

SALESmanago

Dokumentacja Techniczna API

Wersja: 1.38
Ostatnia modyfikacja: 28 lutego 2014
Aktualizował: Konrad Pawlus

Spis treści

Informacje ogólne.....	3
Integracja kodu monitorującego.....	3
Dodawanie kontaktu i monitoring.....	4
Dodawanie nowego lub modyfikacja istniejącego kontaktu.....	4
Dodawanie nowego kontaktu.....	7
Przesyłanie dodatkowych informacji monitorujących.....	8
Modyfikacja istniejącego kontaktu.....	8
Dodawanie wielu kontaktów na raz.....	9
Usuwanie kontaktu.....	10
Sprawdzanie czy kontakt jest już zarejestrowany.....	11
Zarządzanie listą wysyłkową (Opt-in / Opt-out).....	11
Wypisywanie kontaktu z listy (Opt-out).....	11
Zapisywanie kontaktu do listy (Opt-in).....	12
Masowe wypisywanie kontaktu z listy (Opt-out).....	12
Masowe zapisywanie kontaktu do listy (Opt-in).....	13
Wypisywanie kontaktu z listy telefonicznej (Opt-out phone).....	14
Zapisywanie kontaktu do listy telefonicznej (Opt-in phone).....	15
Import listy kontaktów.....	15
Import po adresie e-mail.....	15
Import po ID kontaktu.....	18
Import listy ostatnio zmodyfikowanych kontaktów.....	19
Aktywność kontaktów.....	20
Zdarzenia zewnętrzne.....	23
Dodawanie zdarzenia zewnętrznego.....	23
Modyfikacja zdarzenia.....	24
Usuwanie zdarzenia.....	25
Przykładowa integracja koszyka.....	26
Import listy tagów.....	26
Monitorowanie zdarzeń AJAX/JavaScript.....	27
Edycja listy tagów kontaktu.....	27
Wysyłka e-maili.....	28
Zadania kontaktów.....	29
Autoryzacja tymczasowa.....	31
Rejestracja konta (tylko partnerzy).....	31

Informacje ogólne

Każde zdalne wywołanie API, z wyjątkiem autoryzacji musi zawierać dane autoryzacyjne:

```
"clientId":"your-client-id-123",  
"apiKey":"your-api-key-123",  
"requestTime":1327056031488,  
"sha":"2aa3927a7dee8c2a712adb5375f5fa36dd8fe00c",
```

Wartości **clientId** znajduje się na zakładce „Integracja” menu „Ustawienia”. Wartość **apiKey** to losowy ciąg znaków służący do autoryzacji.

Wartość **sha** generowana jest algorytmem SHA-1 z łańcucha powstałego z połączenia: **apiKey** + **clientId** + **apiSecret**. **apiSecret** znajduje się na wspomnianej zakładce „Integracja”.

requestTime to czas o jakim wykonywane jest zapytanie.

Wszystkie zapytania wykonywane muszą być na adresie: <http://www.salesmanago.pl/api>
Np. <http://xyz.salesmanago.pl/api/contact/upsert>

Gdzie **xyz** to identyfikator Twojego serwera (www lub app1, app2, app3...)

Ważne:

W zapytaniu należy przekazać następujące nagłówki:

```
Accept: application/json, application/json  
Content-Type: application/json; charset=UTF-8
```

Zapytanie wysyłane jest metodą HTTP POST, w przypadku, gdy użyta ma być metoda HTTP GET jest to wyraźnie zaznaczone w instrukcji użycia danej metody.

W dokumentacji znakiem * oznaczone zostały pola wymagane.

Integracja kodu monitorującego

Po aktywacji naszego konta otrzymujemy dostęp do kodu monitorującego naszą witrynę. Kod znajdziemy na stronie głównej, widocznej zaraz po zalogowaniu (do czasu wykrycia przez SALESmanago kodu na naszej stronie), lub w menu „Ustawienia” > „Integracja”.

start

Dashboard:

Ważne! Integracja kodu monitorującego

Aby poznać potencjalnych klientów, którzy odwiedzają Państwa stronę WWW, **1**simy wkleić na monitorowane strony poniższy kod (przed zakończeniem sekcji </body>):

```
<script type="text/javascript">
var _smid = "o7gslwlc8o1e2ry1";
(function() {
var sm = document.createElement('script'); sm.type = 'text/javascript'; sm.async = true;
sm.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + 'www.salesmanago.pl/static/sm.js';
var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(sm, s);
})();
</script>
```

Jeżeli już zintegrowali Państwo kod z witryną wiadomość ta zniknie automatycznie po wykryciu pierwszych wizyt. W razie problemów technicznych z wdrożeniem chętnie udzielimy dodatkowych informacji. tel. +48 531 756 629

Skrypt^[1] należy wgrać na każdą stronę naszego serwisu, zaraz przed zakończeniem sekcji „body”. Najlepiej, gdy istnieje możliwość skrypt ten wgrać w szablon naszego serwisu, tak by sam kod znajdował się w jednym miejscu, a był drukowany identycznie na wszystkich stronach.

Przykładowo, skrypt w CMS Wordpress można wgrać do pliku „footer.php”. Oto przykład:

Edytor

- Wtyczki 8
- Użytkownicy
- Narzędzia
- Ustawienia
- SEO 109
- Zwiń menu

```
</footer><!-- #colophon -->
</div><!-- #page -->
<?php wp_footer(); ?>
<script type="text/javascript">
var _smid = "o7gslwlc8o1e2ry1";
(function() {
var sm = document.createElement('script'); sm.type = 'text/javascript'; sm.async = true;
sm.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + 'www.salesmanago.pl/static/sm.js';
var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(sm, s);
})();
</script>
</body>
</html>
```

Dokumentacja:

Po kilkunastu minutach od wdrożenia, o ile na naszej stronie pojawiają się wizyty, informacja o potrzebie wdrożenia kodu zniknie ze strony głównej SALESmanago.

