LiveSpace API

Update: November 09, 2016
Version: 1.1

1 Introduction

In this document, we have described the ways in which you can communicate with the Livespace API, modify parameters, and make use of SDK. We have also provided examples of calls of the most popular methods.

2 Content

3 Communication

3.1 Request

API requests are executed in the form of HTTP requests (the POST method). The module name and the method name are specified in the address while the method input data and other data required by, for example, the authorization mechanisms are conveyed in the call parameters.

3.1.1 Address format

https://DOMAIN.livespace.pl/api/public/OUTPUT_FORMAT/MODULE/METHOD

The meaning of the specific address elements:

- DOMAIN – subdomain (the name of your Livespace account)
- OUTPUT_FORMAT – a format in which the output data will be provided. Acceptable values: json, xml, php (the result of the *serialize()* method).
- MODULE – the name of an application module, e.g. *Contact, Deal, Todo* etc.
- METHOD – the name of a method in a specific module, e.g. *addContact*

Example:
https://SUBDOMAIN.livespace.io/api/public/json/Contact/addContact

3.1.2 Authorization

Currently, we only support authorization via the API key of a specific Livespace user and unique tokens generated **for every request.**

An API_KEY key (sent along with every request) and an API_SECRET key (not sent in API communication as its purpose is to generate a control sum) can be obtained in the Livespace settings (the API tab) and they are the same for all requests. The keys are related to specific user accounts. After you enter the API credentials of a specific user, all operations are executed on their behalf and with their permission level.

3.1.2.1 Example

Getting a token which can be used in the next request.

URL:

```
https://SUBDOMAIN.livespace.io/api/public/json/_Api/auth_call
      /_api_method/getToken
```

The POST parameters:

```
_api_auth: "key"
_api_key: API_KEY
```

Outcome:

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

3.1.3 Request execution

Among parameters, we send a SHA control sum that is calculated as a result of the *sha1* function on the character string created from the following concatenation: API_KEY, TOKEN, and API_SECRET.

Other parameters depend on the requirements of specific methods.

Request parameters are sent either in the form of multiple POST parameters or in the JSON format as the value of a single *data* parameter.

3.1.3.1 Example

URL:

```
https://SUBDOMAIN.livespace.io/api/public/json/Contact/addContact
```

The POST parameters:

```
data:[
    _api_auth: "key"
    _api_key: API_KEY
    _api_sha: SHA
    _api_session: SESSION_ID
    contact: [
        firstname: "John"
        lastname: "Smith"
    ]
]
```

The *data* parameter value is sent in the JSON format:

```
data:
{"_api_auth":"key","_api_key":"API_KEY","_api_sha":"SHA","_api_session":
"SESSION_ID","contact":{"firstname":"John","lastname":"Smith"}}
```

3.2 Outcome

The outcome is sent in the selected format (we suggest that you use the JSON format) as a four-element table with the following fields:

- *status* – true/false – it informs whether a request has been properly executed
- *result* – a code number (you can find its meaning in the list below)
- *data* – the data provided by the called method, usually in the table form
- *error* – error description

The *data* and *error* values depend on a method that you choose.

Different methods may use their own codes exceeding the data below.

The meanings of the *result* code numbers have been shown below.

- Correct
  - 200 – OK
- Method error
  - 400 – general method error
  - 420 – validation error
- API handling errors
  - 500 – general API error
  - 514 – incorrect module
  - 515 – incorrect method
  - 516 – incorrect output format
  - 520 – database communication error
  - 530 – user not logged in
  - 540 – no permission
  - 550 – parameter-handling error
- Authorization method errors
  - 560 – incorrect method
  - 561 – incorrect parameters
  - 562 – incorrect key
  - 563 – no authorization
  - 564 – general error

3.2.1 Example

An example of a request outcome for the *Contact/addContact* method:

```
data: [
      id: 123
      firstname: John
      lastname: Smith
]
error: null
result: 200
status: true
```

4 SDK

In order to provide you with more ways in which the API can be used in PHP, we have prepared an SDK kit that facilitates communication with Livespace, method calling, and outcome interpretation.

