



Dokumentacja API v 2.0

Kraków, 22 listopada 2011

FreshMail, ul. Fabryczna 20a, 31-553 Kraków
tel. +48 12 617 61 40, info@freshmail.pl, freshmail.pl



Wersja dokumentu: 2.0.5

Autor dokumentu: Michał Białas <michal.bialas@freshmail.pl>

Zmiany

- 2.0.5 – dodano nowy parametr opcjonalny do metody createAccount pozwalający tworzyć konta podrzędne z poziomu API. Zwiększono liczbę przykładów. Poprawiono stylistykę w przykładach.
- 2.0.4 – dodano nową metodę putSubscribers oraz przykład jej wykorzystania. Zmiana url do wsdl'a oraz soap'a.
- 2.0.3 – pierwsza opublikowana wersja.

Opis API

Api systemu FreshMail zostało zrealizowane za pomocą technologii SOAP 1.1. Umożliwia logowanie, zarządzanie: kontami, listami, subskrybentami, polami dodatkowymi oraz wysyłanie kampanii.

Adres pliku wsdl: <https://api.freshmail.pl/soap?wsdl>

Obsługa błędów

Każda metoda dostępna w API zwraca wyjątek w przypadku wystąpienia błędu.

Opis metod

Metody Podstawowe

1. loginAccount

Logowanie do konta. Metoda zwraca id sesji użytkownika.

```
string loginAccount(string login, string password)
```

login	login, email użytkownika
password	hasło użytkownika

2. logoutAccount

Wylogowanie użytkownika.

```
string logoutAccount()
```

Przykład:

```
<?php  
$login = "freshmail@example.com";
```

```

$pass = "examplepassword";
$wsdlUrl = "https://api.freshmail.pl/soap?wsdl";
try {
    $soap = new SoapClient($wsdlUrl);
    $soap->loginAccount(array("login" => $login, "password" => $pass));
    //jestem zalogowany
    $soap->logoutAccount();
    //zostałem wylogowany
} catch (Exception $e) {
    echo "Wystąpił błąd: " . $e->getMessage() . PHP_EOL;
}

```

Metody do zarządzania kontami

1. createAccount

Tworzenie nowego konta w systemie. Metoda zwraca identyfikator użytkownika.

```

ClientAccountHash createAccount(string login, string password, string name,
                                string surname, string company, string phone,
                                bool needActivation, bool sendActivationMail,
                                bool childAccount)

```

login	login konta (adres email)
password	hasło
name	imię głównego użytkownika
surname	nazwisko głównego użytkownika
company	nazwa firmy
phone	numer telefonu
needActivation	czy wymagana jest aktywacja konta poprzez mail
sendActivationMail	czy wysyłać maila aktywacyjnego
childAccount	czy konto ma być podrzędne w stosunku do konta wywołującego metodę

Przykład:

```

<?php
$login = "freshmail@example.com";
$pass = "examplepassword";
$wsdlUrl = "https://api.freshmail.pl/soap?wsdl";
try {
    $soap = new SoapClient($wsdlUrl);
    $soap->loginAccount(array("login" => $login, "password" => $pass));

    $account = array(

```

```

        'login' => 'freshmail2@example.com',
        'password' => 'examplepassword2',
        'name' => 'First name',
        'surname' => 'Last name',
        'company' => 'Example Company',
        'phone' => '999888777',
        'childAccount' => true
    );
    $soap->createAccount($account); //utworzenie konta podrzédnego

    $soap->logoutAccount();
} catch (Exception $e) {
    echo "Wystąpił bład: " . $e->getMessage() . PHP_EOL;
}

```

2. updateAccount

Aktualizacja bieżącego konta. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```

integer updateAccount(string password, string name, string surname,
                    string company, string phone)

```

password	hasło
name	imię głównego użytkownika
surname	nazwisko głównego użytkownika
company	nazwa firmy
phone	numer telefonu

3. activateAccount

Aktywacja konta. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```

integer activateAccount(string clientAccountHash)

```

clientAccountHash	identyfikator konta
--------------------------	---------------------

