

# LiveSpace API



**Aktualizacja:** 15 stycznia 2016

**Wersja:** 0.9



# 1 Wstęp

W dokumencie opisano sposób komunikacji z API LiveSpace, przygotowania parametrów, wykorzystania SDK oraz zaprezentowano przykłady wywołań najpopularniejszych metod.

## 2 Spis treści

1	Wstęp .....	2
2	Spis treści.....	2
3	Komunikacja .....	4
3.1	Żądanie .....	4
3.1.1	Format adresu .....	4
3.1.2	Autoryzacja.....	4
3.1.3	Wykonanie żądania .....	5
3.2	Wynik.....	6
3.2.1	Przykład .....	6
4	SDK.....	7
4.1	Przykład.....	7
5	Identyfikatory.....	8
6	Dostępne metody.....	8
6.1	Zapytanie testowe .....	8
6.2	Dane zalogowanego użytkownika .....	8
7	Kontakty – osoby.....	9
7.1	Pobranie danych .....	9
7.2	Dodanie.....	10
7.3	Edycja.....	12
7.4	Usuwanie .....	13
7.5	Tagi.....	13
7.6	Zadanie .....	13
7.7	Pliki.....	14
7.8	Tablica.....	14
7.8.1	Dodanie wpisu na tablicy .....	14
7.8.2	Pobranie wpisów z tablicy .....	14
7.9	Wyszukiwanie .....	15
7.9.1	Wyszukiwanie wg cech.....	15
7.9.2	Wyszukiwanie wg frazy .....	15
8	Kontakty – firmy .....	16
8.1	Pobranie danych .....	16
8.2	Dodanie.....	17
8.3	Edycja.....	19
8.4	Usuwanie .....	19

8.5	Tagi.....	19
8.6	Zadanie .....	20
8.7	Pliki.....	20
8.8	Tablica.....	20
8.8.1	Dodanie wpisu na tablicy .....	20
8.8.2	Pobranie wpisów z tablicy .....	20
8.9	Wyszukiwanie .....	21
8.9.1	Wyszukiwanie wg cech.....	21
8.9.2	Wyszukiwanie wg frazy .....	21
9	Sprzedaje .....	22
9.1	Pobranie danych .....	22
9.2	Dodanie.....	23
9.3	Edycja.....	25
9.4	Usuwanie .....	25
9.5	Tagi.....	25
9.6	Zadanie .....	26
9.7	Pliki.....	26
9.8	Tablica.....	26
9.8.1	Dodanie wpisu na tablicy .....	26
9.8.2	Pobranie wpisów z tablicy .....	26
9.9	Wyszukiwanie .....	27
9.9.1	Wyszukiwanie wg cech.....	27
9.9.2	Wyszukiwanie wg frazy .....	27
10	Zadania .....	29
10.1	Pobranie danych .....	29
10.2	Dodanie .....	29
10.3	Edycja .....	30
10.4	Usunięcie.....	30
11	Tablica.....	31
11.1	Pobranie danych .....	31
12	Obsługa formularzy .....	32

## 3 Komunikacja

### 3.1 Żądanie

Żądania do API wykonywane są w postaci zapytań HTTP (metoda POST). Nazwa modułu oraz metody jest wyspecyfikowana w adresie, a dane wejściowe metody oraz inne wymagane np. przez mechanizmy autoryzacji są przekazywane w parametrach wywołania.

#### 3.1.1 Format adresu

`https://DOMENA.livespace.pl/api/public/FORMAT_WYJŚCIA/MODUŁ/METODA`

Znaczenie poszczególnych elementów adresu:

- DOMENA – subdomena, nazwa konta w LiveSpace
- FORMAT\_WYJŚCIA – format, w którym zostaną zwrócone dane wyjściowe, dopuszczalne wartości: json, xml, php (wynik metody *serialize()*)
- MODUŁ – nazwa modułu aplikacji, np. *Contact*, *Deal*, *Todo* itp.
- METODA – nazwa metody w danym module np. *addContact*

Przykład:

`https://SUBDOMENA.livespace.pl/api/public/json/Contact/addContact`

#### 3.1.2 Autoryzacja

Obecnie wspierana jest wyłącznie autoryzacja przy pomocy klucza API powiązanego z użytkownikiem aplikacji oraz unikalnych tokenów generowanych **dla każdego żądania**.

Klucze API\_KEY (przesyłany z każdym żądaniem) oraz API\_SECRET (nieprzesyłany w komunikacji, służący do generowania sumy kontrolnej) są odczytywane w ustawieniach LiveSpace (zakładka API) i są stałe dla wszystkich wywołań. Powiązane są one z kontami użytkowników, po wpisaniu danych wybranego użytkownika wszystkie operacje są wykonywane w jego imieniu i z jego uprawnieniami.

##### 3.1.2.1 Przykład

Pobranie tokenu, który będzie użyty w kolejnym wywołaniu.

**URL:**

`https://SUBDOMENA.livespace.pl/api/public/json/_Api/auth_call  
/_api_method/getToken`

**Parametry POST:**

`_api_auth: key  
_api_key: API_KEY`

**Wynik:**

```

data: [
  token: TOKEN
  session_id: SESSION_ID
]
error: null
result: 200
status: true

```

**3.1.3 Wykonanie żądania**

Wśród parametrów przekazujemy sumę kontrolną SHA obliczaną jako wynik funkcji *sha1* na ciągu znaków powstałym z konkatenacji: API\_KEY, TOKENu i API\_SECRET.

Pozostałe parametry zależą od wymagań poszczególnych metod.

Parametry żądania przekazywane są w postaci wielu parametrów POST lub w formacie JSON jako wartość jednego parametru *data*.

**3.1.3.1 Przykład****URL:**

<https://SUBDOMENA.livespace.pl/api/public/json/Contact/addContact>

**Parametry POST:**

```

data:[
  _api_auth: key
  _api_key: API_KEY
  _api_sha: SHA
  _api_session: SESSION_ID
  contact: [
    firstname: Jan
    lastname: Kowalski
  ]
]

```

**Wartość parametru data jest przekazywana w formacie JSON:**

```

data:
{"_api_auth":"key","_api_key":"API_KEY","_api_sha":"SHA","_api_session":
"SESSION_ID","contact":{"firstname":"Jan","lastname":"Kowalski"}}

```

## 3.2 Wynik

Wynik przekazywany we wskazanym formacie (sugerowane jest wykorzystanie formatu JSON) jako tablica czteroelementowa z polami:

- *status* – true/false – informacja czy zapytanie zostało wykonane poprawnie
- *result* – kod (znaczenie zgodnie z listą poniżej)
- *data* – dane zwrócone przez wywołaną metodę, najczęściej w postaci tablicy
- *error* – opis błędów

Wartości *data* i *error* są specyficzne dla każdej metody.

