Dokumentacja REST API v 3.0

Kraków, 1 lipca 2019

FreshMail Sp. z o.o., Al. 29 Listopada 155 c, 31-406 Kraków
tel. +48 12 617 61 40, info@freshmail.pl, freshmail.pl

Spis treści

Opis API **4**
Uwierzytelnienie 4
Odpowiedzi API 6
Obsługa błędów 6
Ścieżka dostępu 7
Ogólne kody błędów 7
Opis metod używanych w API **8**
Ping – testowanie 8
Maile transakcyjne – wysyłanie pojedynczych maili 9
Wiadomości SMS 11
Uwagi ogólne 11
Rodzaje wiadomości SMS 12
Wysyłanie wiadomości SMS 13
Raporty – raporty z kampanii 14
Pobieranie listy wysłanych kampanii 14
Pobieranie zbiorczego raportu z kampanii 15
Pobieranie zachowań subskrybentów w czasie 16
Kampanie – obsługa kampanii 18
Tworzenie nowej kampanii 18
Wysłanie kampanii 20
Kasowanie kampanii 22
Wysłanie kampanii testowej 23
Wysłanie kampanii 24
Subskrybenci – zarządzanie subskrybentami 25
Kody statusów subskrybentów 25
Dodanie subskrybenta 26
Modyfikowanie danych subskrybenta 27
Pobieranie danych subskrybenta 28
Kasowanie subskrybenta 29
Subskrybenci – zarządzanie wieloma subskrybentami w pojedynczym żądaniu 30
Dodawanie wielu subskrybentów 30
Modyfikowanie danych wielu subskrybentów 32
Modyfikowanie pól dodatkowych wszystkich subskrybentów 33
Pobieranie danych wielu subskrybentów 35
Kasowanie wielu subskrybentów 36

Tworzenie kont....37
Listy subskrypcyjne - zarządzanie....39
Tworzenie list....39
Modyfikowanie list....40
Usuwanie list....41
Pobieranie danych o listach....42
Tworzenie pól dodatkowych w listach....43
Wyświetlenie listy pól dodatkowych w podanej liście....44
Spamtesty....45
Sprawdzenie zgodności z filtrami SpamAssassin....45

Wersja dokumentu: 1.0.24

Autorzy: Tadeusz Kania,
Piotr Suszalski,
Grzegorz Gorczyca

Kontakt: api@freshmail.pl

Opis API

API systemu FreshMail zostało zrealizowane według wzorca REST. Komunikacja polega na wysłaniu żądania HTTP pod adres https://api.freshmail.com/rest/ i ewentualnym przesłaniu wymaganych danych. Odpowiedź jest zwracana w formacie JSON.

Uwierzytelnienie

Każde wysłane żądanie po API wymaga uwierzytelnienia. Uwierzytelnienie następuje poprzez wysłanie nagłówków HTTP o nazwach:

X-Rest-ApiKey	zawiera unikalny klucz dostępu do API, każdy z klientów posiada swój unikalny klucz API. subskrypcje → lista subskrypcyjna → opcje zaawansowane i API
X-Rest-ApiSign	zawiera podpis żądania. Podpis jest generowany na podstawie parametrów wysyłanych w żądaniu oraz na podstawie ApiSecret.

Istnieją dwa schematy generowania **X-Rest-ApiSign**, oba zostały przedstawione poniżej:
Schemat oparty na **URL-encoded query**:
wymagane jest dodanie nagłówka Content-Type w żądaniu http o wartości:

text/html lub application/xml lub application/json, np.:

Content-Type: application/json

wtedy można wygenerować **X-Rest-ApiSign** w oparciu o algorytm:

sha1(ApiKey + GET_data + POST_data + ApiSecret)

gdzie:

ApiKey	32 znakowy Api key.
GET_data	pełna ścieżka dostępu – np. /rest/ping, /rest/mail itp. Więcej szczegółów w sekcji „*Ścieżka dostępu*".
POST_data	dane POST umieszczone w żądaniu. Jeśli w żądaniu znajdują się tylko dane GET to wartość POST_data jest pusta. Wartości muszą być escapowane na potrzeby URL, np.: email=test%40test.pl&subject=test+emaila
ApiSecret	40 znaków, unikalny sekret dla każdego z klientów. **W przypadku skompromitowania klucza należy bezzwłocznie wygenerować nowy.**

Schemat oparty na **JSON**:

wymagane jest dodanie nagłówka Content-Type w żądaniu http w postaci:

Content-Type: application/json

wtedy można wygenerować **X-Rest-ApiSign** w oparciu o algorytm:

sha1(ApiKey + GET_data + JSON_data + ApiSecret)

gdzie:

ApiKey	32 znakowy Api key.
GET_data	pełna ścieżka dostępu – np. /rest/ping, /rest/mail itp. Więcej szczegółów w sekcji „*Ścieżka dostępu*”.
JSON_data	dane w formacie JSON jakie są wysyłane w żądaniu. Jeśli w żądaniu znajdują się tylko dane GET to wartość JSON_data jest pusta. Przykładowa wartość JSON_data: {"subscriber":"test@test.pl"}
ApiSecret	40 znaków, unikalny sekret dla każdego z klientów. **W przypadku skompromitowania klucza należy bezzwłocznie wygenerować nowy.**

Odpowiedzi API

Odpowiedź na każde żądanie wysłane do API jest w formacie JSON, np.

```
{"status":"OK","data":[1,2,3]}
```

Jeśli zapytanie powiedzie się wtedy pod kluczem „**status**” będziemy mieli wartość '**OK**'. Jeśli metoda ma zwrócić dane będą one pod kluczem „**data**”. Natomiast jeśli wystąpi błąd pod kluczem „**status**” będzie wartość „**ERROR**”, a pod kluczem „**errors**” znajdzie się opis błędów jakie wystąpiły wraz z kodami błędów. W przypadku błędu kod HTTP zwracany w nagłówku jest adekwatny do zaistniałego błędu, czyli np. **404**, **403** itp. W przypadku braku błędu kod HTTP to **200**.