4. getClientHash

Pobieranie bieżącego identyfikatora konta. Zwraca identyfikator konta

```

clientAccountHash getClientHash()

```

5. createAccountUser

Tworzenie użytkownika do konta. Metoda zwraca identyfikator użytkownika.

```
string createAccountUser(string login, string password, string name,  
                        string surname, string company, string phone,  
                        bool needActivation, bool sendActivationMail)
```

login	login konta (adres email)
password	hasło
name	imię głównego użytkownika
surname	nazwisko głównego użytkownika
company	nazwa firmy
phone	numer telefonu
needActivation	czy wymagana jest aktywacja konta poprzez mail
sendActivationMail	czy wysyłać maila aktywacyjnego

6. updateAccountUser

Aktualizacja użytkownika do konta. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer updateAccountUser(string password, string name, string surname,  
                          string company, string phone)
```

password	hasło
name	imię głównego użytkownika
surname	nazwisko głównego użytkownika
company	nazwa firmy
phone	numer telefonu

7. activateAccountUser

Aktywacja użytkownika do konta. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer activateAccountUser(string clientAccountUserHash)
```

clientAccountUserHash	identyfikator użytkownika
------------------------------	---------------------------

Metody do zarządzania listami subskrypcyjnymi

1. createSubscriberList

Tworzenie nowej listy subskrypcyjnej. Zwraca identyfikator listy subskrypcyjnej.

```
string createSubscriberList(string name, string description)
```

name	nazwa listy
description	opis listy

2. updateSubscriberList

Aktualizacja danych listy subskrypcyjnej. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer updateSubscriberList(string subscriberListHash, string name,  
                             string description)
```

subscriberListHash	Identyfikator listy subskrypcyjnej
name	nazwa listy
description	opis listy

3. getSubscriberLists

Pobranie wszystkich list subskrypcyjnych aktualnego użytkownika zalogowanego na konto. Zwraca tablicę obiektów **subscriberList**.

```
subscriberLists getSubscriberLists()
```

obiekt **subscriberlist**

string subscriberListHash	identyfikator listy subskrypcyjnej
string name	nazwa listy
string description	opis listy
string creation_date	data utworzenia listy

Przykład:

```
<?php  
$login = "freshmail@example.com";  
$pass = "examplepassword";  
$wsdlUrl = "https://api.freshmail.pl/soap?wsdl";  
try {  
    $soap = new SoapClient($wsdlUrl);  
    $soap->loginAccount(array("login" => $login, "password" => $pass));  
    //metoda zwraca wszystkie listy subskrybenta  
    $subscribersLists = $soap->getSubscriberLists();  
    var_dump($subscribersLists);  
    $soap->logoutAccount();  
} catch (Exception $e) {  
    echo "Wystąpił błąd: " . $e->getMessage() . PHP_EOL;
```

```
}
```

4. createCustomField

Tworzenie nowego pola dodatkowego na liście subskrypcyjnej. Zwraca identyfikator pola dodatkowego.

```
string createCustomField(string subscriberListHash, string name,  
                        string personalizationTag, string type)
```

subscriberListHash	identyfikator pola dodatkowego
name	nazwa pola dodatkowego
personalizationTag	tag personalizacji
type	text albo number, oznacza typ pola dodatkowego

5. updateCustomField

Aktualizacja pola dodatkowego. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer updateCustomField(string subscriberListHash, string customFieldHash,  
                        string name, string personalizationTag)
```

subscriberListHash	identyfikator listy subskrypcyjnej
customFieldHash	identyfikator pola dodatkowego
name	nazwa pola dodatkowego
personalizationTag	tag personalizacji

6. getCustomFieldsFromSubscriberList

Pobranie wszystkich pól dodatkowych z listy subskrypcyjnej. Zwraca tablicę obiektów **customField**.

```
customFields getCustomFieldsFromSubscriberList(string subscriberListHash)
```

subscriberListHash	identyfikator listy subskrypcyjnej
---------------------------	------------------------------------

obiekt **customField**

string customFieldHash	identyfikator pola dodatkowego
string name	nazwa pola dodatkowego
string personalizationTag	tag personalizacji
string type	text albo number, oznacza typ pola dodatkowego