Dodawanie kontaktu i monitoring

Dodawanie nowego lub modyfikacja istniejącego kontaktu

Kontakt dodajemy wywołując metodę: <http://www.salesmanago.pl/api/contact/upsert>
Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "contact" : { "company" : "Benhauer Sp. z o.o. Sp. K.",
    "email" : "konrad-test-1@konri.com",
    "fax" : "+48345543345",
    "name" : "Konrad Test",
    "phone" : "+48123321123",
    "address":{
      "streetAddress":"Brzyczyńska 123",
      "zipCode":"43-305",
      "city":"Bielsko-Biała",
      "country":"PL"
    }
  },
  "owner" : "admin@vondor.pl",
  "newEmail" : "",
  "forceOptIn" : true,
  "forceOptOut" : false,
  "forcePhoneOptIn" : true,
  "forcePhoneOptOut" : false,
  "requestTime" : 1327059355361,
  "sha" : "08924f45afc2e4fb8b652c53cdb493c7ddb846a1",
  "tags" : [ "API",
    "ADmanago"
  ],
  "removeTags" : [ "Test_tag",
    "New"
  ],
  "properties":{"custom.nickname":"Konri","custom.sex":"M"},
  "birthday": "19801017"
}
```

Podstawowe elementy jakie można podać dodając nowy kontakt w strukturze to:

name – nazwa kontaktu

email – email kontaktu*

phone – numer telefonu

fax – numer fax

company – firma kontaktu

birthday – data urodzenia, przekazywana jako łańcuch znaków w postaci: yyyyMMdd lub Mmdd (yyyy – 4 cyfrowy rok, MM – dwucyfrowy miesiąc, dd – dwucyfrowy dzień)

address – adres kontaktu

streetAddress – ulica i numer domu

zipCode – kod pocztowy

city – miasto

country – kraj

dodatkowo do zapytania należy dołączyć informację o właścicielu kontaktu (koncie do którego kontakt zostanie przypisany):

owner – właściciel kontaktu (email konta użytkownika SALESmanago) *

Opcjonalnie możemy zmienić adres e-mail kontaktu. Należy wtedy wypełnić pole *newEmail*:

newEmail – nowy adres email (jeżeli chcemy go zmodyfikować)

Możemy również wymusić tzw. opt-in/opt-out kontaktu. Należy wtedy wypełnić pole *forceOptIn*:

forceOptOut – wymuszenie opt-out po dodaniu/modyfikacji

forceOptIn – wymuszenie opt-in po dodaniu/modyfikacji (o ile nie wybrano wcześniejszej opcji)

forcePhoneOptOut – wymuszenie opt-out z telefonu po dodaniu/modyfikacji

forcePhoneOptIn – wymuszenie opt-in na telefon po dodaniu/modyfikacji (o ile nie wybrano wcześniejszej opcji)

W zapytaniu istnieje możliwość oznaczenia kontaktu tagami i usunięcia obecnych tagów. Tagi przesyłamy jako tablica łańcuchów tekstowych w polu *tags*

tags – tablica tagów kontaktu

removeTags – tablica tagów do usunięcia

Istnieje również możliwość przypisać do kontaktu dowolną ilość pól definiowanych przez użytkownika. Przesyłamy ją za pomocą mapy:

properties – definiowane przez użytkownika atrybuty kontaktu. Zaleca się nie stosować polskich znaków oraz spacji w nazwie, ale jest to dozwolone.

* pola wymagane.

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "contactId" : "21c252a6-6de0-436b-bae8-9d0142363266",
  "message" : [ ],
  "success" : true
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

contactId – to unikatowy identyfikator uaktualnionego lub nowo dodanego kontaktu

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Dodawanie nowego kontaktu

Kontakt dodajemy wywołując metodę: <http://www.salesmanago.pl/api/contact/insert>

Metoda ta nie aktualizuje istniejącego kontaktu.

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "contact" : { "company" : "Benhauer Sp. z o.o. Sp. K.",
    "email" : "konrad-test-1@konri.com",
    "fax" : "+48345543345",
    "name" : "Konrad Test",
    "phone" : "+48123321123",
    "address":{
      "streetAddress":"Brzyczyńska 123",
      "zipCode":"43-305",
      "city":"Bielsko-Biała",
      "country":"PL"
    }
  },
  "owner" : "admin@vendor.pl",
  "newEmail" : "",
  "forceOptOut" : false,
  "forcePhoneOptOut" : false,
  "requestTime" : 1327059355361,
  "sha" : "08924f45afc2e4fb8b652c53cdb493c7ddb846a1",
  "tags" : [ "API",
    "ADmanago"
  ],
  "properties":{"custom.nickname":"Konri","custom.sex":"M"},
  "birthday": "19801017"
}
```

Podstawowe elementy jakie można podać dodając nowy kontakt są identyczne jak w metodzie upsert, z wyjątkiem braku pól: forceOptIn, forcePhoneOptIn i removeTags.

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "contactId" : "21c252a6-6de0-436b-bae8-9d0142363266",
  "message" : [ ],
  "success" : true
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

contactId – to unikatowy identyfikator uaktualnionego lub nowo dodanego kontaktu

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Przesyłanie dodatkowych informacji monitorujących

W odpowiedzi na zapytanie **/contact/upsert** otrzymujemy unikatowy identyfikator. Aby usprawnić monitoring zachowania użytkowników na stronie WWW należy ten parametr przekazać przynajmniej na jednej podstronie odwiedzanej przez użytkownika – np. po zalogowaniu – w momencie logowania wykonujemy zapytanie **/contact/upsert** a otrzymaną wartość **contactId** drukujemy wraz z kodem JavaScript susyemu monitorowania SALESmanago.