Znaczenie kodów *result* zostało przedstawione poniżej. Poszczególne metody mogą stosować własne kody, wykraczające poza poniższe zestawienie.

- Poprawne
  - 200 – ok
- Błędy metod
  - 400 – błąd ogólny metody
  - 420 – błąd walidacji
- Błędy obsługi API
  - 500 – błąd ogólny api
  - 514 – niepoprawny moduł
  - 515 – niepoprawna metoda
  - 516 – niepoprawny format wyjścia
  - 520 – błąd w komunikacji z bazą danych
- 530 – użytkownik niezalogowany
- 540 – brak uprawnień
- 550 – błąd obsługi parametrów
- Błędy metod autoryzacji
  - 560 – niepoprawna metoda
  - 561 – niepoprawne parametry
  - 562 – niepoprawny klucz
  - 563 – brak autoryzacji
  - 564 – błąd ogólny

### 3.2.1 Przykład

Przykładowy wynik żądania dla metody *Contact/addContact*:

```
data: [
  id: 123
  firstname: Jan
  lastname: Kowalski
]
error: null
result: 200
status: true
```

## 4 SDK

W celu usprawnienia wykorzystania API w języku PHP zostało przygotowane SDK, które ułatwia komunikację z LiveSpace, wywołanie metod oraz interpretację wyników.

Wymagania biblioteki to PHP w wersji 5.2 lub wyższej oraz zainstalowane rozszerzenia json oraz curl.

### 4.1 Przykład

Dodanie kontaktu z wykorzystaniem SDK.

Ustawienie parametrów API:

```
$ls = new LiveSpace(array(
    'api_url' => 'https://SUBDOMENA.livespace.pl',
    'api_key' => API_KEY,
    'api_secret' => API_SECRET
));
```

Przygotowanie parametrów metody:

```
$contactData = array(
    'contact' => array(
        'firstname' => 'Jan',
        'lastname' => 'Kowalski'
    )
);
```

Wykonanie żądania:

```
$result = $ls->call('Contact/addContact', $contactData);
```

Interpretacja wyniku:

```
if ($result->getStatus()) {
    echo 'Zapytanie wykonane poprawnie' . "\r\n";
    var_export($result->getResponseData());
} else {
    echo 'Wystąpił błąd #' . $result->getResult() . ":\r\n";
    print_r($result->getError());
}
```

## 5 Identyfikatory

W komunikacji z API używane są inne identyfikatory obiektów od tych widocznych w aplikacji np. w adresach URL.

Istnieje możliwość wyświetlenia identyfikatorów API na profilach kontaktów i sprzedaży. Aby to zrobić należy w ustawieniach w zakładce API zaznaczyć opcję 'Pokaż identyfikatory API na profilach obiektów'.

Identyfikatory pól dodatkowych, etapów sprzedaży itp. można odczytać we wspomnianej zakładce API w ustawieniach.

## 6 Dostępne metody

Przykłady dla czytelności powstały w oparciu o SDK. Jeżeli SDK nie będzie używane powinien zostać zachowany schemat komunikacji opisany w rozdziale 3. a kluczowe dla wywołania nazwy metod i parametry należy przekazać zgodnie ze wzorem zawartym w przykładach.

Założmy, że zdefiniowano obiekt `$ls` jak poniżej:

```
require_once 'livespace.php';

$ls = new LiveSpace(array(
    'api_url' => 'https://SUBDOMENA.livespace.pl',
    'api_key' => API_KEY,
    'api_secret' => API_SECRET
));
```

### 6.1 Zapytanie testowe

Zapytanie zwracające w odpowiedzi przekazane parametry.

```
$result = $ls->call('Default/ping', array(
    'param1' => 'value1',
    'param2' => 'value2'
));
```

### 6.2 Dane zalogowanego użytkownika

```
$result = $ls->call('Default/User_getInfo');
```



## 7 Kontakty – osoby

### 7.1 Pobranie danych

Pojedynczy kontakt:

```
$result = $ls->call('Contact/get', array(
    'type' => 'contact',
    'id' => '9497f068-8542-a93d-5721-1a636cebba4a'
));
```

Pobranie podstawowych informacji o wielu osobach:

```
$result = $ls->call('Contact/getAllSimple', array(
    'type' => 'contact',
    'limit' => '1000', // Liczba kontaktów (domyślnie wszystkie)
    'offset' => '0' // Offset (liczba pomijanych początkowych kontaktów,
                    // domyślnie 0)
));
```

Pobranie szczegółowych informacji o wielu osobach (patrz też 6.3.11 wyszukiwanie wg frazy):

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'contact',
    // dla parametrów podajemy jedną wartość lub kilka oddzielonych przecinkami
    // parametry są opcjonalne ale należy podać przynajmniej jeden warunek
    // imiona
    'firstnames' => 'jan,karol'
    // nazwiska
    'lastnames' => 'kowalski'
    // ID firmy
    'companies' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80',
    // adres email
    'emails' => 'jan.kowalski@firma.pl',
    // telefon
    'phones' => '666-55-44,333444555',
    // dla parametrów związanych z datą można podać pojedynczą wartość
    // (traktowana jest jak wartość 'od') lub tablicę z wartościami 'od' i 'do'
    // data utworzenia
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // data modyfikacji
    'modified' => '2015-01-15',
    // data ostatniej aktywności
    'last_active' => '2015-01-20',
    // loginy właścicieli rozdzielone przecinkami
    'owner_login' => 'jan.kowalski@firma.com',
    // opcjonalnie sposób porównywania imion i nazwisk, 'like' - wzorzec,
    // 'equal' - dokładne dopasowanie, domyślnie 'like'
    'condition' => 'like',
    // Tagi
    'tags' => 'tag1,tag2',
    // Sposób wyszukiwania po tagach ('or' lub 'and', domyślnie 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Liczba kontaktów (domyślnie wszystkie)
    'offset' => '0' // Liczba pomijanych początkowych (domyślnie 0)
));
```

## 7.2 Dodanie

Bardzo proste:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'name' => 'Jan Kowalski' // imię i nazwisko (wymagane)
    )
));
```