Obsługa błędów

W razie wystąpienia błędu status odpowiedzi jest równy „**ERROR**” natomiast pole „**errors**” zawiera opis słowny błędu oraz kod błędu, np.

```
{"errors":[
        {"message":"Brak tematu emaila",
         "code": 1202}
        ],
 "status":"ERROR"}
```

lub

```
{"errors":[
        {"message": "Błędny adres odbiorcy",
         "code": 1201},
        {"message": "Brak tematu emaila",
         "code": 1202}
        ],
 "status":"ERROR"}
```

Ścieżka dostępu

Każda z akcji jaką można wykonać ma swoją osobną ścieżkę dostępu. Ścieżka zbudowana jest w następujący sposób:

/rest/[Controller]/[Action]/[param]/[param]/[param]

Controller	nazwa kontrolera np. ping, mail, itp.
Action	nazwa akcji w obrębie kontrolera.
param	parametr, nie może zawierać znaku „/”, w większości przypadków dodatkowe parametry GET są zbędne. Jeśli są wymagane to ilość oraz ich format jest podany w opisie akcji.

Ogólne kody błędów

500	Błąd serwera (Internal Server Error).
1000	Brak autoryzacji.
1001	Nie znaleziono kontrolera / akcji.
1002	Nieobsługiwany protokół. Proszę skorzystać z protokołu https.
1003	Nieobsługiwany rodzaj żądania, np. wysłano GET, a oczekiwano POST.
1004	Brak uprawnień do wywołania danego kontrolera/akcji

Opis metod używanych w API

Ping – testowanie

Kontroler ten służy do przetestowania poprawności działania API. Osobno można przetestować poprawność działania żądań typu GET oraz żądań typu POST.

Wywołanie: **GET /rest/ping**

W odpowiedzi serwer zwróci „pong”.

Wywołanie: **POST /rest/ping**

W odpowiedzi serwer zwróci dane które wysyła się w POST.

Maile transakcyjne – wysyłanie pojedynczych maili

Kontroler służy do wysyłki pojedynczych maili transakcyjnych. Domyślnie wszystkie parametry powinny być przekazane w kodowaniu **UTF-8**. Można to zmienić korzystając z parametru **encoding**.

Wywołanie: **POST /rest/mail**

Dane w POST:

subscriber	Wymagane	Adres email subskrybenta do którego wysyłamy email.
subject	Wymagane	Tytuł emaila.
html	Wymagane*	Treść html emaila (wymagana jeśli nie ma treści text).
text	Wymagane*	Treść tekstowa emaila (wymagana jeśli nie ma treści html).
from	Opcjonalne	Pole „Od”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
from_name	Opcjonalne	Pole „Nazwa nadawcy”, jeśli będzie puste w to miejsce wstawi się domyślna nazwa nadawcy z systemu.
reply_to	Opcjonalne	Pole „Odpowiedz do”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
encoding	Opcjonalne	Kodowanie w jakim ma zostać wysłany email. Domyślnie jest to UTF-8. Pamiętaj, że email musi być przekazany w tym samym kodowaniu, które jest przekazane w parametrze encoding. Obsługiwane kodowania to: UTF-8, iso-8859-2
attachments	Opcjonalne	Link do załącznika, który ma zostać dołączony do maila. Maksymalny rozmiar załączników to 0.5 MB
tracking	Opcjonalne	Pole przyjmujące wartośc 0/1. Jeśli przekazane zostanie 1 to email będzie trackingowany przez kody śledzące zachowania subskrybenta. **Uwaga**: **oznacza to, że domena w jakiej pojawiają się linki zostanie zmieniona oraz do wersji HTML zostanie dodany obrazek śledzący.**
domain	Opcjonalnie	Domena do rebrandingu, wartość aktywna tylko jeśli jest włączony dla maila tracking. Jeśli wartość będzie pusta lub nie zostanie przekazana zostanie umieszczona domyślna domena do rebrandingu. Przykładowa domena: mojanazwafirmy.mailnews.pl **Uwaga: aby korzystać z domen do rebrandingu w pierwszej kolejności należy taką domenę zgłosić korzystając z panelu w aplikacji.**
tag	Opcjonalnie	Oznaczenie typu wysyłanego maila. Wykorzystując tagi, raporty z maili transakcyjnych zostaną podzielone na podgrupy. Przykładowe tagi: Mail Akceptacyjny, Mail przypominający, itp.

Kody błędów:

1201	Błędny adres subskrybenta.
1202	Brak tematu maila.
1203	Brak treści HTML oraz tekstowej.
1204	Błędny adres „odpowiedz do”.
1205	Błędny adres email nadawcy.
1206	Nazwa nadawcy nie może być pusta.
1207	Przekroczono dzienny limit wysłanych maili.
1208	Nieobsługiwane kodowanie.
1209	Nieprawidłowy link do załącznika.
1210	Plik załącznika nie istnieje.
1211	Przekroczono maksymalny rozmiar załączników.
1212	Niepoprawna domena do rebrandingu, sprawdź czy domena została aktywowana.
1213	Tag kampanii niepoprawny (za długi lub zawierający niedozwolone znaki)
1250	Nie udało się wysłać emaila, błąd systemu.
1251	Nie wysłano emaila do tego subskrybenta, zablokowany administracyjnie

Wiadomości SMS

Kontroler służy do wysyłki pojedynczych wiadomości SMS. Domyślnie wszystkie parametry muszą być przekazane w kodowaniu **UTF-8**.

Uwagi ogólne

Numery telefonu muszą być podawane w **formacie 9 lub 11 cyfrowym** (np. 48502602702 lub 502602702), możliwe jest także dodanie znaku „+” przed prefixem kraju (np. +48502602702).