Przykładowy kod wraz z zaznaczoną wartością contactId:

```
<script type="text/javascript">
    var _smid = "your-client-id-123";
    var _smclientid = "contactIdFromResponse-123";
</script>
<script src="http://www.salesmanago.pl/static/sm.js"
type="text/javascript"></script>
```

Alternatywnie, przesyłając formularz przez AJAX można zapisać samo ciasteczko:

```
$.ajax({
    type: 'POST',
    url: '/account/registerDemoVideo.htm',
    data: $("#registerForm").serialize(),
    cache: false,
    timeout: 240000,
    success: function (data) {
        if (data == 'Wrong_Email') {
            alert('Podany email nie jest prawidłowy. Proszę wpisać
ponownie.');
```

```
            $("#email").focus();
        } else {
            createCookie('smclient', data, 365 * 10);
            $(".registeredInfo").show();
        }
    },
    error: function (data) {
        alert("Błąd - nie udało się zarejestrować");
    }
});
```

Modyfikacja istniejącego kontaktu

Kontakt modyfikujemy wywołując metodę: <http://www.salesmanago.pl/api/contact/update>
Przykładowa struktura danych zapytania:

```
{
    "apiKey" : "your-api-key-123",
    "clientId" : "your-client-id-123",
    "email" : "konrad@salesmanago.pl",
    "contactId" : null,
    "contact" : { "company" : "Benhauer Sp. z o.o. Sp. K.",
        "email" : "konrad-test-1@konri.com",
        "fax" : "+48345543345",
        "name" : "Konrad Test",
```



```

    "phone" : "+48123321123",
    "address":{
      "streetAddress":"Brzyczyńska 123",
      "zipCode":"43-305",
      "city":"Bielsko-Biała",
      "country":"PL"
    }
  },
  "owner" : "admin@vendo.pl",
  "forceOptIn" : true,
  "forceOptOut" : false,
  "forcePhoneOptIn" : true,
  "forcePhoneOptOut" : false,
  "requestTime" : 1327059355361,
  "sha" : "08924f45afc2e4fb8b652c53cdb493c7ddb846a1",
  "tags" : [ "API", "ADmanago"],
  "removeTags" : [ "Test_tag", "New"],
  "properties":{"custom.nickname":"Konri","custom.sex":"M"},
  "birthday": "1017"
}

```

Dane zapytania są identyczne z metodą *upsert* opisaną wyżej. Kontakt identyfikuje *email* lub *contactId* (zwrócony przy dodawaniu kontaktu).

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON identycznej jak w przypadku metody *upsert*.

W przypadku podania *emaila* w polu *contact* adres e-mail zostanie zaktualizowany.

Dodawanie wielu kontaktów na raz

Wiele kontaktów naraz dodajemy wywołując metodę:

<http://www.salesmanago.pl/api/contact/batchupsert>

Przykładowa struktura danych zapytania:

```

{
  "clientId":"your-client-id-123",
  "apiKey":"your-api-key-123",
  "requestTime":1348046897664,
  "sha":"8d893f41dd479bb0489686f04b0a169005d22559",
  "owner":"admin@vendo.pl",
  "upsertDetails":[
    {
      "contact":{
        "email":"batchtest2@benhauer.pl",
        "name":"Konrad Test1",
        "phone":"+48123321123",
        "fax":"+48345543345",
        "company":"Benhauer Sp. z o.o. Sp. K.",
        "externalId":null,
        "address":{
          "streetAddress":"Brzyczyńska 123",
          "zipCode":"43-305",
          "city":"Bielsko-Biała",
          "country":"PL"
        }
      }
    },
    "tags":["API", "ADmanago"],

```

```
    "removeTags":["Test_tag", "New"],
    "properties":{
      "custom.nickname":"Konril",
      "custom.sex":"M"
    },
    "birthday": "19801017"
  },
  {
    "contact":{
      "email":null,
      "name":"Konrad Test2",
      "phone":"+48123321123",
      "fax":"+48345543345",
      "company":"Benhauer Sp. z o.o. Sp. K.",
      "externalId":null
    },
    "newEmail":"batchtestNew@benhauer.pl",
    "forceOptIn" : true,
    "forceOptOut" : false,
    "forcePhoneOptIn" : true,
    "forcePhoneOptOut" : false,
    "tags":["API", "ADmanago"],
    "properties":{
      "custom.nickname":"Konri2",
      "custom.sex":"M"
    }
  }
}]}
```

W zapytaniu podajemy tablicę kontaktów w polu *upsertDetails*. Podstawowe elementy jakie można podać w elemencie tablicy *contact* można znaleźć powyżej w opisie metody *upsert* (patrz [Dodawanie kontaktu](#)).

W odpowiedzi otrzymujemy status zapytania oraz w przypadku sukcesu listę identyfikatorów dla kontaktów dodanych.

```
{
  "success":true,
  "message":[],
  "contactIds":{
    "batchtest2@benhauer.pl":"b257d328-2a95-41ce-915a-94b4274e6c29",
    "batchtest1@benhauer.pl":"bf4d6c03-1ca2-4b3f-8131-c28829236b02"
  }
}
```

Usuwanie kontaktu

Kontakt usuwamy wywołując metodę: <http://www.salesmanago.pl/api/contact/delete>
Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "email" : "test@salesmanago.pl",
  "owner" : "admin@vendor.pl",
  "permanently": true/false
}
```

```
}
```

Kontakt można skasować permanentnie lub tylko oznaczyć jako usunięty w tym celu ustawiamy pole:

permanently – definiuje czy kontakt ma zostać skasowany całkowicie, czy tylko oznaczony jako usunięty

Sprawdzanie czy kontakt jest już zarejestrowany

Czy kontakt jest już zapisany w bazie możemy sprawdzić wywołując metodę:

<http://www.salesmanago.pl/api/contact/hasContact>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "email" : "test@salesmanago.pl",
  "owner" : "admin@vendor.pl"
}
```

Zwrócony zostanie status true/false informujący o zapisie kontaktu a w przypadku gdy kontakt już istnieje dodatkowo zwrócony zostanie jego ID.

Zarządzanie listą wysyłkową (Opt-in / Opt-out)

Wypisywanie kontaktu z listy (Opt-out)

Kontakt wypisujemy z listy wysyłek wywołując metodę:

<http://www.salesmanago.pl/api/contact/optout>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",
  "email" : "konrad-test-optout-1@konri.com"
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

email – email kontaktu*

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "contactId" : "21c252a6-6de0-436b-bae8-9d0142363266",
  "message" : [ ],
  "success" : true
}
```

```
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)
contactId – to unikatowy identyfikator uaktualnionego kontaktu
message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Zapisywanie kontaktu do listy (Opt-in)

Kontakt dodajemy na listę wysyłek wywołując metodę:

<http://www.salesmanago.pl/api/contact/optin>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",
  "email" : "konrad-test-optout-1@konri.com"
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

email – email kontaktu*

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "contactId" : "21c252a6-6de0-436b-bae8-9d0142363266",
  "message" : [ ],
  "success" : true
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)
contactId – to unikatowy identyfikator uaktualnionego kontaktu
message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Masowe wypisywanie kontaktu z listy (Opt-out)