Proste:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'Jan', // imię (wymagane)
        'lastname' => 'Kowalski', // nazwisko (wymagane)
        'emails' => array( // adresy email
            'jan@kowalski.pl',
            'jan.kowalski@firma.pl'
        ),
        'phones' => array( // telefony
            '1234567890',
            '0987654321'
        ),
        'company' => array( // firma
            'name' => 'Firma S.A.'
        )
    )
));
```

Więcej danych:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'Jan', // imię (wymagane)
        'lastname' => 'Kowalski', // nazwisko (wymagane)
        'note' => 'notatka', // notatka
        'position' => 'prezes', // stanowisko
        'contact_source' => 'formularz na stronie', // źródło pozyskania
        'created' => '2013-10-10 11:22:33', // data dodania (opcjonalna)
        'www' => 'strona-prywatna.pl', // strona WWW
        'emails' => array( // adresy email
            0 => array(
                'email' => 'jan@kowalski.pl' // adres email
            ),
            1 => array(
                'email' => 'jan.kowalski@firma.pl', // adres email
                'is_default' => 1
            )
        ),
        'phones' => array( // telefony
            0 => array(
                'phone_no' => '1234567890', // numer telefonu
                'type' => 1, // typ telefonu: 1 - telefon,
                // 2 - tel. komórkowy, 3 - fax
            ),
        ),
    )
));
```

```

1 => array(
    'phone_no' => '0987654321', // numer telefonu
    'type' => 2, // typ telefonu: 1 - telefon,
                                     2 - tel. komórkowy, 3 - fax
    'is_default' => 1
)
),
'addresses' => array( // adresy
    0 => array(
        'street' => 'Ulica 1/2', // ulica
        'city' => 'Warszawa', // miasto
        'postcode' => '00-999', // kod pocztowy
        'province' => 10, // numer województwa, kolejność alfabetyczna:
        // zna: 1 - dolnośląskie, 2 - kujawsko-pomorskie, ... 16 - zachodniopomorskie
        'country' => 'Polska' // nazwa kraju
    )
),
'company' => array( // firma -
    // składnia jak dla dodawania firm lub id
    'name' => 'Firma S.A.', // nazwa, wymagana
    'nip' => '111-222-55-00', // nip
    'regon' => '141981336' // regon
),
'groups' => array( // grupy, do których zostanie dodany kontakt
    '87de5630-d889-bdb1-bd09-d61335d27ff8',
    '1e5d3fa6-da59-da59-4657-4998c2184e6d'
),
'dataset' => array( // wartości dla pól dodatkowych, identyfikatory
    // pól do odczytania w ustawieniach LiveSpace, inne dla każdej instancji
    '87de5630-d889-bdb1-bd09-d61335d27ff8' => '56e966a4-f265-39ac-
7c6e-ccf5a8caebb1', // dla pól typu select podaje id odpowiedzi
    '1e5d3fa6-da59-da59-4657-4998c2184e6d' => 'value_2'
    // dla pól tekstowych podajemy pełną odpowiedź
),
'owner_id' => 'ff3a0bdb-7348-50c8-071a-
ced692fdb898', // id użytkownika, właściciela kontaktu -
    // opcjonalne, domyślnie ustawiany jest użytkownik w imieniu, którego następu
    // je połączenie z api
'notification' => 'Treść' // Powiadomienie dla właściciela kontaktu
    // (opcjonalne)
)
));

```

Dodanie wielu osób jednocześnie:

```

$result = $ls->call('Contact/addContacts', array(
    'contacts' => array(
        0 => array(
            'firstname' => 'Janusz', // imię (wymagane)
            'lastname' => 'Nowak' // nazwisko (wymagane)
        ),
        1 => array(
            'firstname' => 'Karol', // imię (wymagane)
            'lastname' => 'Nowakowski' // nazwisko (wymagane)
        )
    )
));

```

Standardowo podczas dodawania kontaktu następuje sprawdzenie czy kontakt o takim imieniu i nazwisku już istnieje, aby uniknąć dublowania danych. W celu wymuszenia dodania nowego kontaktu należy użyć przełącznika `__check_if_exists`:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'Jan', // imię (wymagane)
        'lastname' => 'Kowalski', // nazwisko (wymagane)
        '__check_if_exists' => false, // wymuś dodanie
    )
));
```

Pod dodaniem kontaktu na tablicy pojawia się informacja o tej aktywności, aby tego uniknąć należy wyłączyć komunikat przy pomocy przełącznika `_wall`:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'Jan', // imię (wymagane)
        'lastname' => 'Kowalski', // nazwisko (wymagane)

        '_wall' => false, // nie pokazuj aktywności na tablicy
    )
));
```

## 7.3 Edycja

Składnia identyczna jak dla dodawania, wymagany parametr *id*.

Zmiana nazwiska i dodanie adresu email:

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'lastname' => 'Nowak', // nowe nazwisko
        'emails' => array( // nowy, kolejny adres email
            'jan.nowak@firma.pl'
        )
    )
));
```

Przypisanie do firmy:

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'company' => array( // id istniejącej firmy lub nazwa
            'id' => '1e5d3fa6-da59-da59-4657-4998c2184e6d'
        )
    )
));
```

Usunięcie z firmy:

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'company' => '__no_company'
    )
));
```

Dodanie nowych numerów telefonu i usunięcie starych:

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'phones' => array( // telefony
            0 => array(
                'phone_no' => '1234567890', // numer telefonu
                'type' => 1, // typ telefonu: 1 - telefon, 2 -
tel . komórkowy, 3 - fax
            ),
            1 => array(
                'phone_no' => '0987654321', // numer telefonu
                'type' => 2, // typ telefonu: 1 - telefon, 2 -
tel. komórkowy, 3 - fax
                'is_default' => 1
            ),
            '_delete_old' => false // czy usunąć stare
        ),
    ),
));
```

## 7.4 Usuwanie

```
$result = $ls->call('Contact/deleteContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
    )
));
```

## 7.5 Tagi

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'tag_add' => 'nowy tag1,nowy tag2', // tagi do dodania
        'tag_remove' => 'tag3,tag4', // tagi do usunięcia
    )
));
```

## 7.6 Zadanie

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'todo' => array(
            'title' => 'Nowe zadanie', // tytuł (wymagany)
            'date' => '2013-10-10', // data (opcjonalna)
                format rrrr-mm-dd lub rrrr-mm-dd gg:mm
            'description' => 'opis', // opis (opcjonalny)
            'user_id' => '87de5630-d889-bdb1-bd09-d61335d27ff8' // id
użytkownika, do którego zostanie przypisane zadanie (opcjonalne)
        )
    )
));
```

## 7.7 Pliki

Dodanie na tablicę kontaktu pliku dostępnego pod wskazanym adresem URL.