The requirements of the kit are: PHP 5.2 or higher and json / curl extensions.

4.1 Example

Adding a contact with the use of the SDK kit.

Setting the API parameters:

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

Preparing the method parameters:

```
$contactData = array(
    'contact' => array(
        'firstname' => 'John',
        'lastname' => 'Smith'
    )
);
```

The request execution:

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

The outcome interpretation:

```
if ($result->getStatus()) {
    echo 'Request executed correctly' . "\r\n";
    var_export($result->getResponseData());
} else {
    echo 'Error occurred #' . $result->getResult()  . ":\r\n";
    print_r($result->getError());
}
```

5 Identifiers

In the API communication, you can use object identifiers other than those that are present in application. For example, you can use the identifiers from the URL addresses.

You can show API identifiers on the profiles of contacts and deals. To do this, turn on the option “Show API identifiers on the object profiles” in the account settings (under the API tab).

The identifiers of, for example, additional fields or sales process stages are visible under the API tab in the account settings.

6 Available methods

For the sake of clarity, we have prepared examples based on the SDK kit. If you decide not to use the SDK, you should follow the communication pattern as described in the Chapter 3 while the method names and their parameters that are essential for your request can be sent according to the pattern shown in our examples.

Let’s say that the $*ls* object has been defined:

```
require_once 'livespace.php';

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

6.1 Sample request

A request that displays the sent parameters as an outcome:

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

6.2 The logged-in user data

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

7 Contacts - people

7.1 Getting data

Single contact:

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

Getting data for many people (basic information):

```
$result = $ls->call('Contact/getAllSimple', array(
    'type' => 'contact',
    'limit' => '1000', // Number of contacts (all by default)
    'offset' => '0' // Offset (number of contacts skipped initially ,
                                                            0 by default)
));
```

Getting data for many people (detailed information) (see also 6.3.11 search by phrase):

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'contact',
// for parameters we provide one or more values divided by comas
// parameters are optional, however you need to provide at least one
// parameter
    // first names
    'firstnames' => 'john,charles'
    // last names
    'lastnames' => 'smith'
    // company ID
    'companies' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80',
    // email address
    'emails' => 'john.smith@company.com',
    // phone numbers
    'phones' => '666-55-44,333444555',
    // for parameters connected with dates, you can provide only one value
    // (treated as ‘from’ value)or a board with ‘from’ and ‘to’ values
    // creation date
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // modification date
    'modified' => '2015-01-15',
    // last activity date
    'last_active' => '2015-01-20',
    // owner’s logins devided by comas
    'owner_login' => 'john.smith@company.com',
    // first and last names mode of comparison, ‘like’ – example, ‘equal’ –
    exact match, default ‘like’
     'condition' => 'like',
    // tags
    'tags' => 'tag1,tag2',
    // searching by tags (‘or’ or ‘and’, default ‘or’)
    'tags_condition' => 'and',
    'limit' => '1000', // Number of contacts (all  by default)
    'offset' => '0' // Number of contacts skipped initially ,
                                                            0 by default)
));
```

7.2 Adding

Very simple:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'name' => 'John Smith' // first and last name (required)
    )
));
```

Simple:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'John', // first name (required)
        'lastname' => 'Smith', // last name (required)
        'emails' => array( // email addresses
            'john@smith.com',
            'john.smith@company.com’
        ),
        'phones' => array( // phone numbers
            '1234567890',
            '0987654321'
        ),
        'company' => array( // companies
            'name' => 'Company Inc.'
        )
    )
));
```