Treść pojedynczej wiadomości to standardowo do 160 znaków lub 70 znaków w przypadku wystąpienia chociaż jednego znaku specjalnego. Maksymalna długość wiadomości wynosi 459 znaków (lub 200 ze znakami specjalnymi) i jest wysłana jako 3 połączone SMS-y.

Znakami specjalnymi są wszystkie znaki nie mieszczące się w zestawie znaków:

@£$¥èéùìòÇØøÅå_^{}\[~]| ÆæßÉ!"#¤%&'()*+,-./0-9:;<=>?A-ZÄÖÑÜ§¿a-zäöñüà <enter>

Uwaga!

Znaki: ^ { } [ ] ~ \ | <enter> zawsze liczone są podwójnie (bez względu na to czy wysyłamy SMS ze znakami specjalnymi czy bez) ze względu na wymogi specyfikacji GSM.

SMS bez znaków specjalnych		**SMS ze znakami specjalnymi**	
Maksymalna ilość znaków	Ilość SMSów	Maksymalna ilość znaków	Ilość SMSów
160	1	70	1
306	2	134	2
459	3	200	3

Rodzaje wiadomości SMS

Bramka umożliwia wysłanie wiadomości dwóch rodzajów: wiadomości typu **ECO** oraz wiadomości typu **PRO.**

SMS typu ECO to wiadomość wysyłana z losowego 9cio cyfrowego numeru telefonu.

SMS typu PRO to wiadomość wysyłana z nadpisem alfanumerycznym, czyli zdefiniowanym polem nadawcy. Pole to nie może zawierać więcej niż 11 znaków z zestawu a-zA-Z0-9 oraz znak myślnika „-”. Aby korzystać z SMS typu PRO konieczne jest **zgłoszenie nadpisu**.

Aby móc korzystać z bramki SMS należy w pierwszej kolejności podpisać umowę. Ceny za sms typu **ECO** oraz typu **PRO** są zawarte w umowie.

Wysyłanie wiadomości SMS

Wywołanie: **POST /rest/sms**

Dane w POST:

gsm	Wymagane	Numer telefonu
text	Wymagane	Treść SMSa. Więcej szczegółów w sekcji „**Uwagi ogólne**”
from	Opcjonalne	Nadpis / nadawca. Więcej szczegółów w sekcji „**Rodzaje wiadomości SMS**”. Jeśli nadpis będzie poprawny zostanie wysłany sms typu **PRO**

Kody błędów:

1101	Numer GSM jest błędny
1102	Treść SMSa jest pusta
1103	Treść SMSa jest za długa (SMS bez znaków specjalnych)
1104	Treść SMSa jest za długa (SMS ze znakami specjalnymi)
1105	Nadawca jest błędny
1106	Nadawca został niepotwierdzony (nadawca jest poprawny, ale oczekuje na weryfikację)
1107	Żądanie jest w złym kodowaniu, obsługiwane kodowanie to UTF-8
1150	SMS nie zostanie wysłany, konieczne jest podpisanie umowy

Raporty – raporty z kampanii

Kontroler służy do pobierania raportów oraz list wysłanych kampanii.

Pobieranie listy wysłanych kampanii

Wywołanie: **GET /rest/reports/campaignsList/[page]**

Dane w GET:

page	Opcjonalne	Numer strony z kampaniami, domyślnie pobierana jest pierwsza strona

W odpowiedzi serwer zwróci listę 25 kampanii. Zwracane dane:

name	Nazwa kampanii
topic	Temat emaila w kampanii
subscribers	Ilość subskrybentów do których była wysyłka
id_hash	Unikalny identyfikator kampanii
sent	Data wysyłki kampanii, format: rrrr-mm-dd hh:ii:ss

Pobieranie zbiorczego raportu z kampanii

Wywołanie: **GET /rest/reports/campaign/[id_hash]**

Dane w GET:

id_hash	Wymagane	Unikalny identyfikator kampanii, można go otrzymać wywołując metodę /rest/reports/campaignsList

Zwracane dane:

subscribers	Liczba wszystkich subskrybentów do których wysłano kampanię
delivered	Liczba dostarczonych wiadomości
hard_bounce	Liczba wiadomości odbitych „twardo” (http://www.ietf.org/rfc/rfc1893.txt)
soft_bounce	Liczba wiadomości odbitych „miękko” (http://www.ietf.org/rfc/rfc1893.txt)
opened	Liczba wszystkich otwartych wiadomości
clicked	Liczba wszystkich klikniętych linków
unique_opened	Liczba unikalnych otwarć wiadomości
unique_clicked	Liczba unikalnych klikniętych wiadomości
resigned	Liczba wypisanych subskrybentów

Pobieranie zachowań subskrybentów w czasie

Wywołanie: **GET /rest/reports/campaignTimeDetails/[id_hash]**

Dane w GET:

id_hash	Wymagane	Unikalny identyfikator kampanii, można go otrzymać wywołując metodę /rest/reports/campaignsList

Zwracane dane – tablica z kolejnymi działaniami subskrybentów w czasie:


```
{"status":"OK","data":
    [
        {"opened":"4",
         "unique_opened":"4",
         "clicked":"3",
         "unique_clicked":"1",
         "time":"2012-03-26 16:50"},
        {"opened":"1",
         "unique_opened":"1",
         "clicked":"1",
         "unique_clicked":"1",
         "time":"2012-03-27 08:30"}
    ]
}
```


opened	Liczba wszystkich otwarć w danym przedziale czasowym
clicked	Liczba wszystkich kliknięć w danym przedziale czasowym
unique_opened	Liczba unikalnych otwartych emaili w danym przedziale czasowym
sunique_clicked	Liczba unikalnych emaili z klikniętym linkiem w danym przedziale czasowym
time	Czas w formacie „rrrr-mm-dd HH:ii”, rozdzielczość okna to 10 minut

Kody błędów:

1401	Niepoprawny numer strony
1402	Kampania o podanych id_hash nie istnieje

Kampanie – obsługa kampanii

Kontroler służy do tworzenia, sprawdzania i wysyłania kampanii.