Metody do zarządzania subskrybentami oraz polami dodatkowymi

1. createSubscriber

Tworzenie nowego subskrybenta. Metoda zwraca identyfikator subskrybenta.

```
string createSubscriber(string subscriberListHash, string email, string name,  
                        bool needActivation, bool sendActivationMail)
```

subscriberListHash	identyfikator listy subskrypcyjnej
email	email subskrybenta
name	nazwa subskrybenta
needActivation	czy wymagana jest aktywacja konta poprzez mail
sendActivationMail	czy wysyłać maila aktywacyjnego

2. updateSubscriber

Aktualizacja danych subskrybenta. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer updateSubscriber(ChooseSubscriber chooseSubscriber, Set set)
```

chooseSubscriber	obiekt ChooseSubscriber (wybór subskrybenta do modyfikacji)
set	obiekt Set (modyfikacja subskrybenta)

obiekt **ChooseSubscriber**

string byColumn	możliwe 2 wartości <code>subscriberHash</code> lub <code>subscriberEmail</code> .
string columnValue	w zależności od wyboru kolumny albo identyfikator subskrybenta albo jego adres email
string subscriberListHash	identyfikator listy subskrypcyjnej (argument opcjonalny)

obiekt **Set**

string email	adres email subskrybenta
string name	nazwa subskrybenta
string state	status subskrybenta. Należy użyć jednego z poniższych: dontChange , active , toActivation , bounceSoft , bounceHard , unsubscribed , deleted ; co znaczy odpowiednio: „nie zmieniaj”, „aktywny”, „do aktywacji”, „odbitcie miękkie”, „odbitcie twarde”, „wypisany”, „usunięty”
customFieldsValues	Tablica obiektów customFieldsValue

Przykład:

```
<?php  
$login = "freshmail@example.com";
```



```

$pass = "examplepassword";
$wsdlUrl = "https://api.freshmail.pl/soap?wsdl";
try {
    $soap = new SoapClient($wsdlUrl);
    $soap->loginAccount(array("login" => $login, "password" => $pass));
    $updateSubscribers = array(
        'chooseSubscriber' => array(
            'byColumn' => 'subscriberHash',
            'columnValue' => 'xxxxxxxxxx',
            'subscriberListHash' => 'xxxxxyyyyyy'
        ),
        'set' => array(
            'customFieldsValues' => array(
                array(
                    'customFieldHash' => 'yyyyyxxxxx',
                    'value' => 'Example1'
                )
            ),
        )
    );

    $soap->updateSubscriber($updateSubscribers);
    $soap->logoutAccount();
} catch (Exception $e) {
    echo "Wystąpił błąd: " . $e->getMessage() . PHP_EOL;
}

```

3. activateSubscriber

Aktywacja subskrybenta. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer activateSubscriber(string subscriberListHash, string email)
```

subscriberListHash	identyfikator listy subskrypcyjnej
email	adres email subskrybenta

4. getSubscriberByEmail

Pobranie danych subskrybenta przy pomocy adresu email. Zwraca tablicę obiektów **subscriber**.

```
subscribers getSubscriberByEmail(string email, string subscriberListHash)
```

email	adres email szukanego subskrybenta
subscriberListHash	identyfikator listy subskrypcyjnej (argument opcjonalny)

obiekt **subscriber**

string subscriberListHash	identyfikator listy subskrypcyjnej
string subscriberHash	identyfikator subskrybenta
string email	adres email subskrybenta
string name	nazwa subskrybenta
string added_on	data dodania subskrybenta
string state	status subskrybenta
string last_update	ostatnia aktualizacja danych subskrybenta
string last_send	data ostatniej wysyłki do subskrybenta
customField[] customFields	tablica obiektów customField

5. **getSubscriberByHash**

Pobranie danych subskrybenta przy pomocy identyfikatora subskrybenta. Zwraca tablicę obiektów **subscriber**.

```
subscribers getSubscriberByHash(string subscriberHash)
```

subscriberHash	identyfikator subskrybenta
-----------------------	----------------------------