More data:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'John', // first name (required)
        'lastname' => ' Smith ', // last name (required)
        'note' => 'note, // note
        'position' => 'CEO, // position
        'contact_source' => 'webpage form', // source
        'created' => '2013-10-10 11:22:33', // date of creation (optional)
        'www' => 'private-page.com', // webpage
        'emails' => array( // email addresses
            0 => array(
                'email' => 'john.smith@company.com' // email address
            ),
            1 => array(
                'email' => 'john.smith@company.com', // email address
                'is_default' => 1
            )
        ),
        'phones' => array( // phone numbers
            0 => array(
                'phone_no' => '1234567890', // phone number
                'type' => 1, // type of device: 1 – telephone,
                                                  2 - mobile, 3 - fax
            ),
            1 => array(
                'phone_no' => '0987654321', // phone number
                'type' => 2, // type of device: 1 – telephone,
                                              2 - mobile, 3 - fax
                'is_default' => 1
            )
        ),
        'addresses' => array( // addresses
            0 => array(
                'street' => 'Street 1/2', // street
                'city' => 'Warsaw', // city
                'postcode' => '00-999', // postal code
                'province' => 10, // province
                'country' => 'Poland' // country name
            )
        ),
        'company' => array( // company – same syntax as to adding
                            // companies or ID's
      'name' => 'Company Inc.', // company name, required
      'nip' => '111-222-55-00', // VAT Identification Number
      'regon' => '141981336' // regon
              ),
        'groups' => array( // groups to which contact will be added
            '87de5630-d889-bdb1-bd09-d61335d27ff8',
            '1e5d3fa6-da59-da59-4657-4998c2184e6d'
        ),
        'dataset' => array( // additional slots' values, slots'
identifiers, to be read in Livespace settings, different for each instance
            '87de5630-d889-bdb1-bd09-d61335d27ff8' => '56e966a4-f265-39ac-
7c6e-ccf5a8caebb1', // for select slot type – answer ID
            '1e5d3fa6-da59-da59-4657-4998c2184e6d' => 'value_2'
                    // for text box provide full answer
        ),
        'owner_id' => 'ff3a0bdb-7348-50c8-071a-ced692fdb898', // user's ID,
                                                 // the contact owner – optional
        'notification' => 'Content'// Notification for the contact owner
                                                          (optional)
      )
));
```

Adding many people at once:

```
$result = $ls->call('Contact/addContacts', array(
    'contacts' => array(
        0 => array(
            'firstname' => 'Michael', // first name (required)
            'lastname' => 'Jones' // last name (required)
        ),
        1 => array(
            'firstname' => 'Charles', // first name (required)
            'lastname' => 'Taylor' // last name (required)
        )
    )
)); switch
```

By default, when adding a contact, the system will check whether such contact already exists, to avoid data duplication. In order to enforce adding a new contact, we should use *__check_if_exists* switch:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'John', // first name (required)
        'lastname' => 'Smith', // last name (required)
        '__check_if_exists' => false, // enforce adding
    )
));
```

After a new contact is added, the informtion about this activity will appear on the wall. In order to avoid that, you need to turn this notification off, on the wall, using the _*wall* switch:

```
$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'firstname' => 'John', // first name (required)
        'lastname' => 'Smith', // last name (required)

        '_wall' => false, // do not show activity on the wall
    )
));
```

7.3 Edition

Syntax similar to adding proces, *id* parameter required.

Changing the last name and adding email address:

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'lastname' => 'Smith', // new last name
        'emails' => array( // new, another email address
            'john.smith@company.com'
        )
    )
));
```

Assigning to the company:

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

Deleting from company:

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

Adding new phone numbers and deleting old ones:

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'phones' => array( // phone numbers
            0 => array(
                'phone_no' => '1234567890', // phone number
                'type' => 1, // type of device: 1 - telephone,
                                              2 - mobile, 3 - fax
            ),
            1 => array(
                'phone_no' => '0987654321', // phone number
                'type' => 2, // type of device: 1 - telephone,
                                              2 - mobile, 3 - fax
                'is_default' => 1
            ),
            '_delete_old' => false // delete all?
        ),
    )
));
```