Tworzenie nowej kampanii

Wywołanie: **POST /rest/campaigns/create**

Dane w POST:

name	Wymagane	Nazwa kampanii
url	Opcjonalne	Adres strony z której ma zostać pobrany kod HTML
html	Wymagane*	Treść html emaila (wymagana jeśli nie ma treści text oraz adresu url).
text	Wymagane*	Treść tekstowa emaila (wymagana jeśli nie ma treści html).
subject	Opcjonalne	Tytuł jaki ma mieć email wysłany przez system. W razie braku jako tytuł będzie wzięta nazwa kampanii.
from_address	Opcjonalne	Pole „Od”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
from_name	Opcjonalne	Pole „Nazwa nadawcy”, jeśli będzie puste w to miejsce wstawi się domyślna nazwa nadawcy z systemu.
reply_to	Opcjonalne	Pole „Odpowiedz do”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
list	Wymagane*	Hash listy subskrypcyjnej do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jedne listy. W przypadku wysyłki do grupy nie trzeba podawać listy.
group	Wymagane*	Hash grupy do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jedne grupy. W przypadku wysyłki do listy subskrypcyjnej nie trzeba podawać grupy.
resignlink	Opcjonalne	Adres URL na jaki zostanie przekierowany odbiorca po kliknięciu na link rezygnacji

Zwracane dane:

hash	W przypadku poprawnego utworzenia kampanii zwrócony zostaje jej hash

Kody błędów:

1701	Brak nazwy kampanii
1702	Brak treści (nie podano treści text, html ani url)
1703	Błędny adres url
1704	Brak klienta (błąd wewnętrzny)
1705	Brak parametrów (błąd wewnętrzny)
1706	Błędny adres nadawcy (from_address)
1707	Błędny adres „reply to”
1708	Brak listy subskrypcyjnej i grupy
1709	Co najmniej jeden hash listy jest błędny.
1710	Co najmniej jeden hash grupy jest błędny.
1711	Nie znaleziono list o podanym hashu.
1712	Nie znaleziono grupy o podanym hashu.
1713	Link rezygnacji musi być poprawnym adresem URL

Wysłanie kampanii

Wywołanie: **POST /rest/campaigns/edit**

Dane w POST:

id_hash	Wymagane	Hash kampanii do edytowania
name	Opcjonalne	Nazwa kampanii
url	Opcjonalne	Adres strony z której ma zostać pobrany kod HTML
html	Opcjonalne	Treść html emaila
text	Opcjonalne	Treść tekstowa emaila
subject	Opcjonalne	Tytuł jaki ma mieć email wysłany przez system. W razie braku jako tytuł będzie wzięta nazwa kampanii.
from_address	Opcjonalne	Pole „Od”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
from_name	Opcjonalne	Pole „Nazwa nadawcy”, jeśli będzie puste w to miejsce wstawi się domyślna nazwa nadawcy z systemu.
reply_to	Opcjonalne	Pole „Odpowiedz do”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
list	Opcjonalne	Hash listy subskrypcyjnej do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jedne listy. W przypadku wysyłki do grupy nie trzeba podawać listy.
Group	Opcjonalne	Hash grupy do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jedne grupy. W przypadku wysyłki do listy subskrypcyjnej nie trzeba podawać grupy.
resignlink	Opcjonalne	Adres URL na jaki zostanie przekierowany odbiorca po kliknięciu na link rezygnacji

Uwaga:

W żądaniu można podać tylko te parametry które chcemy aby uległy zmianie, jeśli dodamy parametr **list** lub **group** (nie można podać ich obu naraz) to stare ustawienie grup/list subskrypcyjnych zostanie całkowicie nadpisane.

Kody błędów:

1750	Brak kampanii do edycji o podanym hashu
1751	Kampania już ostała ustawiona do wysyłki, modyfikacja niemożliwa

Reszta kodów błędów jest identyczna jak w przypadku tworzenia kampanii.

Kasowanie kampanii

Wywołanie: **POST /rest/campaigns/delete**

Dane w POST:

hash	Wymagane	Hash kampanii do skasowania

Kody błędów:

1724	Brak kampanii o takim hashu
1798	Kampania już została skasowana

Wysłanie kampanii testowej

Wywołanie: **POST /rest/campaigns/sendTest**

Dane w POST:

hash	Wymagane	Hash kampanii którą chcemy wysłać
emails	Wymagane	Adres lub tablica adresów na które test ma być wysłany.
custom_fields	Opcjonalne	Tablica z polami dodatkowymi, w przypadku pozostawienia pola pustego wartości pól dodatkowych zostaną uzupełnione losowymi danymi z listy subskrypcyjnej

Przykładowe żądanie:

{"hash":"hash_kampanii",

"emails":"test@test.pl",

"custom_fields":{

"imie":"Jan",

"nazwisko":"Kowalski"}}

lub

{"hash":"hash_kampanii",

"emails":["test1@test.pl","test2@test.pl"]}

Kody błędów:

1721	Brak hasha lub jest on niepoprawny.
1722	Brak adresu email.
1723	Co najmniej jeden adres email jest niepoprawny.
1724	Kampania o takim hashu nie istnieje.
1725	Kampania nie jest jeszcze gotowa do wysyłki testowej.
1726	Wysyłka nie powiodła się.

Wysłanie kampanii

Wywołanie: **POST /rest/campaigns/send**

Dane w POST:

hash	Wymagane	Hash kampanii którą chcemy wysłać
time	Opcjonalne	Data wysłania kampanii w formacie YYYY-MM-DD H:i:s

Kody błędów:

1731	Brak hasha lub jest on niepoprawny.
1732	Czas wysyłki jest niepoprawny
1733	Czas wysyłki nie może być wcześniejszy niż obecna godzina.
1734	Kampania o takim hashu nie istnieje.
1735	Kampania nie jest jeszcze gotowa do wysyłki testowej.
1736	Kampania już jest ustawiona do wysyłki lub jest w jej trakcie.