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'file' => array(
            'url' => 'http://domain/file.ext', // url pliku (wymagany)
            'userpwd' => 'login:password', // login i hasło do autoryzacji
                                     (opcjonalne)
            'description' => 'opis', // opis (opcjonalny)
        )
    )
));
```

## 7.8 Tablica

### 7.8.1 Dodanie wpisu na tablicy

```
$result = $ls->call('Contact/addContactNote', array(
    'contact' => array(
        'id' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80', // Id osoby
        'note' => 'treść wpisu', // treść wpisu
        'tags' => 'tag1,tag2' // opcjonalne tagi
    )
));
```

### 7.8.2 Pobranie wpisów z tablicy

```
$result = $ls->call('Contact/getWall', array(
    'type' => 'contact',
    'id' => '9497f068-8542-a93d-5721-1a636cebba4a'
));
```

## 7.9 Wyszukiwanie

### 7.9.1 Wyszukiwanie wg cech

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'contact',
    // dla parametrów podajemy jedną wartość lub kilka oddzielonych przecinkami
    // parametry są opcjonalne ale należy podać przynajmniej jeden warunek
    // imiona
    'firstnames' => 'jan, karol'
    // nazwiska
    'lastnames' => 'kowalski'
    // ID firmy
    'companies' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80',
    // adres email
    'emails' => 'jan.kowalski@firma.pl',
    // telefon
    'phones' => '666-55-44,333444555',
    // dla parametrów związanych z datą można podać pojedynczą wartość
    // (traktowana jest jak wartość 'od') lub tablicę z wartościami 'od' i 'do'
    // data utworzenia
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // data modyfikacji
    'modified' => '2015-01-15',
    // data ostatniej aktywności
    'last_active' => '2015-01-20',
    // loginy właścicieli rozdzielone przecinkami
    'owner_login' => 'jan.kowalski@firma.com',
    // opcjonalnie sposób porównywania imion i nazwisk, 'like' - wzorzec,
    // 'equal' - dokładne dopasowanie, domyślnie 'like'
    'condition' => 'like',
    // Tagi
    'tags' => 'tag1,tag2',
    // Sposób wyszukiwania po tagach ('or' lub 'and', domyślnie 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Liczba kontaktów (domyślnie wszystkie)
    'offset' => '0' // Liczba pomijanych początkowych (domyślnie 0)
));
```

### 7.9.2 Wyszukiwanie wg frazy

Podstawowe:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'contact', // typ obiektu
    'q' => 'jan' // szukana fraza
));
```

Zaawansowane:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'contact', // typ obiektu
    'q' => 'jan kowalski', // szukana fraza
    'type' => 'like', // sposób wyszukiwania like lub equal
    'condition' => 'AND', // AND lub OR, spójnik dla warunków na kolejnych
    // słowach frazy, decyduje czy w kontekście danego obiektu muszą wystąpić wsz
    // ystkie słowa z frazy (AND) czy tylko jedno z nich (OR)
    'limit' => 10, // liczba wyników
    'offset' => 0 // offset, przydatny do stronicowania
));
```

## 8 Kontakty - firmy

### 8.1 Pobranie danych

Pojedyncza firma:

```
$result = $ls->call('Contact/get', array(
    'type' => 'company',
    'id' => '9497f068-8542-a93d-5721-1a636cebba4a'
));
```

Pobranie podstawowych informacji o wielu firmach:

```
$result = $ls->call('Contact/getAllSimple', array(
    'type' => 'company',
    'limit' => '1000', // Liczba kontaktów (domyślnie wszystkie)
    'offset' => '0' // Offset (liczba pomijanych początkowych kontaktów,
                    // domyślnie 0)
));
```

Pobranie szczegółowych informacji o wielu firmach (patrz też 8.9.2 wyszukiwanie wg frazy):

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'company',
    // dla parametrów podajemy jedną wartość lub kilka oddzielonych przecinkami
    // parametry są opcjonalne ale należy podać przynajmniej jeden warunek
    // nazwa firmy
    'names' => 'Bank XYZ,Firma ABC',
    // NIP (w dowolnym formacie 123-456-78-90, 1234567890, PL123-45-67-890)
    'nip' => '123-456-78-90',
    // REGON
    'regon' => '1234567890123',
    // adres email
    'emails' => 'kontakt@firma.pl',
    // telefon
    'phones' => '666-55-44,333444555',
    // dla parametrów związanych z datą można podać pojedynczą wartość
    // (traktowana jest jak wartość 'od') lub tablicę z wartościami 'od' i 'do'
    // data utworzenia
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // data modyfikacji
    'modified' => '2015-01-15',
    // data ostatniej aktywności
    'last_active' => '2015-01-20',
    // loginy właścicieli rozdzielone przecinkami
    'owner_login' => 'jan.kowalski@firma.com',
    // opcjonalnie sposób porównywania nazwy, nipu i regonu,
    // 'like' - wzorzec, 'equal' - dokładne dopasownie, domyślnie 'like'
    'cond' => 'like',
    // Tagi
    'tags' => 'tag1,tag2',
    // Sposób wyszukiwania po tagach ('or' lub 'and', domyślnie 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Liczba kontaktów (domyślnie wszystkie)
    'offset' => '0' // Liczba pomijanych początkowych (domyślnie 0)
));
```



## 8.2 Dodanie

Bardzo proste:

```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Firma S.A.' // nazwa firmy (wymagana)
    )
));
```

Proste:

```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Firma S.A.', // nazwa (wymagana)
        'emails' => array( // adresy email
            'biuro@firma.pl'
        ),
        'phones' => array( // telefony
            '1234567890',
            '0987654321'
        )
    )
));
```

Więcej danych:

```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Firma S.A.', // nazwa (wymagana)
        'nip' => '111-222-55-00', // nip
        'regon' => '141981336', // regon
        'note' => 'notatka', // notatka
        'created' => '2013-10-10 11:22:33', // data dodania (opcjonalna)
        'www' => 'strona-prywatna.pl', // strona WWW
        'emails' => array( // adresy email
            0 => array(
                'email' => 'jan@kowalski.pl' // adres email
            ),
            1 => array(
                'email' => 'jan.kowalski@firma.pl', // adres email
                'is_default' => 1
            )
        ),
        'phones' => array( // telefony
            0 => array(
                'phone_no' => '1234567890', // numer telefonu
                'type' => 1 // typ telefonu: 1 - telefon,
                                     2 - tel. komórkowy, 3 - fax
            ),
            1 => array(
                'phone_no' => '0987654321', // numer telefonu
                'type' => 2, // typ telefonu
                'is_default' => 1
            )
        )
    )
));
```