7.4 Removing

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

7.5 Tags

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

7.6 Task

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'todo' => array(
            'title' => 'New task',  // title (required)
            'date' => '2013-10-10',  // date (optional)
                                   format yyyy-mm-dd or yyyy-mm-dd hh:mm
            'description' => 'description', // description (optional)
            'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8'  // id of
user to which the task will be assigned
        )
    )
));
```

7.7 Files

Adddding a file to contact’s wall (under specific URL address)

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'file' => array(
            'url' => 'http://domain/file.ext',  // file url(required)
            'userpwd' => 'login:password', // login and password for
authorisation purposeslogin(optional)
            'description' => 'description',  // description (optional)
        )
    )
));
```

7.8 Wall

7.8.1 Adding a post to a wall

```
$result = $ls->call('Contact/addContactNote', array(
    'contact' => array(
        'id' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80', // person’s id
        'note' => 'post content', // post content
        'tags' => 'tag1,tag2' // optional tags
    )
));
```

7.8.2 Getting posts from a wall

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

7.9 Searching

7.9.1 Searching by attributes

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'contact',
    // parameters - provide one value or more, divided by comas
    // parameters are optional, however you need to provide at least one
    'firstnames' => 'john,charles' // first names
    'lastnames' => 'smith' // last names
    // company ID
    'companies' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80',
    // email address
    'emails' => ' john.smith@company.com',
    // phone number
    'phones' => '666-55-44,333444555',
    // for parameters connected with dates, you can provide only one value
   (treated as 'from' value)or a board with 'from' and 'to' values
    // creation date
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // modification date
    'modified' => '2015-01-15',
    // last active date
    'last_active' => '2015-01-20',
    // owner's logins divided by comas
    'owner_login' => ' john.smith@company.com',
    // optional first and last names mode of comparison, 'like' - example,
'equal' - exact match, default 'like'
    'condition' => 'like',
    // Tags
    'tags' => 'tag1,tag2',
    // searching by tags ('or' or 'and', default 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Number of contacts (all  by default)
    'offset' => '0' // Number of contacts skipped initially ,
0 by default)
));
```

7.9.2 Searching by phrases

Basic:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'contact', // object type
    'q' => 'john' // searched phrase
));
```

Advanced:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'contact', // object type
    'q' => 'john smith', // searched phrase
    'type' => 'like', // like or equal search mode
    'condition' => 'AND', // AND or OR, the conjunction for parameters in
the following words in a phrase, decides whether in a particular object,
there have to be all words within the phrase (AND) or only one of them
(OR)

    'limit' => 10, // number of results
    'offset' => 0 // offset, useful for page numbering
));
```

8 Contacts - companies

8.1 Getting data

Single company:

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

Getting data for many companies (basic information):

```
$result = $ls->call('Contact/getAllSimple', array(
    'type' => 'company',
    'limit' => '1000', // Number of contacts (all by default)
    'offset' => '0' // Offset (number of contacts skipped initially ,
0 by default)));
```

Getting data for many companies (detailed information) (see also 8.9.2 search by phrase):

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'company',
     // parameters - provide one value or more, divided by comas
    // arameters are optional, however you need to provide at least one
'for' parameter
    // company name
    'names' => 'Bank XYZ,Company ABC',
    // VAT Identification Number (optional format 123-456-78-90,
1234567890, PL123-45-67-890)
    'nip' => '123-456-78-90',
    // REGON
    'regon' => '1234567890123',
    // email address
    'emails' => 'contact@company.com',
    // phone number
    'phones' => '666-55-44,333444555',
    // for parameters connected with dates, you can provide only one value
   (treated as 'from' value)or a board with 'from' and 'to' values
    // creation date
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // modification date
    'modified' => '2015-01-15',
    // last activity date
    'last_active' => '2015-01-20',
    // owners' login divided by comas
    'owner_login' => 'john.smith@company.com',
    // optional - comparing the names, VAT Identification Numbers and
regon, 'like' - example, 'equal' - exact match, 'like' - default
    'cond' => 'like',
    // Tags
    'tags' => 'tag1,tag2',
    // Ways of searching by tags ('or' or 'and', default 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Number of contacts (all by default)
    'offset' => '0' // Offset (number of contacts skipped initially ,
0 by default)
));
```