Subskrybenci – zarządzanie subskrybentami

Kontroler służy do dodawania i aktualizowania danych o subskrybentach w listach subskrypcyjnych.

Kody statusów subskrybentów

Opis słowny	Wartość	Objaśnienie
Aktywny	**1**	Subskrybent aktywny do którego mogą być wysyłane kampanie
Do aktywacji	**2**	Subskrybent który musi kliknąć w link akceptacyjny w mailu potwierdzającym jego dodanie do listy
Nieaktywowany	**3**	Subskrybent który po okresie kilku dni nie kliknął w link akceptacyjny w mailu potwierdzającym
Wypisany	**4**	Subskrybent który wypisał się z subskrypcji
Odbijający „miękko”	**5**	Subskrybent którego skrzynka kilka razy z rzędu zwróciła maila z odpowiednim kodem (http://www.ietf.org/rfc/rfc1893.txt)
Odbijający „twardo”	**8**	Subskrybent którego skrzynka kilka razy z rzędu zwróciła maila z odpowiednim kodem (http://www.ietf.org/rfc/rfc1893.txt)

Dodanie subskrybenta

Metoda dodaje subskrybenta jeśli nie było go w liście subskrypcyjnej lub aktualizuje jego status i dane jeśli nie był w statusie „**Aktywny**” oraz „**Do aktywacji**”

Wywołanie: **POST /rest/subscriber/add**

Dane w POST:

email	Wymagane	Adres email subskrybenta
list	Wymagane	Hash list subskrypcyjnej
state	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste zostanie nadany status „**Do aktywacji**”, lista dostępnych statusów w dziale „**Kody Statusów Subskrybentów**”
confirm	Opcjonalne	Czy do subskrybenta ma być wysłany email potwierdzający. Możliwe wartości to **0** oraz **1**. Jeśli pole będzie puste zostanie wysłany email z potwierdzeniem. To pole jest ignorowane (email nie zostanie wysłany) jeśli nadamy subskrybentowi status inny niż „**Do aktywacji**”.
custom_fields	Opcjonalne	Tablica z polami dodatkowymi i wartościami w postaci: {"tag personalizacji pola 1":"wartość", "tag personalizacji pola 2":"wartość"}

Kody błędów:

1301	Adres email jest niepoprawny
1302	Lista subskrypcyjna nie istnieje lub brak hash'a listy
1303	Jedno lub więcej pól dodatkowych jest niepoprawne
1304	Subskrybent już istnieje w tej liście subskrypcyjnej i ma status **Aktywny** lub **Do aktywacji**
1305	Próbowano nadać niepoprawny status subskrybenta

Modyfikowanie danych subskrybenta

Metoda aktualizuje status i dane subskrybenta

Wywołanie: **POST /rest/subscriber/edit**

Dane w POST:

email	Wymagane	Adres email subskrybenta
list	Wymagane	Hash list subskrypcyjnej
state	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste aktualny status subskrybenta nie zostanie zmieniony
custom_fields	Opcjonalne	Tablica z polami dodatkowymi i wartościami w formacie: {"tag personalizacji pola 1":"wartość", "tag personalizacji pola 2":"wartość"}

Kody błędów:

1331	Subskrybent nie istnieje
1302	Lista subskrypcyjna nie istnieje lub brak hash'a listy
1305	Próbowano nadać niepoprawny status subskrybenta

Pobieranie danych subskrybenta

Wywołanie: **GET /rest/subscriber/get/[list]/[email]**

Dane w GET:

email	Wymagane	Adres email subskrybenta
list	Wymagane	Hash list subskrypcyjnej

Zwracane dane:

email	Adres email subskrybenta
state	Status subskrybenta (zgodny z opisem w dziale „**Kody Statusów Subskrybentów**”)
custom_fields	Ilość subskrybentów do których była wysyłka
sent	Data wysyłki kampanii, format: rrrr-mm-dd hh:ii:ss

Kody błędów:

1311	Adres email jest niepoprawny
1312	Lista subskrypcyjna nie istnieje

Kasowanie subskrybenta

Metoda kasuje subskrybenta z listy subskrypcyjnej

Wywołanie: **POST /rest/subscriber/delete**

Dane w POST:

email	Wymagane	Adres email subskrybenta
list	Wymagane	Hash list subskrypcyjnej

Kody błędów:

1321	Adres email nie istnieje w tej liście subskrypcyjnej
1322	Lista subskrypcyjna nie istnieje lub brak hash'a listy

Pobieranie historii akcji subskrybenta

Metoda pobiera ostatnie akcje subskrybenta.

Wywołanie: POST **/rest/subscriber/getHistory**

Dane w POST:

email	Wymagane	Adres email subskrybenta
list	Wymagane	Hash list subskrypcyjnej
limit	Opcjonalne	Ilość pobranych akcji subskrybenta (domyślnie 10)

Przykładowy request:

```
{
	"email":"tester1@freshmail.pl"
	"list":"4zcnmd2ski"
	"limit":"30"
}
```

Zwracane dane:

name	Nazwa kampanii
email_topic	Temat kampanii
scheduled_sent	Data i czas wysłania kampanii (format: rrrr-mm-dd hh:ii:ss)
state	Status subskrybenta (dostarczony, odbicie miękkie, odbicie twarde)
opens_count	Ilość otwarć subskrybenta
clicks_count	Ilość kliknięć subskrybenta

Kody błędów:

1311	Adres email jest niepoprawny
1312	Lista subskrypcyjna nie istnieje
1313	Subskrybent nie istnieje w podanej liście subskrypcyjnej
1314	Wartość atrybutu limit jest niepoprawna
1315	Subskrybent nie posiada akcji

Przykład odpowiedzi niepoprawnej:


```
{"status":"ERROR",
"errors":[{
        "message":"Subscriber doesn't exist",
        "code":1313
}]}
```


Przykład odpowiedzi poprawnej:


```
{"status":"OK",
"data":[{
        "name":"4 kampania wtorkowa z dnia 30.07.2013",
        "email_topic":"Jesienna promocja",
        "scheduled_sent":"2013-07-30 21:37:42",
        "state":"DELIVERED",
        "opens_count":"1",
        "clicks_count":"2"
}]}
```

Subskrybenci – zarządzanie wieloma subskrybentami w pojedynczym żądaniu

Za pomocą poniższych metod można wykonywać dokładnie te same działania co przy zarządzaniu pojedynczymi subskrybentami, ale w żądaniu można przesłać do 100 subskrybentów naraz.