```

'addresses' => array( // adresy
    0 => array(
        'street' => 'Ulica 1/2', // ulica
        'city' => 'Warszawa', // miasto
        'postcode' => '00-999', // kod pocztowy
        'province' => 10, // numer województwa, kolejność alfabetyczna: 1 - dolnośląskie, 2 - kujawsko-pomorskie, ... 16 - zachodniopomorskie
        'country' => 'Polska' // nazwa kraju
    )
),
'groups' => array( // grupy, do których zostanie dodany kontakt
    '87de5630-d889-bdb1-bd09-d61335d27ff8',
    '1e5d3fa6-da59-da59-4657-4998c2184e6d '
),
'dataset' => array( // wartości dla pól dodatkowych, identyfikatory pól do odczytania w ustawieniach LiveSpace, inne dla każdej instancji
    '87de5630-d889-bdb1-bd09-d61335d27ff8' => '56e966a4-f265-39ac-7c6e-ccf5a8caebb1', // dla pól typu select podaje id odpowiedzi
    '1e5d3fa6-da59-da59-4657-4998c2184e6d' => 'value_2' // dla pól tekstowych podajemy pełną odpowiedź
),
'owner_id' => 'ff3a0bdb-7348-50c8-071a-ced692fdb898', // id użytkownika, właściciela kontaktu - opcjonalne, domyślnie ustawiany jest użytkownik w imieniu, którego następuje połączenie z api
'notification' => 'Treść' // Powiadomienie dla właściciela kontaktu (opcjonalne)
)
));

```

Dodanie wielu firm jednocześnie:

```

$result = $ls->call('Contact/addCompanies', array(
    'companies' => array(
        0 => array(
            'name' => 'Firma S.A.' // nazwa (wymagana)
        ),
        1 => array(
            'name' => 'Bank XYZ' // nazwa (wymagana)
        )
    )
));

```

Standardowo podczas dodawania kontaktu następuje sprawdzenie czy kontakt o takim imieniu i nazwisku już istnieje, aby uniknąć dublowania danych. W celu wymuszenia dodania nowego kontaktu należy użyć przełącznika `__check_if_exists`:

```

$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Firma S.A.', // nazwa (wymagana)
        '__check_if_exists' => false, // wymuś dodanie
    )
));

```

Pod dodaniem kontaktu na tablicy pojawia się informacja o tej aktywności, aby tego uniknąć należy wyłączyć komunikat przy pomocy przełącznika `_wall`:

```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Firma S.A.', // nazwa (wymagana)
        '_wall' => false, // nie pokazuj aktywności na tablicy
    )
));
```

### 8.3 Edycja

Składnia identyczna jak dla dodawania, wymagany parametr *id*.

Zmiana nazwy i dodanie adresu email:

```
$result = $ls->call('Contact/editCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'name' => 'Firma sp. z o.o.', // nowa nazwa
        'emails' => array( // nowy, kolejny adres email
            'marketing@firma.pl'
        )
    )
));
```

Dodanie nowych numerów telefonu i usunięcie starych:

```
$result = $ls->call('Contact/editCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'phones' => array( // telefony
            0 => array(
                'phone_no' => '1234567890', // numer telefonu
                'type' => 1, // typ telefonu: 1 - telefon, 2 -
                tel . komórkowy, 3 - fax
            ),
            1 => array(
                'phone_no' => '0987654321', // numer telefonu
            ),
            '_delete_old' => false // czy usunąć stare
        ),
    )
));
```

### 8.4 Usuwanie

```
$result = $ls->call('Contact/deleteCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
    )
));
```

### 8.5 Tagi

```
$result = $ls->call('Contact/editCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'tag_add' => 'nowy tag1,nowy tag2', // tagi do dodania
        'tag_remove' => 'tag3,tag4', // tagi do usunięcia
    )
));
```

## 8.6 Zadanie

```
$result = $ls->call('Contact/editCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'todo' => array(
            'title' => 'Nowe zadanie', // tytuł (wymagany)
            'description' => 'opis', // opis (opcjonalny)
            'date' => '2013-10-10', // data (opcjonalna)
                                format rrrr-mm-dd lub rrrr-mm-dd gg:mm
            'user_id' => '87de5630-d889-bdb1-bd09-d61335d27ff8' // id
                                użytkownika, do którego zostanie przypisane zadanie (opcjonalne)
        )
    )
));
```

## 8.7 Pliki

Dodanie na tablicę kontaktu pliku dostępnego pod wskazanym adresem URL.

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'file' => array(
            'url' => 'http://domain/file.ext', // url pliku (wymagany)
            'userpwd' => 'login:password', // login i hasło do autoryzacji
                                (opcjonalne)
            'description' => 'opis', // opis (opcjonalny)
        )
    )
));
```

## 8.8 Tablica

### 8.8.1 Dodanie wpisu na tablicy

```
$result = $ls->call('Contact/addCompanyNote', array(
    'company' => array(
        'id' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80', // Id firmy
        'note' => 'treść wpisu', // treść wpisu
        'tags' => 'tag1,tag2' // opcjonalne tagi
    )
));
```

### 8.8.2 Pobranie wpisów z tablicy

```
$result = $ls->call('Contact/getWall', array(
    'type' => 'company',
    'id' => '9497f068-8542-a93d-5721-1a636cebba4a'
));
```

## 8.9 Wyszukiwanie

### 8.9.1 Wyszukiwanie wg cech

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'company',
    // dla parametrów podajemy jedną wartość lub kilka oddzielonych przecinkami
    // parametry są opcjonalne ale należy podać przynajmniej jeden warunek
    // nazwa firmy
    'names' => 'Bank XYZ,Firma ABC',
    // NIP (w dowolnym formacie 123-456-78-90, 1234567890, PL123-45-67-890)
    'nip' => '123-456-78-90',
    // REGON
    'regon' => '1234567890123',
    // adres email
    'emails' => 'kontakt@firma.pl',
    // telefon
    'phones' => '666-55-44,333444555',
    // dla parametrów związanych z datą można podać pojedynczą wartość
    // (traktowana jest jak wartość 'od') lub tablicę z wartościami 'od' i 'do'
    // data utworzenia
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // data modyfikacji
    'modified' => '2015-01-15',
    // data ostatniej aktywności
    'last_active' => '2015-01-20',
    // loginy właścicieli rozdzielone przecinkami
    'owner_login' => 'jan.kowalski@firma.com',
    // opcjonalnie sposób porównywania nazwy, nipu i regonu,
    // 'like' - wzorzec, 'equal' - dokładne dopasowanie, domyślnie 'like'
    'cond' => 'like',
    // Tagi
    'tags' => 'tag1,tag2',
    // Sposób wyszukiwania po tagach ('or' lub 'and', domyślnie 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Liczba kontaktów (domyślnie wszystkie)
    'offset' => '0' // Liczba pomijanych początkowych (domyślnie 0)
));
```

### 8.9.2 Wyszukiwanie wg frazy

Podstawowe:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'company', // typ obiektu
    'q' => 'bank' // szukana fraza
));
```