8.2 Adding

Very simple:
```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Company Inc.' // company name (required)
    )
));
```

Simple:
```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Company Inc.' // company name (required)
        'emails' => array( // email addresses
            'office@company.com’
        ),
        'phones' => array( // phone numbers
            '1234567890',
            '0987654321'
        )
    )
));
```

More data:
```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => ''Company Inc.', // company name (required)
        'nip' => '111-222-55-00', // VAT Identification Numbers
        'regon' => '141981336', // regon
        'note' => 'note', // note
        'created' => '2013-10-10 11:22:33', // date of adding (optional)
        'www' => 'private-page.com', // web page
        'emails' => array( // email addresses
            0 => array(
                'email' => ' john.smith@company.com’ // email address
            ),
            1 => array(
                'email' => 'john.smith@company.com', // email address
                'is_default' => 1
            )
        ),
        'phones' => array( // phone numbers
            0 => array(
                'phone_no' => '1234567890', // phone number
                'type' => 1 // phone type: 1-phone, 2-mobile, 3-fax
            ),
            1 => array(
                'phone_no' => '0987654321', // phone number
                'type' => 2, // phone type
                'is_default' => 1
            )
        ),
        'addresses' => array( // addresses
            0 => array(
                'street' => 'Street 1/2', // street
                'city' => 'Warsaw', // city
                'postcode' => '00-999' , // postal code
                'country' => 'Poland' // country name
            )
        ),
        'groups' => array( // groups to which contact will be added
            '87de5630-d889-bdb1-bd09-d61335d27ff8',
         ),
        'dataset' => array( // additional slots' values, slots'
identifiers, to be read in Livespace settings, different for each instance
            '87de5630-d889-bdb1-bd09-d61335d27ff8' => '56e966a4-f265-39ac-
7c6e-ccf5a8caebb1', // for select slot type - answer ID
            '1e5d3fa6-da59-4657-4998c2184e6d' => 'value_2'
                    // for text box provide full answer
        ),
        'owner_id' => 'ff3a0bdb-7348-50c8-071a-ced692fdb898', // user's ID,
the contact owner - optional
        'notification' => 'Treść'// Notification for the contact owner
(optional)     )
));
```

Adding many companies at once:

```
$result = $ls->call('Contact/addCompanies', array(
    'companies' => array(
        0 => array(
            'name' => 'Company Inc.' // company name (required)
        ),
        1 => array(
            'name' => 'Bank XYZ' // company name (required)
        )
    )
));
```

By default, when adding a contact, the system will check whether such contact already exists, to avoid data duplication. In order to enforce adding a new contact, we should use *__check_if_exists* switch:

```
$result = $ls->call('Contact/addCompany', array(
    'company' => array(
        'name' => 'Company Inc.' // company name (required)
        '__check_if_exists' => false, // enforce adding
    )
));
```

After a new contact is added, the informtion about this activity will appear on the wall. In order to avoid that, you need to turn this notification off, on the wall, using the *_wall* switch:

```
$result = $ls->call('Contact/addCompany', array(
    ' company ' => array(
        'name' => 'Company Inc.' // company name (required)
        '_wall' => false, // do not show activity on the wall
    )
));
```

8.3 Editing

Syntax similar to adding proces, id parameter required.

