test", out addressBookCollection, out AddressBookNumberCollection);
```

Usunięcie numeru z grupy numerów

Do usunięcia numeru z grupy numerów należy użyć parametru **operationId** z wartością 4, podać nazwę grupy, jaką chcemy usunąć w parametrze **groupName** i podać numer, jaki chcemy usunąć w parametrze **number**.

Przykład użycia w języku C#:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<AddressBook> addressBookCollection = new List<AddressBook>();  
List<AddressBookNumber> AddressBookNumberCollection = new List<AddressBookNumber>();  
client.AddressBook("LOGIN", "PASSWORD",  
4, "test", "test", out addressBookCollection, out AddressBookNumberCollection);
```

Pobranie listy grup numerów

Do pobrania listy grup numerów należy użyć parametru **operationId** z wartością 5.

Przykład użycia:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<AddressBook> addressBookCollection = new List<AddressBook>();  
List<AddressBookNumber> AddressBookNumberCollection = new List<AddressBookNumber>();  
client.AddressBook("LOGIN", "PASSWORD",  
5, null, null, out addressBookCollection, out AddressBookNumberCollection);
```



Pobranie listy numerów przypisanych do danej grupy

Do pobrania listy przypisanych do danej grupy numerów należy użyć parametru **operationId** z wartością 6, a także podać nazwę danej grupy w parametrze **groupName**.

Przykład użycia:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<AddressBook> addressBookCollection = new List<AddressBook>();  
List<AddressBookNumber> AddressBookNumberCollection = new List<AddressBookNumber>();  
client.AddressBook("LOGIN", "PASSWORD",  
6, "test", null, out addressBookCollection, out AddressBookNumberCollection);
```



RAPORTY SMS

Do pobrania raportów SMS służy metoda *GetSendSmsCollection*

Parametry obowiązkowe

userName - nazwa klienta

userPassword – hasło zakodowane MD5

Parametry do filtrowania

startDate - data od jakiej mają zostać pobrane raporty w formacie RRRR-MM-DD GG:MM

endDate - data do jakiej mają zostać pobrane raporty w formacie RRRR-MM-DD GG:MM

pageIndex - numer strony jaka ma być wyświetlona

pageCount - rozmiar strony do wyświetlenia

Parametry dodatkowe

messageId - parametr potrzebny do pobrania pojedynczego raportu, id uzyskane podczas wysłania wiadomości

Przykład użycia w języku C#:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<SendSms> collection = new List<SendSms>();  
SendSmsFilter filter = new SendSmsFilter();  
filter.IsSend = true;  
client.GetSendSmsCollection("LOGIN", "PASSWORD", filter, null, null, null, out collection);
```



USUWANIE ZAPLANOWANYCH SMSÓW

Do pobrania usuwanie zaplanowanych smsów służy metoda RemovePlannedSms

Parametry obowiązkowe

userName - nazwa klienta

userPassword – hasło zakodowane MD5

smsGroupId – id grupy odbiorców(GUID), które otrzymujemy podczas wysyłania smów.

Parametr opcjonalny

receiver - jest to numer odbiorcy(MSISDN). Jeżeli go wstawimy zostanie usunięty jedynie podany odbiorca wiadomości z zaplanowanego sms'u.

Przykład użycia w języku C#:

```
AhmesSmsServiceClient client = new AhmesSmsServiceClient();  
List<SendSms> collection = new List<SendSms>();  
SendSmsFilter filter = new SendSmsFilter();  
filter.IsSend = false;  
client.RemovePlannedSms("LOGIN", "PASSWORD",  
Guid.Parse("A97D12B8-B6C9-4E7E-B4E7-7A7A29281780"), null);
```