Zaawansowane:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'company', // typ obiektu
    'q' => 'bank prywatny', // szukana fraza
    'type' => 'like', // sposób wyszukiwania like lub equal
    'condition' => 'AND', // AND lub OR, spójnik dla warunków na kolejnych
    // słowach frazy, decyduje czy w kontekście danego obiektu muszą wystąpić wszy
    // stkie słowa z frazy (AND) czy tylko jedno z nich (OR)
    'limit' => 10, // liczba wyników
    'offset' => 0 // offset, przydatny do stronicowania
));
```

## 9 Sprzedaże

### 9.1 Pobranie danych

Jedna sprzedaż:

```
$result = $ls->call('Deal/get', array(
    'id' => '9497f068-8542-a93d-5721-1a636cebba4a' // id sprzedaży
));
```

Wiele sprzedaży (patrz też 6.5.11 wyszukiwanie wg frazy):

```
$result = $ls->call('Deal/getAll', array(
// dla parametrów podajemy jedną wartość lub kilka oddzielonych przecinkami
// parametry są opcjonalne ale należy podać przynajmniej jeden warunek
    // nazwa sprzedaży
    'names' => 'Ubezpieczenia',
    // status sprzedaży - open, won, lost, outdated
    'status' => 'outdated,open',
    // id procesów sprzedaży
    'processes' => 'c93ce2dd-aa21-1172-34b0-430af636c674',
    // id etapów i podetapów sprzedaży
    'main_stages' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244',
    'stages' => '430af6dd-aa21-1172-34b0-c674f636c674',
    // id firm i osób
    'companies' => '3fd46cc0-9de5-4269-4b04-85147f7cb42f',
    'contacts' => '5c480d66-7521-3fbf-f372-bdb41a393c58',
    // dla parametrów związanych z datą można podać pojedynczą wartość
    (traktowana jest jak wartość 'od') lub tablicę z wartościami 'od' i 'do'
    // data utworzenia
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // data modyfikacji
    'modified' => '2015-01-15',
    // data zmiany statusu
    'status_change_date' => '2015-01-15',
    // data finalizacji
    'date_end' => '2015-01-15',
    // data ostatniej aktywności
    'last_active' => '2015-01-20',
    // loginy właścicieli rozdzielone przecinkami
    'owner_login' => 'jan.kowalski@firma.com',
    // opcjonalnie sposób porównywania nazwy, 'like' - wzorzec,
    'equal' - dokładne dopasowanie, domyślnie 'like'
    'cond' => 'like',
    // Tagi
    'tags' => 'tag1,tag2',
    // Sposób wyszukiwania po tagach ('or' lub 'and', domyślnie 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Liczba sprzedaży (domyślnie wszystkie)
    'offset' => '0' // Liczba pomijanych pocz. Sprzedaży (domyślnie 0)
));
```

## 9.2 Dodanie

Bardzo proste:

```
$result = $ls->call('Deal/addDeal', array(
    'deal' => array(
        'name' => 'Sprzedaż usługi', // nazwa (wymagana)
        'company' => array('name' => 'Firma') // składnia jak dla firmy)
    )
));
```

Proste:

```
$result = $ls->call('Deal/addDeal', array(
    'deal' => array(
        'name' => 'Sprzedaż usługi', // nazwa (wymagana)
        'date_end' => '2013-12-10', // data finalizacji
        'process_id' => '4c9d3e56-9aa7-ff1e-4597-88e70a0213bf', //proces
                                                                    (opcjonalny)
        'company' => array('name' => 'Firma') // składnia jak dla firmy)
        'contact' => array('name' => 'Jan Nowak') // składnia jak dla osoby)
    )
));
```

Więcej danych:

```
$result = $ls->call('Deal/addDeal', array(
    'deal' => array(
        'name' => 'Sprzedaż usługi', // nazwa (wymagana)
        'date_end' => '2013-12-10', // data finalizacji
        'process_id' => '4c9d3e56-9aa7-ff1e-4597-88e70a0213bf', // proces
        (opcjonalny)
        'company' => array( // firma (składnia jak dla dod./edycji firmy)
            'name' => 'Firma'
        ),
        'contact' => array( // osoba (składnia jak dla dod./edycji osoby)
            'name' => 'Jan Kowalski'
        ),
        'note' => 'notatka', // notatka
        'created' => '2013-10-10 11:22:33', // data dodania (opcjonalna)
        'dataset' => array( // wartości dla pól dodatkowych, identyfikatory
            // pól do odczytania w ustawieniach LiveSpace, inne dla każdej instancji
            '4998c21d-53de-da59-4657-1e5d3fa684e6' => '8b977d9e-48ad-dc34-
f780-8bc088e6076b', // dla pól typu select podaje id odpowiedzi
            '7859680b-6091-dc05-bbd6-d5fe260319fe' => 'value_2'
            // dla pól tekstowych podajemy pełną odpowiedź
        ),
        'stages' => array(
            'b7193e52-d695-8f6d-5da7-7fb9e792273c', // ID etapu do wł.
            '447c38f5-7384-c41e-047b-6aacecac0919' => 0, // ID etapu do wył.
            '6dd5e857-4a8e-d98e-8c00-64480ebb792a' => 1 // ID etapu do wł.
        ),
        'budget' => array(
            array(
                'category_id' => '6dd5e857-4a8e-d98e-8c00-64480ebb792a',
                    // ID lub nazwa kategorii (wymagane)
                'value' => '123000,00', // Wartość (wymagana)
                'date' => '2013-11-01', // Data (opcjonalna)
                'is_forecast' => true, // Prognoza czy wpływ (opcjonalny)
            )
        )
    )
));
```

```

        'is_profit' => true, // Przychód czy koszt (opcjonalny)
        'description' => 'Prognoza', // Opis (opcjonalny)
        'discount_percent' => '10', // Rabat w % (opcjonalny)
        'cycle_period' => '1 year', // Cykliczność (czas - day,
            week, month, quarter, year) (opcjonalny)
        'cycle_repeat' => '2 month' // Cykliczność (częstotliwość
            czas - day, week, month, quarter, year) (opcjonalny)
    ),
    array(
        'category_name' => 'Kategoria 1', // Nazwa lub ID kategorii
            (wymagane)
        'value' => '12345.99', // Wartość (wymagana)
        'date' => '2013-12-01', // Data (opcjonalna)
    ),
    '_delete_old' => false, // Czy usunąć poprzednie wartości (opc.)
    '_delete_old_details_forecast' => 1,
        // Czy usuwać tylko prognozę (1) / realizację (0) (opc.)
    '_delete_old_details_profit' => 1,
        // Czy usuwać tylko przychód (1) / koszt (0) (opc.)
    ),
    'is_public' => false // Czy sprzedaż jest publiczna (domyślnie nie)
    'owner_id' => 'ff3a0bdb-7348-50c8-071a-ced692fdb898',
        // id użytkownika, właściciela sprzedaży -
        opcjonalne, domyślnie ustawiany jest użytkownik w imieniu, którego następu
        je połączenie z API
    'notification' => 'Treść' // Powiadomienie dla właściciela (opc.)
    )
));

```

Dodanie wielu sprzedaży jednocześnie:

```

$result = $ls->call('Deal/addDeals', array(
    'deals' => array(
        0 => array(
            'name' => 'Sprzedaż produktu' // nazwa (wymagana)
        ),
        1 => array(
            'name' => 'Sprzedaż usługi' // nazwa (wymagana)
        )
    )
));

```

Pod dodaniem sprzedaży na tablicy pojawia się informacja o tej aktywności, aby tego uniknąć należy wyłączyć komunikat przy pomocy przełącznika `_wall`:

```

$result = $ls->call('Deal/addDeal', array(
    'deal' => array(
        'name' => 'Sprzedaż produktu', // nazwa (wymagana)
        'company' => array( // firma (składnia jak dla dod./edycji firmy)
            'name' => 'Firma'
        ),
        '_wall' => false, // nie pokazuj aktywności na tablicy
    )
));

```



## 9.3 Edycja

Składnia identyczna jak dla dodawania, wymagany parametr *id*.

Zmiana nazwy i opisu:

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'name' => 'Super sprzedaż produktu', // nowa nazwa
        'note' => 'nowy opis'
    )
));
```

Zmiana statusu:

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'status' => 'open' // nowy status (open, won, lost, outdated)
    )
));
```

Zaznaczenie działania w ramach etapu:

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'stages' => array(
            'b7193e52-d695-8f6d-5da7-7fb9e792273c' // ID działania
        )
    )
));
```

## 9.4 Usuwanie

```
$result = $ls->call('Deal/deleteDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
    )
));
```

## 9.5 Tagi

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'tag_add' => 'nowy tag1,nowy tag2', // tagi do dodania
        'tag_remove' => 'tag3,tag4', // tagi do usunięcia
    )
));
```

## 9.6 Zadanie

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'todo' => array(
            'title' => 'Nowe zadanie', // tytuł (wymagany)
            'date' => '2013-10-10', // data (opcjonalna)
                                format rrrr-mm-dd lub rrrr-mm-dd gg:mm
            'description' => 'opis', // opis (opcjonalny)
            'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8' // id
                                użytkownika, do którego zostanie przypisane zadanie (opcjonalne)
        )
    )
));
```

## 9.7 Pliki

Dodanie na tablicę sprzedaży pliku dostępnego pod wskazanym adresem URL.

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'file' => array(
            'url' => 'http://domain/file.ext', // url pliku (wymagany)
            'userpwd' => 'login:password', // login i hasło do autoryzacji
                                (opcjonalne)
            'description' => 'opis', // opis (opcjonalny)
        )
    )
));
```

## 9.8 Tablica

### 9.8.1 Dodanie wpisu na tablicy

```
$result = $ls->call('Deal/addDealNote', array(
    'deal' => array(
        'id' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80', // Id sprzedaży
        'note' => 'treść wpisu', // treść wpisu
        'tags' => 'tag1,tag2' // opcjonalne tagi
    )
));
```

### 9.8.2 Pobranie wpisów z tablicy

```
$result = $ls->call('Deal/getWall', array(
    'id' => '9497f068-8542-a93d-5721-1a636cebba4a'
));
```

## 9.9 Wyszukiwanie

### 9.9.1 Wyszukiwanie wg cech

```
$result = $ls->call('Deal/getAll', array(
// dla parametrów podajemy jedną wartość lub kilka oddzielonych przecinkami
// parametry są opcjonalne ale należy podać przynajmniej jeden warunek
// nazwa sprzedaży
'names' => 'Ubezpieczenia',
// status sprzedaży - open, won, lost, outdated
'status' => 'outdated,open',
// id procesów sprzedaży
'processes' => 'c93ce2dd-aa21-1172-34b0-430af636c674',
// id etapów i podetapów sprzedaży
'main_stages' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244',
'stages' => '430af6dd-aa21-1172-34b0-c674f636c674',
// id firm i osób
'companies' => '3fd46cc0-9de5-4269-4b04-85147f7cb42f',
'contacts' => '5c480d66-7521-3fbf-f372-bdb41a393c58',
// dla parametrów związanych z datą można podać pojedynczą wartość
(traktowana jest jak wartość 'od') lub tablicę z wartościami 'od' i 'do'
// data utworzenia
'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
// data modyfikacji
'modified' => '2015-01-15',
// data zmiany statusu
'status_change_date' => '2015-01-15',
// data finalizacji
'date_end' => '2015-01-15',
// data ostatniej aktywności
'last_active' => '2015-01-20',
// loginy właścicieli rozdzielone przecinkami
'owner_login' => 'jan.kowalski@firma.com',
// opcjonalnie sposób porównywania nazwy, 'like' - wzorzec,
'equal' - dokładne dopasowanie, domyślnie 'like'
'cond' => 'like',
// Tagi
'tags' => 'tag1,tag2',
// Sposób wyszukiwania po tagach ('or' lub 'and', domyślnie 'or')
'tags_condition' => 'and',
'limit' => '1000', // Liczba sprzedaży (domyślnie wszystkie)
'offset' => '0' // Liczba pomijanych początkowych (domyślnie 0)
));
```

### 9.9.2 Wyszukiwanie wg frazy

Podstawowe:

```
$result = $ls->call('Search/getResult', array(
'object_type' => 'deal', // typ obiektu
'q' => 'moja sprzedaż' // szukana fraza
));
```

**Zaawansowane:**

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'deal', // typ obiektu
    'q' => 'moja sprzedaż', // szukana fraza
    'type' => 'like', // sposób wyszukiwania like lub equal
    'condition' => 'AND', // AND lub OR, spójnik dla warunków na kolejnych
    słowach frazy, decyduje czy w kontekście danego obiektu muszą wystąpić wszy
    stkie słowa z frazy (AND) czy tylko jedno z nich (OR)
    'limit' => 10, // liczba wyników
    'offset' => 0 // offset, przydatny do stronicowania
));
```

## 10 Zadania

### 10.1 Pobranie danych

Jedno zadanie:

```
$result = $ls->call('Todo/get', array(
    'id' => '9497f068-8542-a93d-5721-1a636cebba4a' // id zadania
));
```