Kontakty masowo wypisujemy z listy wysyłek wywołując metodę:

<http://www.salesmanago.pl/api/contact/batchoptout>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
```

```
"clientId" : "your-client-id-123",
"requestTime" : 1329128188409,
"sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",
"emails" : {
  "batchtest2@benhauer.pl", batchtest1@benhauer.pl"
}
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

emails – emaile kontaktów*

W odpowiedzi otrzymujemy status zapytania oraz w przypadku sukcesu listę identyfikatorów dla kontaktów wypisanych.

```
{
  "success":true,
  "message":[],
  "contactIds":{
    "batchtest2@benhauer.pl":"b257d328-2a95-41ce-915a-94b4274e6c29",
    "batchtest1@benhauer.pl":"bf4d6c03-1ca2-4b3f-8131-c28829236b02"
  }}
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

contactIds – to unikatowe identyfikatory uaktualnionych kontaktów

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Masowe zapisywanie kontaktu do listy (Opt-in)

Kontakty masowo dopisujemy do listy wysyłając wywołując metodę:

<http://www.salesmanago.pl/api/contact/batchoptin>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",
  "emails" : {
    "batchtest2@benhauer.pl", batchtest1@benhauer.pl"
  }
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

emails – emaile kontaktów*

W odpowiedzi otrzymujemy status zapytania oraz w przypadku sukcesu listę

identyfikatorów dla kontaktów zapisanych.

```
{
  "success": true,
  "message": [],
  "contactIds": {
    "batchtest2@benhauer.pl": "b257d328-2a95-41ce-915a-94b4274e6c29",
    "batchtest1@benhauer.pl": "bf4d6c03-1ca2-4b3f-8131-c28829236b02"
  }
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

contactIds – to unikatowe identyfikatory uaktualnionych kontaktów

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Wypisywanie kontaktu z listy telefonicznej (Opt-out phone)

Kontakt wypisujemy z listy wysyłając wywołując metodę:

<http://www.salesmanago.pl/api/contact/phoneoptout>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",
  "email" : "konrad-test-optout-1@konri.com"
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

email – email kontaktu*

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "contactId" : "21c252a6-6de0-436b-bae8-9d0142363266",
  "message" : [ ],
  "success" : true
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

contactId – to unikatowy identyfikator uaktualnionego kontaktu

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Zapisywanie kontaktu do listy telefonicznej (Opt-in phone)

Kontakt dodajemy na listę wysyłek wywołując metodę:

<http://www.salesmanago.pl/api/contact/phoneoptin>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",
  "email" : "konrad-test-optout-1@konri.com"
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

email – email kontaktu*

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "contactId" : "21c252a6-6de0-436b-bae8-9d0142363266",
  "message" : [ ],
  "success" : true
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

contactId – to unikatowy identyfikator uaktualnionego kontaktu

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Import listy kontaktów

Import po adresie e-mail

Kontakty importujemy wywołując metodę: <http://www.salesmanago.pl/api/contact/list>.

Przykładowa struktura danych zapytania:

```
{ "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "email" : [ "****@gmail.com" ],
  "contactId" : [ "123-XYZ" ],
  "owner" : "admin@vondor.pl",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7"
}
```

Podstawowe elementy jakie należy podać importując listę kontaktów:

email – adresy email importowanych kontaktów*,
owner – adres email właściciela kontaktów*.

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{ "contacts" : [ { "contactEvents" : [ { "date" : 2012-02-13 10:38:53,
    "description" : "Otagowano kontakt. Użyte tagi: ADmanago",
    "detail1" : "ADmanago",
    "detail2" : "",
    "detail3" : "",
    "detail4" : "",
    "detail5" : ""
  }
],
  "contactTags" : [ { "tag" : "ADmanago",
    "tagName" : "ADMANAGO",
    "score" : 12,
  }
],
  "contactVisits" : [ {
    "vhost" : "salesmanago.pl",
    "time" : 2012-02-13 10:38:53,
    "duration" : 12345677,
    "visitSource" : "SEARCH_ENGINE",
    "visitSourceHost" : "www.google.pl",
    "visitSourceKeywords" : "SALESmanago",
    "visitHost" : null,
    "visitScore" : 78,
    "url" : null
  }
],
  "email" : "konrad\*\*\*\*@gmail.com",
  "emailMessages" : [ {
    "name" : "Moja wiadomość ",
    "subject" : "Dzień dobry ",
    "date" : 2012-02-13 10:38:53,
    "sent" : 1,
    "dateSent" : 2012-02-13 10:38:54,
    "opened" : 1,
    "dateOpened" : 2012-02-13 10:38:55,
    "clicked" : 1,
    "dateClicked" : 2012-02-13 10:38:56,
  }
],
  "incomingEmailMessages" : [ {
    "subject" : "Dzień dobry ",
    "date" : 2012-02-13 10:38:53,
  }
],
  "fax" : 12131415,
  "name" : "Konrad ****",
  "phone" : 111222333,
  "score" : 56,
  "company" : "Benhauer Sp. z o.o. Sp. K.,
```