6. **updateCustomFieldsValues**

Aktualizacja pól dodatkowych. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer updateCustomFieldsValues(UpdateCustomFieldValue[] updateCustomFieldValues)
```

updateCustomFieldValues	tablica obiektów UpdateCustomFieldValue
--------------------------------	---

obiekt **UpdateCustomFieldValue**

string customFieldHash	identyfikator pola dodatkowego
string subscriberHash	identyfikator subskrybenta
string value	wartość pola dodatkowego

Przykład:

```
<?php
$login = "freshmail@example.com";
$pass = "examplepassword";
$wsdlUrl = "https://api.freshmail.pl/soap?wsdl";
try {
    $soap = new SoapClient($wsdlUrl);
```

```

$soap->loginAccount(array("login" => $login, "password" => $pass));
$customFields = array(
    array(
        'customFieldHash' => 'xxxxxyyyyy',
        'subscriberHash' => 'xxxxxxxxxx',
        'value' => 'Example1'
    ),
    array(
        'customFieldHash' => 'xxxxxyyyyz',
        'subscriberHash' => 'xxxxxxxxxx',
        'value' => 'Example2'
    )
);

$soap->updateCustomFieldsValues($customFields);
$soap->logoutAccount();
} catch (Exception $e) {
    echo "Wystąpił błąd: " . $e->getMessage() . PHP_EOL;
}

```

7. getCustomFieldsValuesForSubscriber

Pobranie pól dodatkowych dla subskrybenta. Zwraca tablicę obiektów **customFieldsValue**.

`customFieldsValues` `getCustomFieldsValuesForSubscriber`(string **subscriberHash**)

subscriberHash	identyfikator subskrybenta
-----------------------	----------------------------

obiekt **customFieldsValue**

string customFieldHash	identyfikator pola dodatkowego
string personalizationTag	tag personalizacji
string name	nazwa pola dodatkowego
string value	wartość pola dodatkowego

Metody do wysyłania kampanii

1. createCampaign

Tworzenie nowej kampanii o określonej nazwie na podstawie przesłanego tekstu i html lub przy pomocy pobierania kampanii z url'a. Metoda zwraca hash kampanii potrzebny do dalszego manipulowania ustawieniami oraz do wysyłki.

string `createCampaign`(string **name**, string **html**, string **text**, string **url**)

name	nazwa kampanii
-------------	----------------

html	treść html kampanii, jeżeli parametr jest pusty to jest tworzona tylko wersja tekstowa.
text	treść tekstowa kampanii, jeżeli parametr jest pusty to jest tworzona tylko wersja htmlowa.
url	odnośnik z którego zostanie pobrana kampania. Jeżeli nie jest pusty albo nie podany to zastępuje wersję html'ową kampanii. (Parametr opcjonalny).

2. setCampaignParameters

Ustawienie podstawowych parametrów kampanii takich jak: temat, nadawca czy pole „odpowiedz do”. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer setCampaignParameters(string campaign, CampaignParameter[] campaignParameter)
```

campaign	identyfikator kampanii
campaignParameter	tablica obiektów CampaignParameter

obiekt **CampaignParameter**

string key	klucz parametrów
string value	wartości parametrów

Lista parametrów

subject	temat mailingu
fromEmail	nadawca maila, adres email
fromName	nadawca maila, nazwa
replyTo	adres email „odpowiedz do”

3. setCampaignSubscribersLists

Ustawienie list subskrypcyjnych do których jest wysyłana kampania. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer setCampaignSubscribersLists(string campaign, SubscribersList[] subscribersLists)
```

campaign	identyfikator kampanii
subscribersLists	tablica obiektów SubscribersList

obiekt **SubscribersList**

string subscribersListHash	identyfikator listy subskrypcyjnej, opcjonalnie można przekazać identyfikator listy subskrypcyjnej i po przecinku identyfikator grupy
-----------------------------------	---

4. sendTestCampaign

Wysyłanie testowej kampanii. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer sendTestCampaign(string campaign, string emails, string customFieldsFill)
```

campaign	identyfikator kampanii
emails	adresy email na które zostanie wysłana kampania. Jeżeli jest więcej niż jeden to powinny być oddzielone przecinkami
customFieldsFill	sposób wypełniania pól dodatkowych w mailingu testowym. Na razie jedynym obsługiwanym jest random (wybiera losowego subskrybenta z listy subskrypcyjnej wybranych do wysłania i używa jego pól dodatkowych do personalizacji mailingu).