Wiele zadań:

```
$result = $ls->call('Todo/getForObject', array(
    // typ obiektu - contact, company, deal, space
    'object_type' => 'contact',
    // id obiektu
    'id' => 'c93ce2dd-aa21-1172-34b0-430af636c674'
));
```

### 10.2 Dodanie

Bardzo proste:

```
$result = $ls->call('Todo/addTodo', array(
    'todo' => array(
        'title' => 'Nowe zadanie' // tytuł (wymagany)
    )
));
```

Proste:

```
$result = $ls->call('Todo/addTodo', array(
    'todo' => array(
        'title' => 'Nowe zadanie', // tytuł (wymagany)
        'date' => '2013-10-10', // data (opcjonalna)
                                format rrrr-mm-dd lub rrrr-mm-dd gg:mm
        'description' => 'opis', // opis (opcjonalny)
        'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8' // id
                                użytkownika, do którego zostanie przypisane zadanie (opcjonalne)
    )
));
```

Więcej danych:

```
$result = $ls->call('Todo/addTodo', array(
    'todo' => array(
        'title' => 'Nowe zadanie', // tytuł (wymagany)
        'date' => '2013-10-10', // data (opcjonalna)
                                format rrrr-mm-dd lub rrrr-mm-dd gg:mm
        'description' => 'opis', // opis (opcjonalny)
        'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8' // id
                                użytkownika, do którego zostanie przypisane zadanie (opcjonalne)
        'is_completed' => false, // wykonane true|false, domyślnie false
                                (opcjonalne)
        'is_private' => false, // prywatne true|false, domyślnie false
                                (opcjonalne)
        'priority' => 0, // priorytet 0-normalny, 1-wysoki, domyślnie 0
                                (opcjonalne)
        'status_id' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244', // id status
                                (opcjonalne)
        'type_id' => 'bb43bf24-b243-ad7c-bbac-1f2b33f48254', // id typu
                                (opcjonalne)
    )
));
```

```

'dataset' => array( // wartości dla pól dodatkowych, identyfikatory
                  // pól do odczytania w ustawieniach LiveSpace,
                  // inne dla każdej instancji
                  '87de5630-d889-bdb1-bd09-d61335d27ff8' => '56e966a4-f265-39ac
                                                            7c6e-ccf5a8caebb1',
                  // dla pól typu select podajemy id odpowiedzi
                  '1e5d3fa6-da59-da59-4657-4998c2184e6d' => 'value_2'
                  // dla pól tekstowych podajemy pełną odpowiedź
                ),
'objects' => array( // powiązane elementy (opcjonalne)
                  array(
                    'type' => 'contact', // contact|company|deal|space
                    'id' => '56e966a4-f265-39ac-7c6e-ccf5a8caebb1'
                  ),
                  array(
                    'type' => 'company',
                    'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8'
                  )
                )
            );

```

Dodanie wielu zadań jednocześnie:

```

$result = $ls->call('Todo/addTodos', array(
    'todos' => array(
        0 => array(
            'title' => 'Zadanie 1' // tytuł (wymagany)
        ),
        1 => array(
            'title' => 'Zadanie 2' // tytuł (wymagany)
        )
    )
));

```

## 10.3 Edycja

Składnia identyczna jak dla dodawania, wymagany parametr *id*.

Zmiana nazwy i opisu:

```

$result = $ls->call('Todo/editTodo', array(
    'todo' => array(
        'id' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244' // id zadania
        'title' => 'Nowe zadanie', // tytuł
        'description' => 'opis', // opis
    )
));

```

## 10.4 Usunięcie

```

$result = $ls->call('Todo/removeTodo', array(
    'todo' => array(
        'id' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244' // id zadania do
                                                         usunięcia
    )
));

```

## 11 Tablica

### 11.1 Pobranie danych

```
$result = $ls->call('Wall/getList', array(  
    // typ obiektu - contact, company, deal, space (opcjonalnie)  
    'object_type' => 'contact',  
    // id obiektu (opcjonalnie)  
    'object_id' => 'c93ce2dd-aa21-1172-34b0-430af636c674',  
    // typ obiektu, do którego jest dodana notatka (opcjonalnie,  
    // rozdzielone przecinkami)  
    'type_name' => 'note,email,activity,phone',  
    // data od (opcjonalnie)  
    'date_from' => '2015-05-12',  
    // data do (opcjonalnie)  
    'date_to' => '2016-01-23',  
    // wyszukiwana fraza (opcjonalnie)  
    'query' => 'tekst',  
    'limit' => '100', // Liczba elementów (domyślnie 50)  
    'offset' => '0' // Liczba pomijanych początkowych (domyślnie 0)  
));
```

## 12 Obsługa formularzy

Przykład obsługi formularza osadzonego na stronie WWW. Efektem będzie dodanie w LiveSpace nowej osoby, firmy oraz zadania przypominającego o kontakcie.

```
// Załączenie SDK
require_once 'livespace.php';

// Dane z formularza
$data = $_POST;

// Walidacja danych itd.
// ...

// Zapis do LiveSpace
$ls = new LiveSpace(array(
    'api_url' => 'https://SUBDOMENA.livespace.pl',
    'api_key' => API_KEY,
    'api_secret' => API_SECRET
));

$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'name' => $data['imie_i_nazwisko'], // Imię i nazwisko
        'emails' => array( // Adres email
            $data['e-mail']
        ),
        'phones' => array( // Telefon
            $data['telefon']
        ),
        'addresses' => array( // Adres
            array(
                'city' => $data['miasto']
                'street' => $data['ulica']
            )
        ),
        'todo' => array( // Zadanie
            'title' => 'nawiazac kontakt',
            'date' => date('Y-m-d', strtotime('+3days'))
        ),
        'company' => array( // Dane firmy
            'name' => $data['firma'], // Nazwa
            'nip' => $data['nip'], // NIP
            'emails' => array( // Adres email
                $data['e-mail']
            ),
            '_wall' => false,
            '__check_if_exists' => false // Wymuszenie dodania nowego
            kontaktu,
        ),
        '__check_if_exists' => false, // Wymuszenie dodania nowego
        kontaktu,
        '_wall' => false
    )
));
```



```
// Zapisanie wyniku do logów i przekierowanie
if (!$result->getStatus()) { // Gdy wystąpi błąd
    saveLog('Błąd:' . $result->__toString());
    Location('/zapis_bledny.html');
} else { // Zapis poprawny
    $contactData = $result->getData();
    saveLog('Wynik:' . $result->__toString());
    Location('/zapis_poprawny.html');
}
```