```
"externalId" : "aaa-123,"
"address":{
  "streetAddress":"Brzyczyńska 123",
  "zipCode":"43-305",
  "city":"Bielsko-Biała",
  "country":"PL"
}
},
"message" : [ ],
"success" : true
```

gdzie:

contacts – importowane kontakty

contactEvents – zdarzenia kontaktu

date – data zdarzenia

description – opis zdarzenia

detail1 – detale zdarzenia

detail2 – detale zdarzenia

detail3 – detale zdarzenia

detail4 – detale zdarzenia

detail5 – detale zdarzenia

contactTags – tagi kontaktu

tagName – nazwa tagu

tag – identyfikator tagu

score – skoring tagu

contactVisits – wizyty kontaktu

host – strona którą odwiedzał kontakt

time – czas wizyty

duration – czas trwania

visitSource – źródło wizyty, możliwe wartości tego pola to:

EMAIL_CONVERSATION - wejście z maila (kliknięcie w link)

SEARCH_ENGINE - wejście z wyszukiwarki

ADVERTISEMENT - wejście z boks reklamowego (AdWords)

REFERRER - wejście z witryny odsyłającej

DIRECT - wejście bezpośrednie, lub przejście na kolejną stronę

visitSourceHost – host odsyłający (referrer)

visitSourceKeywords – słowa kluczowe

visitScore – liczba punktów

url – URL odwiedzanej strony

email – adres email kontaktu

emailMessages – wiadomości kontaktu

name – tytuł wiadomości

subject – temat wiadomości

date – data stworzenia

sent – stan wysłania

dateSent – data wysłania

opened – stan otwarcia

dateOpened – data otwarcia
clicked – stan kliknięcia
dateClicked – data kliknięcia
properties – dodatkowe detale kontaktu
name – nazwa detalu
value – wartość
contactFunnels – kampanie (lejki) sprzedażowe
salesFunnel – nazwa kampanii
salesStage – etap na którym jest kontakt
contactNotes - notatki
note – treść notatki
date – data dodania
priv – czy jest prywatna
contactTasks – zadania
note – treść notatki (zadania)
date – data zadania
cc – kopia powiadomienia (email)
reminder – data przypomnienia
fax – numer fax kontaktu
name – imię kontaktu
phone – numer telefonu
score – liczba punktów
address – adres kontaktu
streetAddress – ulica i numer domu
zipCode – kod pocztowy
city – miasto
country – kraj
company – firma kontaktu
externalId – zewnętrzny identyfikator kontaktu
success – informacja o poprawności zakończenia importu
message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Import po ID kontaktu

Kontakty po ID importujemy wywołując metodę:
<http://www.salesmanago.pl/api/contact/listById>.

Przykładowa struktura danych zapytania:

```
{ "apiKey" : "your-api-key-123",  
  "clientId" : "your-client-id-123",  
  "contactId" : [ "123-XYZ" ],  
  "owner" : "admin@vondor.pl",  
  "requestTime" : 1329128188409,  
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7"  
}
```

Podstawowe elementy jakie należy podać importując listę kontaktów:

contactId – identyfikatory importowanych kontaktów*,
owner – adres email właściciela kontaktów*.

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON identyczna z wcześniejszym zapytaniem.

Import listy ostatnio zmodyfikowanych kontaktów

Kontakty importujemy wywołując metodę:

<http://www.salesmanago.pl/api/contact/modifiedContacts>.

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "requestTime":1362056589362,
  "sha":"64656d78b80d5df677700dabd363e1ffe51b59a7",
  "owner":"admin@vondor.pl",
  "from":1359673200361,
  "to":1363042800362
}
```

Podstawowe elementy jakie należy podać importując listę kontaktów:

owner – adres email właściciela kontaktów*,
from – początkowy zakres dat modyfikacji*,
to – końcowy zakres dat modyfikacji*.

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "success":true,
  "message":[],
  "modifiedContacts":[
    {
      "id":"f66ca32b-c117-4b52-b3b8-863be077e710",
      "email":"aleksander.***@benhauer.pl"
    },
    {
      "id":"426e0ef8-675f-47fc-8ea8-745ac1706904",
      "email":"konrad.***@salesmanago.pl"
    },
    {
      "id":"1775d70e-cd61-4dd6-983a-64f067486adf",
      "email":"marek.***@salesmanago.pl"
    }
  ]
}
```

```
    }  
  ]  
}
```

gdzie:

modifiedContacts – kontakty zmodyfikowane w podanym przedziale czasu,
id – id kontaktu,
email – email kontaktu

Aktywność kontaktów

Z SALESmanago możemy pobrać informacje o aktywności kontaktów w danym okresie. W tym celu wykonujemy metodę: <http://www.salesmanago.pl/api/contact/recentActivity>
Przykładowa struktura danych zapytania:

```
{  
  "apiKey" : "your-api-key-123",  
  "clientId" : "your-client-id-123",  
  "requestTime" : 1329128188409,  
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",  
  "from":1328050800504,  
  "to":1333231200504,  
  "allVisits":true  
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

from – data początkowa (timestamp, czyli czas jaki upłynął w milisekundach od północy 1 stycznia 1970 UTC)*

to – data początkowa (timestamp, czyli czas jaki upłynął w milisekundach od północy 1 stycznia 1970 UTC)*

allVisits – w przypadku ustawienia na true SALESmanago zwróci w detalach odwiedzin informacje o wszystkich stronach otwartych przez klienta w wybranym okresie*

ipDetails – w przypadku ustawienia na true SALESmanago zwróci w detalach odwiedzin informacje dodatkowe wyszukane dla IP klienta*

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{  
  "success":true,  
  "message":[],  
  "monitoredContacts": 12300,  
  "totalContacts":234000,  
  "recentActivities": {  
    "from":1328050800504,  
    "to":1333231200504,  
    "customers":[{  
      "host": "salesmanago.pl",  
      "time":1330239675000,  
      "duration":22000,  
      "visitSource":"REFERRER",  
      "visitSourceHost":null,  
    }  
  ]  
}
```