5. sendCampaign

Wysłanie kampanii. Metoda zwraca 1 jeżeli wszystko się powiodło, 0 w przeciwnym wypadku.

```
integer sendCampaign(string campaign, datetime sendTime)
```

campaign	identyfikator kampanii
sendTime	data wysłanie kampanii (argument opcjonalny), jeżeli data nie jest określona to kampania jest ustawiona do wysłania natychmiast

6. getCampaignState

Pobranie statusu kampanii. Zwraca liczbowy status kampanii.

```
int getCampaignState(string campaign)
```

campaign	identyfikator kampanii
-----------------	------------------------

Statusy kampanii

1	Kampania ze zdefiniowaną treścią maila
2	Kampania z ustawionymi parametrami
3	Kampania z ustawionymi listami subskrypcyjnymi
4	Kampania ustawiona do wysyłki
5	Kampania wysłana
6	Kampania w trakcie wysyłki

7. putSubscribers

Utworzenie tymczasowej listy subskrypcyjnej na potrzeby wysłania kampanii. Metoda może być wywołana wiele razy dla jednej kampanii. Każde wywołanie zwraca ten sam zestaw identyfikatorów. Metoda zwraca identyfikator listy i identyfikator grupy tymczasowej rozdzielony przecinkiem, służy jako parametr wejściowy dla metody **setCampaignSubscribersLists** jako lista subskrypcyjna.

```
string putSubscribers(string campaign, subscribersArray subscribers)
```

campaign	identyfikator kampanii
subscribers	obiekt zawierający tablicę subskrybentów wraz z polami dodatkowymi

Obiekt subscribersArray

subscribers	tablica obiektów quickSubscriber
--------------------	---

Obiekt **quickSubscriber**

email	adres email
name	nazwa
customFields	tablica obiektów quickCustomFieldsValues

Obiekt **quickCustomFieldsValues**

personalizationTag	nazwa tagu personalizacji, nazwa pola dodatkowego
value	wartość pola dodatkowego

Przykład:

```
<?php
$login = "freshmail@example.com";
$pass = "examplepassword";
$wsdlUrl = "https://api.freshmail.pl/soap?wsdl";
try {
    $soap = new SoapClient($wsdlUrl);
    $soap->loginAccount(array("login" => $login, "password" => $pass));
    $result = $soap->createCampaign(array(
        "name" => "Campaign name",
        "html" => '<p>Witaj $$imie$$</p>',
        "text" => "",
    ));
    $campaignHash = $result->return;
    $result = $soap->setCampaignParameters(array(
        "campaign" => $campaignHash,
        "campaignParameter" => array(
            array("key" => "subject", "value" => "Example subject"),
            array("key" => "fromEmail", "value" => "from@example.com"),
            array("key" => "fromName", "value" => "Example Name"),
            array("key" => "replyTo", "value" => "replyto@example.com"),
        )
    ));
    $result = $soap->putSubscribers(
        array(
```

```

        "campaign" => $campaignHash,
        "subscribers" => array(
            array(
                "email" => "test2@example.com",
                "name" => "Michał",
                "customFields" => array(
                    array(
                        'personalizationTag' => 'imie',
                        'value' => 'Michał'
                    )
                )
            ),
        )
    )
);
$subscribersList = array(array("subscribersListHash" => $result->return));
$result = $soap->setCampaignSubscribersLists(array(
    "campaign" => $campaignHash,
    "subscribersLists" => $subscribersList
));
$result = $soap->sendCampaign(array("campaign" => $campaignHash);
$soap->logoutAccount();
} catch (Exception $e) {
    echo "Wystąpił błąd: " . $e->getMessage() . PHP_EOL;
}

```