Dodawanie wielu subskrybentów

Metoda dodaje wielu subskrybentów jeśli nie było ich w liście subskrypcyjnej lub aktualizuje ich status i dane jeśli nie był w statusie „**Aktywny**” oraz „**Do aktywacji**”. W przypadku dodania tylko części z adresów (np. jeden z adresów jest niepoprawny) zostaje zwrócony kod 200 oraz dane dotyczące błędnego/błędnych adresów.

Wywołanie: **POST /rest/subscriber/addMultiple**

Dane w POST:

subscribers	Wymagane	Tablica z adresami email i polami dodatkowymi w postaci {"email" : "adres_email", "custom_fields": { "tag personalizacji pola 1" : "wartość", "tag personalizacji pola 1" : "wartość" } } Pole ”custom_fields” nie jest obowiązkowe.
list	Wymagane	Hash list subskrypcyjnej
state	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste zostanie nadany status „**Do aktywacji**”, lista dostępnych statusów w dziale „**Kody Statusów Subskrybentów**”
confirm	Opcjonalne	Czy do subskrybenta ma być wysłany email potwierdzający. Możliwe wartości to **0** oraz **1**. Jeśli pole będzie puste zostanie wysłany email z potwierdzeniem. To pole jest ignorowane (email nie zostanie wysłany) jeśli nadamy subskrybentowi status inny niż „**Do aktywacji**”.

Przykładowy request:

```
{"list":"id_hash",
 "subscribers" : [ {"email":"tester1@freshmail.pl",
                    "custom_fields":{"imie":"Hanna",
                                     "kod_promocyjny":"abcde"} },
                   {"email":"tester2@freshmail.pl",
                    "custom_fields":{"imie":"Anna",
```

"kod_promocyjny":"efgh"} },
{"email":"tester3@freshmail.pl",
"custom_fields":{"imie":"Agata",
"kod_promocyjny":"wxyz"} }] }

Kody błędów:

1331	Nie udało się dodać/aktualizować żadnego subskrybenta
1332	Lista subskrypcyjna nie istnieje lub brak hash'a listy
1335	Próbowano nadać niepoprawny status subskrybenta
1336	Nie przekazano adresów email do dodania
1399	Za dużo subskrybentów w jednym żądaniu – limit 100

Przykład częściowych błędów, wysłany request:

{"list":"4zcnmd2ski",
"subscribers":[{"email":"poprawny@adres.email.pl"},
{"email":"niepoprawny adres email"}]}

Zwracane dane (kod odpowiedzi 200):

{"status":"OK",
"data":{
"inserted":1,
"not_inserted":1,
"errors":[{"email":"niepoprawny adres email",
"error":"Email address not correct",
"code":1301}]}}

Kody błędów zwracane dla poszczególnych subskrybentów są identyczne jak kody błędów dla metody **/rest/subscriber/add**

Modyfikowanie danych wielu subskrybentów

Metoda aktualizuje status i dane wielu subskrybentów

Wywołanie: **POST /rest/subscriber/editMultiple**

Dane w POST:

subscribers	Wymagane	Tablica z adresami email i polami dodatkowymi w postaci {"email" : "adres_emaiil", "custom_fields": { "tag personalizacji pola 1" : "wartość", "tag personalizacji pola 1" : "wartość" } } Pole ”custom_fields” nie jest obiwiązkowe.
list	Wymagane	Hash list subskrypcyjnej
state	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste aktualny status subskrybenta nie zostanie zmieniony

Kody błędów:

1331	Nie udało się dodać/aktualizować żadnego subskrybenta
1332	Lista subskrypcyjna nie istnieje lub brak hash'a listy
1335	Próbowano nadać niepoprawny status subskrybenta
1336	Nie przekazano adresów email do zaktualizowania
1399	Za dużo subskrybentów w jednym żądaniu – limit 100

Zwracane dane/informacje o błędach są identyczne jak w przypadku metody **/rest/subscriber/addMultiple**

Modyfikowanie pól dodatkowych wszystkich subskrybentów

Metoda aktualizuje wybrane pole dodatkowe i ustawia ustaloną wartość dla wszystkich subskrybentów

Wywołanie: **POST /rest/subscriber/updateFieldValue**

Dane w POST:

listHash	Wymagane	Hash list subskrypcyjnej
tag	Wymagane	Tag pola dodatkowego (np. kod_promocyjny)
value	Wymagane	Wartość jak zostanie ustawiona w danym polu dodatkowym dla wszystkich subskrybentów
url	Wymagane	Adres www na który ma zostać wysłana odpowiedź po zakończeniu procesu aktualizacji

Zwracane dane:

id	Id procesu

Kody błędów:

1381	Pole dodatkowe nie istnieje
1382	Brak hasha listy lub tag'a pola dodatkowego
1383	Niepoprawny adres url
1384	Zbyt duża liczba znaków w polu „value” - max. 255
1385	Pole „value” jest wymagane
1386	Proces aktualizacji tego pola wciąż trwa, poczekaj na jego zakończenie

Przykład odpowiedzi niepoprawnej:


```
{"status":"ERROR",
    "errors":[{
        "message":"Niepoprawny adres url",
        "code":1383
}]}
```


Przykład odpowiedzi poprawnej (kod odpowiedzi 200):


```
{"status":"OK",
    "data":{
        "id":5
```

}}

Po procesie aktualizacji zostanie wysłane żądanie na wskazany adres URL. Odpowiedz będzie żądaniem typu POST, w treści żądania znajdą się następujące dane:

id	Id procesu
updated	Ilość zaktualizowanych pól subskrybentów

Przykładowe żądanie:

```
{"status":"OK",
        "data":{
                "id":5,
                "updated":150
}}
```

W odpowiedzi skrypt oczekuje kody se statusem „OK”:
{"status":"OK"}

Jeśli powyższa odpowiedz nie będzie poprawna, skrypt ponowi kilka razy próbę komunikacji, przy większej ilości błędów nastąpi zaniechanie wysyłania kolejnych żądań.