```

    "visitSourceKeywords": "localhost",
    "visitSourceDetails": null,
    "visitScore": 120,
    "client": "Benhauer Sp. z o.o. Sp. K. - Konrad Pawlus",
    "email": "konradpawlus@gmail.com",
    "contactId": "1d8cba47-f4b2-4efe-8250-5bdab5346628",
    "ipOrganization": "TP SA",
    "ipDetails": {
      "ip": "123.123.123.123",
      "countryCode": "PL",
      "countryName": "Poland",
      "regionCode": "77",
      "regionName": "Malopolskie",
      "city": "Cracow",
      "postalCode": "",
      "latitude": "50.083300",
      "longitude": "19.916700",
      "isp": "Neostrada Plus",
      "organization": "Neostrada Plus",
    },
    "contactVisits": [{ "time": 2012-02-13 10:38:53,
      "duration": 12345677,
      "visitSource": "www.google.pl",
      "visitSourceKeywords": "SALESmanago",
      "visitHost": null,
      "visitScore": 78,
      "url": null
    }, ... ]},
    "partners": [... jw. ...],
    "prospects": [... jw. ...],
    "anonymous": [... jw. ...],
    "visitSources": [
      {"label": "example.com", "value": 123},
      {"label": "other", "value": 1432},
      ... ],
    "visitSearchTerms": [
      {"label": "tell me google", "value": 123},
      {"label": "what is life?", "value": 1432},
      ... ],
    "visitStats": [
      {
        "date": 1330239675000,
        "partnersVisits": 123
        "prospectsVisits": 234
        "customersVisits": 456
        "otherVisits": 4321
      },
      ... ],
  }
}

```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)
message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd
recentActivity – struktura składająca się z trzech tablic wizyt: klientów

(customers), partnerów (partners), potencjalnych klientów (prospects).

from – data początkowa

to – data końcowa

monitoredContacts – ilość monitorowanych kontaktów

totalContacts – ilość wszystkich kontaktów

customers – lista wizyt klientów

partners – lista wizyt partnerów

prospects – lista wizyt potencjalnych klientów

anonymous – lista wizyt anonimowych

element wizyty klienta w listach:

host – strona która była odwiedzana

time – czas wizyty

duration – czas trwania

visitSource – zasoby wizyty

visitSourceKeywords – słowa kluczowe

visitSourceHost – host

visitScore – liczba punktów

url – URL odwiedzanej strony

client – nazwa klienta

email – email klienta

contactId – unikatowy identyfikator klienta,

ipOrganization – nazwa organizacji pobrana z IP (ISP) /deprecated/

ipDetails – detale zdekodowane z IP klienta

ip – numer IP

countryCode – kod ISO kraju (2-znaki)

countryName – nazwa kraju

regionCode – kod regionu

regionName – nazwa regionu

city – miasto

postalCode – kod pocztowy

latitude – szerokość geograficzna

longitude – długość geograficzna

isp – nazwa ISP

organization – nazwa organizacji

contactVisits – wizyty kontaktu

time – czas wizyty

duration – czas trwania

visitSource – zasoby wizyty

visitSourceKeywords – słowa kluczowe

visitSourceHost – host

visitScore – liczba punktów

url – URL odwiedzanej strony

visitSources – lista źródeł odwiedzin:

label – nazwa źródła

value – ilość odwiedzin

visitSearchTerms – lista wyszukiwanych fraz:
label – wyszukiwana fraza
value – ilość odwiedzin
visitStats – statystyki wizyt z ostatniego tygodnia:
date – czas wizyty
partnersVisits – ilość wizyt partnerów
prospectsVisits – ilość wizyt potencjalnych klientów
customersVisits – ilość wizyt klientów
otherVisits – ilość innych wizyt

Zdarzenia zewnętrzne

W SALESmanago można rejestrować zdarzenia zewnętrzne dla kontaktu, nie koniecznie związane z jego aktywnością online. Przykładowo: zakup produktu w sklepie stacjonarnym, wizytę w lokalu itp.

Dodawanie zdarzenia zewnętrznego

W celu dodania zdarzenia wykonujemy metodę:

<http://www.salesmanago.pl/api/contact/addContactExtEvent>.

Przykładowa struktura zapytania rejestrującego nowe zdarzenie:

```
{
  "clientId": "your-client-id-123",
  "apiKey": "your-api-key-123",
  "requestTime": 1356180568127,
  "sha": "3e4ec39722326150aae60f41e038d1def4450f46",
  "owner": "admin@vondor.pl",
  "email": "konrad@benhauer.com",
  "contactEvent": {
    "date": 1356180568153,
    "description": "Zakup z kartą \"Super Bonus\"",
    "products": "p01, p02",
    "location": "Krupnicza 3, Kraków",
    "value": 1234.43,
    "contactExtEventType": "PURCHASE",
    "detail1": "C.ID: *** ** 234",
    "detail2": "Płatość kartą",
    "detail3": null,
    "externalId": "A-123123123"
  }
}
```

W powyższym zapytaniu należy przesłać następujące wartości:

owner – właściciel kontaktu*,

email – email kontaktu dla którego dodajemy zdarzenie*,

contactEvent – detale zdarzenia:

date – data zdarzenia (timestamp, czyli czas jaki upłynął w milisekundach od północy 1 stycznia 1970 UTC)*,

description – opis zdarzenia,

products – opcjonalna lista produktów oddzielona przecinkami,
location – opcjonalne miejsce zdarzenia np. adres sklepu, placówki,
value – opcjonalna wartość zdarzenia np. wydana kwota,
contactExtEventType – typ zdarzenie, dopuszczalne wartości: PURCHASE, CART, VISIT, PHONE_CALL, OTHER*
detail1, detail2, detail3 – opcjonalne detale zdarzenia,
externalId – opcjonalne ID zdarzenia np. ID z systemu kasowego i.t.p.,

W odpowiedzi otrzymamy ID zdarzenia w systemie SALESmanago:

```
{
  "success":true,
  "message":[],
  "eventId":"7284e317-3bb6-4505-afbe-55b9a101339a"
}
```

gdzie:

eventId – ID dodanego zdarzenia,

Modyfikacja zdarzenia

W celu zmodyfikowania zdarzenia wykonujemy metodę:
<http://www.salesmanago.pl/api/contact/updateContactExtEvent>.
 Przykładowa struktura zapytania modyfikująca zdarzenie:

```
{
  "clientId":"your-client-id-123",
  "apiKey":"your-api-key-123",
  "requestTime":1356180568127,
  "sha":"3e4ec39722326150aae60f41e038d1def4450f46",
  "owner":"admin@vondor.pl",
  "contactEvent":{
    "eventId":"7284e317-3bb6-4505-afbe-55b9a101339a",
    "date":1356180568153,
    "description":"Zakup z kartą \"Super Bonus\"",
    "products":"p01, p02",
    "location":"Krupnicza 3, Kraków",
    "value":1234.43,
    "contactExtEventType":"PURCHASE",
    "detail1":"C.ID: *** ** 234",
    "detail2":"Płatość kartą",
    "detail3":null,
    "externalId":"A-123123123"
  }
}
```

W powyższym zapytaniu należy przesłać następujące wartości:

owner – właściciel kontaktu*,
contactEvent – detale zdarzenia:
eventId – ID zdarzenia (zwrócone przez metodę dodającą)*,

date – data zdarzenia (timestamp, czyli czas jaki upłynął w milisekundach od północy 1 stycznia 1970 UTC) *,
description – opis zdarzenia,
products – opcjonalna lista produktów oddzielona przecinkami,
location – opcjonalne miejsce zdarzenia np. adres sklepu, placówki,
value – opcjonalna wartość zdarzenia np. wydana kwota,
contactExtEventType – typ zdarzenie, dopuszczalne wartości: PURCHASE, CART, VISIT, PHONE_CALL, OTHER
detail1, detail2, detail3 – opcjonalne detale zdarzenia,
externalId – opcjonalne ID zdarzenia np. ID z systemu kasowego i.t.p.,

W odpowiedzi otrzymamy ID zdarzenia w systemie SALESmanago:

```
{
  "success": true,
  "message": [],
  "eventId": "7284e317-3bb6-4505-afbe-55b9a101339a"
}
```

gdzie:

eventId – ID dodanego zdarzenia,

Usuwanie zdarzenia

W celu usunięcia zdarzenia wykonujemy metodę:

<http://www.salesmanago.pl/api/contact/deleteContactExtEvent>.

Przykładowa struktura zapytania usuwającego zdarzenie:

```
{
  "clientId": "your-client-id-123",
  "apiKey": "your-api-key-123",
  "requestTime": 1356180568127,
  "sha": "3e4ec39722326150aae60f41e038d1def4450f46",
  "owner": "admin@vondor.pl",
  "eventId": "7284e317-3bb6-4505-afbe-55b9a101339a"
}
```

W powyższym zapytaniu należy przesłać następujące wartości:

eventId – ID zdarzenia (zwrócone przez metodę dodającą) *,

Rezultat informuje o statusie usunięcia zdarzenia:

```
{
  "success": true,
  "message": [],
  "result": "deleted"
}
```

Przykładowa integracja koszyka

W celu przekazania informacji o koszyku należy użyć operacji

<http://www.salesmanago.pl/api/contact/addContactExtEvent>

oraz

<http://www.salesmanago.pl/api/contact/updateContactExtEvent>

W ramach sesji należy koszykowi przydzielić unikatowe ID zwracane z pierwszego zapytania (*addContactExtEvent*) a następnie przekazywać je jako *eventId* a w kolejnych (gdy osoba dodaje/usuwa produkty z koszyka) należy użyć tegoż ID w metodzie aktualizującej zdarzenie (*updateContactExtEvent*).

Import listy tagów

Tagi importujemy wywołując metodę: <http://www.salesmanago.pl/api/contact/tags>.

Przykładowa struktura danych zapytania:

```
{ "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "showSystemTags" : true,
  "owner" : "admin@vendor.pl",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7"
}
```

Podstawowe elementy jakie należy podać importując listę kontaktów:

showSystemTags – przy ustawieniach na true SALESmanago zwróci również tagi systemowe*,
owner – adres email właściciela kontaktów*.

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{ "tags" : [ { "tag" : "ADmanago",
              "numberOfTagged" : 12
            } ],
  "message" : [ ],
  "success" : true
}
```

gdzie:

tags – importowane tagi
tag – nazwa tagu
numberOfTagged – ilość otagowanych kontaktów

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)
message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Monitorowanie zdarzeń AJAX/JavaScript

SALESmanago pozwala obok samych wizyt rejestrować zdarzenia JavaScript – np. otwarcia tzw. zakładek czy pobrani plików.

Aby zarejestrować zdarzenie zewnętrzne należy w skrypcie JavaScript na naszej stronie wywołać metodę SALESmanago:

```
smEvent(eventName)
```

gdzie:

eventName to nazwa zdarzenia np. „detale-lokata”, „detale-kredyt”.

należy takie zdarzenie podpiąć pod konkretne elementy, które chce się monitorować, np.:

```
<a href="download.pdf" onclick="smEvent('PDF-download');return true;">download</a>
```

Edycja listy tagów kontaktu

Tagi importujemy wywołując metodę:

<http://www.salesmanago.pl/api/contact/managetags>.

Przykładowa struktura danych zapytania:

```
{
  "clientId": "h4jsu6pc5txybj04",
  "apiKey": "-2203694140088941698545922683614585589",
  "requestTime": 1360792407226,
  "sha": "437b8cac9a30283137f611ad820cdbdb937a20f0",
  "email": "konradpawlus@gmail.com",
  "contactId": null,
  "owner": "admin@vondor.pl",
  "tags": ["new_api_tag_a", "new_api_tag_b", "new_api_tag_c"],
  "removeTags": ["old_api_tag_a", "old_api_tag_b", "old_api_tag_c"]
}
```

Podstawowe elementy jakie należy podać importując listę kontaktów:

email/contactId – identyfikator kontaktu (zamiennie e-mail lub SALESmanago contact ID)*,

owner – adres e-mail właściciela kontaktu*,

tags – tablica nowych tagów kontaktu

removeTags – tablica tagów do usunięcia

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "success": true,
  "message": [],
  "contactId": "be691c54-c515-4dc2-8692-325c3faf4cfd"
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)
contactId – ID modyfikowanego kontaktu,
message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Wysyłka e-maili

W celu wysłania maila poprzez API należy skorzystać z metody:
<http://www.salesmanago.pl/api/email/send>

Przykładowa struktura danych zapytania:

```
{
  "clientId": "o7gslwlc8o1e2ry1",
  "apiKey": "-8773203084919279780-2743038323156910252",
  "requestTime": 1391167514795,
  "sha": "184db1df6ec4893a1f50809bf8d1a4fe88cde4dc",
  "user": "admin@finplan.pl",
  "emailId": "029c504e-193a-43f2-84c7-3b7ee3c4438c",
  "contacts": [
    {
      "email": "user1@example.com",
      "contactId": null,
      "properties": [
        {
          "name": "ext_detal_01",
          "value": "value_01"
        },
        {
          "name": "ext_detal_02",
          "value": "value_02"
        }
      ]
    },
    {
      "email": "user2@example.com",
      "contactId": null,