Pobieranie danych wielu subskrybentów

W przypadku gdy request jest częściowo niepoprawny (np. część adresów email nie istnieje) lub całkowicie niepoprawny (np. żaden z adresów email nie istnieje) zostaje zwrócony kod 200 oraz dane dotyczące błędnego/błędnych adresów.

Wywołanie: POST **/rest/subscriber/getMultiple**

Dane w GET:

subscribers	Wymagane	Tablica z adresami email użytkowników: [{"email":"tester1@freshmail.pl"}, {"email":"tester2@freshmail.pl"}, {"email":"tester3@freshmail.pl"}]
list	Wymagane	Hash list subskrypcyjnej

Zwracane dane:

email	Adres email subskrybenta
state	Status subskrybenta (zgodny z opisem w dziale „**Kody Statusów Subskrybentów**”)
custom_fields	Pola dodatkowe wraz z wartościami
errors	Tablica z błędami (w przypadku gdy któryś z adresów nie istnieje)

Kody błędów:

1341	Lista subskrypcyjna nie istnieje
1342	Tablica z subskrybentami jest pusta
1399	Za dużo subskrybentów w jednym żądaniu – limit 100
1336	Nie przekazano adresów odbiorców do pobrania

Kasowanie wielu subskrybentów

Metoda kasuje subskrybentów z listy subskrypcyjnej

Wywołanie: **POST /rest/subscriber/deleteMultiple**

Dane w POST:

subscribers	Wymagane	Tablica z adresami email użytkowników: [{"email":"tester1@freshmail.pl"}, {"email":"tester2@freshmail.pl"}, {"email":"tester3@freshmail.pl"}]
list	Wymagane	Hash list subskrypcyjnej

Kody błędów:

1351	Lista subskrypcyjna nie istnieje
1352	Żaden z subskrybentów nie istnieje/nie został skasowany
1399	Za dużo subskrybentów w jednym żądaniu – limit 100

Zwracane dane/informacje o błędach są identyczne jak w przypadku metody /rest/subscriber/addMultiple

Dodawanie subskrybentów do listy odbiorców blokowanych

Metoda dodaje adresy subskrybentów do listy odbiorców blokowanych.

Wywołanie: POST **/rest/subscriber/addBlocks**

Dane w POST:

emails	Wymagane	Tablica z adresami email użytkowników: { "emails":["tester1@freshmail.pl", "tester2@freshmail.pl", "tester3@freshmail.pl"] }

Zwracane dane:

stats['added']	Ilość dodanych odbiorców
stats['existing']	Ilość już istniejących odbiorców
stats['incorrect']	Ilość błędnych odbiorców
Incorrect_mails	Tablica z błędnymi adresami (w przypadku gdy jakiś mail był błędny)

Kody błędów:

1301	Adresy odbiorców są niepoprawne
1336	Nie przekazano adresów odbiorców do dodania
1343	Format przekazanych danych jest niepoprawny

Kasowanie subskrybentów z listy odbiorców blokowanych

Metoda kasuje subskrybentów z listy odbiorców blokowanych. Akcja możliwa jedynie przy adresach zablokowanych własnoręcznie.

Wywołanie: POST **/rest/subscriber/removeBlocks**

Dane w POST:

emails	Wymagane	Tablica z adresami email użytkowników: { "emails":["tester1@freshmail.pl", "tester2@freshmail.pl", "tester3@freshmail.pl"]}

Zwracane dane:

stats['removed']	Ilość skasowanch odbiorców
stats['not_existing']	Ilość nieistniejących odbiorców
stats['incorrect']	Ilość błędnych odbiorców

Kody błędów:

1336	Nie przekazano adresów odbiorców do dodania
1343	Format przekazanych danych jest niepoprawny

Tworzenie kont

Metoda tworzy konto użytkownika. Można utworzyć zarówno konto główne jak i konto podrzędne.

Wywołanie: **POST /rest/account/create**

Dane w POST:

login	Wymagane	Login nowego użytkownika (adres email)
password	Wymagane	Hasło którym użytkownik będzie się logował
firstname	Wymagane	Imię użytkownika
lastname	Wymagane	Nazwisko użytkownika
company	Opcjonalne	Nazwa firmy
phone	Wymagane	Numer telefonu
activation_mail	Opcjonalne	Czy ma być wysłany mail aktywacyjny (domyślnie true)
activation	Opcjonalne	Czy konto ma wymagać aktywacji (domyślnie true)
child_account	Opcjonalne	Czy konto ma być kontem podrzędnym (domyślnie true)

Zwracane dane:

hash	Hash nowo założonego konta
api_key	Api key nowo założonego konta
api_secret	Secret nowo założonego konta

Kody błędów:

1501	Brak loginu (adres email)
1502	Nieprawidłowy login
1503	Login już istnieje w systemie
1504	Brak hasła
1505	Hasło jest za słabe - musi zawierać co najmniej 8 znaków - w tym jedną literę, jedną liczbę oraz jeden znak specjalny
1506	Brak imienia

1507 Brak nazwiska

1509 Brak telefonu

1510 Wystąpił błąd podczas tworzenia konta

1511 Konto nie zostało utworzone.

1512 Nie udało się przypisać konta do aplikacji

1513 Brak uprawnień do tworzenia kont.

Listy subskrypcyjne - zarządzanie

Kontroler służy do zarządzania listami subskrypcyjnymi. Umożliwia ich dodawanie, modyfikowanie, listowanie oraz usuwanie.

Tworzenie list

Funkcja służy do tworzenia list subskrypcyjnych. Zwraca hash listy, którym można się dalej posługiwać.

Wywołanie: **/rest/subscribers_list/create**

Dane:

name	Wymagane	Nazwa listy subskrypcyjnej
description	Opcjonalne	Opis listy subskrypcyjnej
custom_fields	Opcjonalne	Tablica definiująca pola dodatkowe (wedle opisu z punktu ***Tworzenie pól dodatkowych w listach***)

Zwracane dane:

hash	Hash nowej listy subskrypcyjnej
custom_fields	Tablica z pełnymi opisami utworzonych pól – jeśli miały być utworzone.

Kody błędów:

1601	Pusta wartość w polu name
1602	Nie udało się utworzyć listy subskrypcyjnej
1603	Błędna lista pól dodatkowych.
1604	Tag personalizacji powinien składać się jedynie z liter, cyfr i znaku podkreślenia.
1605	Nieprawidłowy typ pola.
1606	Osiągnięto maksymalną ilość list na koncie

Modyfikowanie list

Funkcja służy do modyfikowania list subskrypcyjnych.

Wywołanie: **/rest/subscribers_list/update**

Dane:

hash	Wymagane	Hash listy subskrybcyjnej
name	Wymagane	Nazwa listy subskrypcyjnej
description	Opcjonalne	Opis listy subskrypcyjnej

Kody błędów:

1611	Pusta wartość w polu name lub brak parametru name
1604	Brak uprawnień do listy

Usuwanie list

Funkcja służy do usuwania list subskrypcyjnych.

Wywołanie: **/rest/subscribers_list/delete**

Dane:

hash	Wymagane	Hash listy subskrybcyjnej

Kody błędów:

1604	Brak uprawnień do listy
1605	Nie udało się usunąć listy subskrypcyjnej

Pobieranie danych o listach

Funkcja służy do listowania wszystkich list subskrypcyjnych. Zwraca tablice z danymi wszystkich list do których konto ma uprawnienia.

Wywołanie: **/rest/subscribers_list/lists**

Zwracane dane:

hash	Hash listy subskrypcyjnej
name	Nazwa listy subskrypcyjnej
description	Opis listy subskrypcyjnej
creation_date	Data utworzenia listy subskrypcyjnej
subscribers_number	Liczba aktywnych subskrybentów
list_type	Typ listy, możliwe wartości to: „single opt-in” oraz „double opt-in”

Tworzenie pól dodatkowych w listach

Funkcja służy do tworzenia pól dodatkowych w listach subskrypcyjnych

Wywołanie: **/rest/subscribers_list/addField**

Dane:

hash	Wymagane	Hash listy subskrybcyjnej
name	Wymagane	Nazwa pola w liście
tag	Opcjonalne	Tag pod którym wartości z tego pola będą dostępne w mailach (w przypadku braku jest tworzony na podstawie nazwy pola)
type	Opcjonalne	Typ pola – 0 tekstowe, 1 numeryczne (domyślnie tekstowe)

Zwracane dane:

id_hash	Hash pola dodatkowego
field_name	Nazwa pola dodatkowego
personalization_tag	Tag pod którym wartości z tego pola będą dostępne w mailach
field_type	Typ pola 0 – tekstowe, 1 - numeryczne

Kody błędów:

1622	Błędny lub pusty hash listy subskrypcyjnej
1623	Błędna lub pusta nazwa pola
1624	Błędna nazwa tagu
1625	Błędny typ pola
1626	Pole dodatkowe o takim tagu personalizacji już istnieje

Wyświetlenie listy pól dodatkowych w podanej liście

Funkcja służy do wyświetlenia wszystkich pól dodatkowych w wybranej liście subskrypcyjnej

Wywołanie: **/rest/subscribers_list/getFields**

Dane:

hash	Wymagane	Hash listy subskrypcyjnej

Zwracane dane:

hash	Hash pola dodatkowego
name	Nazwa pola dodatkowego
tag	Tag pod którym wartości z tego pola będą dostępne w mailach
type	Typ pola

Kody błędów:

1632	Błędny lub pusty hash listy subskrypcyjnej

Spamtesty

Kontroler służy do sprawdzania maili pod kątem prawdopodobieństwa wpadnięcia wiadomości do spamu.

Sprawdzenie zgodności z filtrami SpamAssassin

Medota służy do przetestowania ile punktów spamowych oraz za co dostanie Twoja wiadomość w filtrze SpamAssassin.

Pamiętaj: Nie wszystkie systemy pocztowe korzystają z tego oprogramowania, wynik jest poglądowy i przy innej konfiguracji programów pocztowych może się różnić od pokazanego w tym teście. Wynik pozytywny (poniżej 5.0 punktów spamowych NIE oznacza, że inne filtry antyspamowe nie zatrzymają twojej wiadomości).

Wywołanie: **/rest/spam_test/check**

Dane:

subject	Wymagane	Temat maila
from	Opcjonalne	Adres nadawcy
from_name	Opcjonalne	Nazwa nadawcy
html	Opcjonalne *)	Treść html
text	Opcjonalne *)	Treść tekstowa

*) **Co najmniej jedno z wyróżnionych pól musi być wypełnione**

Zwracane dane:

tests	Lista testów antyspamowych których mail nie przeszedł
score	Sumaryczna punktacja za testy antyspamowe

Kody błędów:

1901	Brak tematu
1902	Niepoprawny adres nadawcy „from”
1904	Brak treści maila
1906	Błąd systemu sprawdzania.
1907	Przekroczono limit testów w ciągu jednego dnia