```

```
        "properties": [
          {
            "name": "ext_detal_03",
            "value": "value_03"
          },
          {
            "name": "ext_detal_04",
            "value": "value_04"
          }
        ]
      },
      "date": 1391167515515,
      "subject": "Sample API subject",
      "campaign": "monitor_me_in_ga"
    }
  ]
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

emailId – identyfikator wiadomości z systemu SALESmanago*.

date – data wysyłki*

subject – temat wysyłki (jeżeli nie podany – zostanie użyty domyślny)*

campaign – kampania dla śledzenia Google UTM (jeżeli nie podana – zostanie użyta domyślna)*

contacts – tablica kontaktów do których wysłać email*

email/contactId – opcjonalnie – adres e-mail kontaktu lub jego identyfikator,

properties – definiowane przez użytkownika atrybuty dodatkowe wiadomości e-mail. Zaleca się nie stosować polskich znaków oraz spacji w nazwie, ale jest to dozwolone. W wiadomości e-mail należy użyć konstrukcji `ścst.nazwaParametru$` w celu podstawienia odpowiedniej wartości.

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "success": true,
  "message": ["Emails are scheduled to send."]
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Zadania kontaktów

Dodawania, aktualizację oraz usuwanie zadania kontaktu wykonujemy za pomocą jednej

metody:

<http://www.salesmanago.pl/api/contact/updateTask>

Przykładowa struktura danych zapytania:

```
{
  "apiKey" : "your-api-key-123",
  "clientId" : "your-client-id-123",
  "requestTime" : 1329128188409,
  "sha" : "02bfe70541d3907cf487f26dc2665b184b1221a7",
  "finished" : false,
  "smContactTaskReq" : {
    "id" : "task-id-123",
    "note" : "Zadzwoń do klienta",
    "date" : "2012-02-13 10:38:53",
    "cc" : "konrad-test-123@konri.com",
    "reminder" : "_30_MIN"
  }
}
```

Podstawowe elementy jakie należy podać dodając nowy kontakt w strukturze to:

finished – przypisanie wartości true spowoduje skasowanie zadania i wtedy wymagany jest jedynie dodatkowy parametr id. Przy dodawaniu i aktualizacji zadania należy przypisać wartość false*.

id – id zadania*

note – notatka zadania

date – data wykonania zadania*

cc – lista emaili do których zostanie wysłane przypomnienie (emaili powinny być oddzielone przecinkami)

reminder – przypomnienie o wykonaniu zadania. Określa kiedy przypomnienia ma zostać wysłane*. Dopuszczalne wartości:

- 15_MIN – 15 minut przed,
- 30_MIN – 30 minut przed,
- 1_HOUR – godzinę przed,
- 12_HOUR – 12 godzin przed,
- 1_DAY – 1 dzień przed,
- 1_WEEK – 1 tydzień przed,

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{ "taskId" : "task-Id-123",
  "message" : [ ],
  "success" : true
}
```

gdzie:

taskId – id zadania którego dotyczy akcja.

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane)

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd

Autoryzacja tymczasowa

System SALESmanago umożliwia uzyskanie tymczasowego tokenu wykorzystując nazwę użytkownika i hasło. W tym celu należy wykonać metodę:

<http://www.salesmanago.pl/api/system/authorise>

Przykładowa struktura danych zapytania:

```
{
  "userName" : "admin@vendor.pl",
  "password" : "*****"
}
```

Podstawowe elementy jakie można podać dodając nowy kontakt w strukturze to:

userName – nazwa użytkownika*

password – hasło*

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "token" : "b426c6663d844305b2539e9bc27b75dc",
  "clientId" : "your-client-id-123",
  "message" : [ ],
  "success" : true
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane),

token – to tymczasowy token, który można użyć w celu autoryzacji w zamian za apiSecret,

clientId – identyfikator klienta, wymagany do kolejnych operacji,

message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd,

Rejestracja konta (tylko partnerzy)

System SALESmanago umożliwia rejestrację nowego konta. W tym celu należy wykonać metodę: <http://www.salesmanago.pl/api/system/registeraccount>

Przykładowa struktura danych zapytania:

```
{
  "clientId": "h4jsu6pc5txybj04",
  "apiKey": "-89769101308486819292381884901962874326",
  "requestTime": 1357130990877,
  "sha": "8bc153ea5110a418d7dac3dc4ed11f442e1a08ac",
  "email": "account.72@salesmanago.pl",
  "password": "haslo123",
  "contactPerson": "Konrad Pawlus",
  "invoiceCity": "Kraków",
}
```

```
"invoiceCountry": "PL",
"invoiceStreetAddress": "Krupnicza 3",
"invoiceZipCode": "31-123",
"companyName": "Benhauer Sp z o.o. Sp. K.",
"contactCity": "Kraków",
"contactCountry": "PL",
"contactPhone": "0048600500400",
"contactStreetAddress": "Krupnicza 3",
"contactZipCode": "31-123",
"nip": "123-123-123-123",
"lang": "pl",
"version": "PRO"
}
```

Podstawowe elementy jakie można podać dodając nowy kontakt w strukturze to:

email – adres e-mail nowego konta *,
password – hasło dla nowego konta *,
contactPerson – osoba kontaktowa,
companyName – nazwa firmy,
nip – numer NIP firmy,
invoiceStreetAddress – adres fakturowania: ulica i numer domu,
invoiceCity – adres fakturowania: miasto,
invoiceZipCode – adres fakturowania: kod pocztowy,
invoiceCountry – adres fakturowania: 2 literowy kod państwa (PL - Polska),
contactStreetAddress – adres kontaktowy: ulica i numer domu,
contactCity – adres kontaktowy: miasto,
contactZipCode – adres kontaktowy: kod pocztowy,
contactCountry – adres kontaktowy: 2 literowy kod państwa (PL - Polska),
lang – domyślny język (2 literowy kod języka: pl – j. polski),
version – wersja konta: PRO, LIGHT, SUPER_LIGHT,

Rezultat zapytania przesyłany jest w odpowiedzi jako struktura JSON:

```
{
  "success": true,
  "message": [],
  "clientId": "ujk3j74yfan2jeit",
  "apiSecret": "xivw1g7h1m7mxraambh2drwfb8fth0tn"
}
```

gdzie:

success – wartość logiczna informująca o rezultacie zapytania (udane / nie udane),
clientId – id klienta (używane w komunikacji API oraz kodach monitorujących),
apiSecret – klucz API,
message – tablica dodatkowych informacji pozwalająca zidentyfikować błąd,