Changing the company name and adding email address:

```
$result = $ls->call('Contact/editCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'name' => 'Company Inc.' // new company name
        'emails' => array( // new, another email address
            'marketing@company.com'
        )
    )
));
```

Adding new phone numbers and deleting old ones::

```
$result = $ls->call('Contact/editCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'phones' => array( // phone numbers
            0 => array(
                'phone_no' => '1234567890', // phone number
                'type' => 1, // type of device: 1 – telephone,
                                              2 - mobile, 3 - fax
            ),
            1 => array(
                'phone_no' => '0987654321', // phone number
            ),
            '_delete_old' => false // delete all?
        ),
    )
));
```

8.4 Removing

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

8.5 Tags

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

8.6 Task

```
$result = $ls->call('Contact/editCompany', array(
    'company' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'todo' => array(
            'title' => 'New task',  // title (required)
            'description' => 'description',  // description (optional)
            'date' => '2013-10-10',  // date (optional)
                      format yyyy-mm-dd or yyyy-mm-dd hh:mm
            'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8'  // id of
user to which the task will be assigned (optional)
        )
    )
));
```

8.7 Files

Addding a file to contact's wall (under specific URL address)

```
$result = $ls->call('Contact/editContact', array(
    'contact' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'file' => array(
            'url' => 'http://domain/file.ext',  // file url(required)
            'userpwd' => 'login:password', // login and password for
authorisation purposes (optional)
            'description' => 'description',  // description (optional)
        )
    )
));
```

8.8 Wall

8.8.1 Adding a post to a wall

```
$result = $ls->call('Contact/addCompanyNote', array(
    'company' => array(
        'id' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80', // company ID
        'note' => 'post content', // post content
        'tags' => 'tag1,tag2' // optional tags
    )
));
```

8.8.2 Getting posts from a wall

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

8.9 Searching

8.9.1 Searching by attributes

```
$result = $ls->call('Contact/getAll', array(
    'type' => 'company',
    // parameters - provide one value or more, divided by comas
    // arameters are optional, however you need to provide at least one
'for' parameter
    // company name
    'names' => 'Bank XYZ,Company ABC',
    // VAT Identification Number (optional format 123-456-78-90,
1234567890, PL123-45-67-890)
    'nip' => '123-456-78-90',
    // REGON
    'regon' => '1234567890123',
    // email address
    'emails' => 'contact@company.com',
    // telefon
    'phones' => '666-55-44,333444555',
    // for parameters connected with dates, you can provide only one value
   (treated as 'from' value)or a board with 'from' and 'to' values
    // creation date
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // modification date
    'modified' => '2015-01-15',
    // last active date
    'last_active' => '2015-01-20',
    // owner's logins divided by comas
    'owner_login' => ' john.smith@company.com',
    // optional company names, VAT Identification Number or regon mode of
comparison, 'like' - example, 'equal' - exact match, default 'like'
    'cond' => 'like',
    // Tags
    'tags' => 'tag1,tag2',
    // searching by tags ('or' or 'and', default 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Number of contacts (all  by default)
    'offset' => '0' // Number of contacts skipped initially,
(0 by default)
));
```

8.9.2 Searching by phrases

Basic:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'company', // object type
    'q' => 'bank' // searched phrase
));
```

Advanced:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'company', // object type
    'q' => 'bank prywatny', // searched phrase
    'type' => 'like', // like or equal search mode
    'condition' => 'AND', // AND or OR, the conjunction for parameters in
the following words in a phrase, decides whether in a particular object,
there have to be all words within the phrase (AND) or only one of them (OR)
```

```
    'limit' => 10, // number of results
    'offset' => 0 // offset, useful for page numbering
));
```

9 Deals

9.1 Getting data

Single sale:

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

Multiple sales (see also 6.5.11 search by phrase)):

```
$result = $ls->call('Deal/getAll', array(
    // parameters - provide one value or more, divided by comas
    // arameters are optional, however you need to provide at least one
'for' parameter
    // sale name
    'names' => 'Insurance',
    // deal status - open, won, lost, outdated
    'status' => 'outdated,open',
    // sales id
    'processes' => 'c93ce2dd-aa21-1172-34b0-430af636c674',
    // id main stages and substages id
    'main_stages' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244',
    'stages' => '430af6dd-aa21-1172-34b0-c674f636c674',
    // companies and contacts id
    'companies' => '3fd46cc0-9de5-4269-4b04-85147f7cb42f',
    'contacts' => '5c480d66-7521-3fbf-f372-bdb41a393c58',
    // for parameters connected with dates, you can provide only one value
   (treated as 'from' value)or a board with 'from' and 'to' values
    // creation date
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // modification date
    'modified' => '2015-01-15',
    // status change date
    'status_change_date' => '2015-01-15',
    // end date
    'date_end' => '2015-01-15',
    // last activity date
    'last_active' => '2015-01-20',
    // owners' login divided by comas
    'owner_login' => 'john.smith@company.com',
    // optional - comparing the names, VAT Identification Numbers and
regon, 'like' - example, 'equal' - exact match, 'like' - default
'cond' => 'like',
    // Tags
    'tags' => 'tag1,tag2',
    // Ways of searching by tags ('or' or 'and', default 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Number of sales (all by default)
    'offset' => '0' // Number of sales skipped initially,
0 by default)
));
```

9.2 Adding

Very simple:
```
$result = $ls->call('Deal/addDeal', array(
  'deal' => array(
      'name' => 'Service sale', // sale name (required)
      'company' => array('name' => 'Company') // syntax as for the company
    )
));
```

Simple:
```
$result = $ls->call('Deal/addDeal', array(
   'deal' => array(
      'name' => 'Service sale', // sale name (required)
      'date_end' => '2013-12-10', // end date
      'process_id' => '4c9d3e56-9aa7-ff1e-4597-88e70a0213bf', //process
                                                                (optional)
      'company' => array('name' => 'Company') // syntax as for the company
      'contact' => array('name' => 'John Smith') // syntax as for the
person    )
));
```

More data:

```
$result = $ls->call('Deal/addDeal', array(
    'deal' => array(
        'name' => 'Service sale', // sale name (required)
        'date_end' => '2013-12-10', // end date
        'process_id' => '4c9d3e56-9aa7-ff1e-4597-88e70a0213bf', // process
                                                                (optional)
        'company' => array( // company (syntax as for the company
adding/edition)
            'name' => 'Company'
        ),
        'contact' => array( // person (syntax as for the contact
adding/edition)

            'name' => 'John Smith'
        ),
        'note' => 'note', // note
        'created' => '2013-10-10 11:22:33', // addition date (optional)
        'dataset' => array( // additional slots' values, slots'
identifiers, to be read in Livespace settings, different for each instance
'4998c21d-53de-da59-4657-1e5d3fa684e6' => '8b977d9e-48ad-dc34-f780-
8bc088e6076b', // for select slot type - answer ID
            '7859680b-6091-dc05-bbd6-d5fe260319fe' => 'value_2'
                    // for text box provide full answer
        ),
        'stages' => array(
            'b7193e52-d695-8f6d-5da7-7fb9e792273c', // ID stage for on
            '447c38f5-7384-c41e-047b-6aacecac0919' => 0,// ID stage for off
            '6dd5e857-4a8e-d98e-8c00-64480ebb792a' => 1 // ID stage for on
        ),
        'budget' => array(
            array(
                'category_id' => '6dd5e857-4a8e-d98e-8c00-64480ebb792a',
                                 // ID or category name (required)
                'value' => '123000,00', // Value (required)
                'date' => '2013-11-01', // Date (optional)
                'is_forecast' => true, // Forecast or income (optional)
                'is_profit' => true, // Profit or cost (optional)
                'description' => 'Prognoza', // Description (optional)
                'discount_percent' => '10', // Discount in % (optional)
                'cycle_period' => '1 year', // Cycle period (time - day,
                             week, month, quearter, year) (optional)
                'cycle_repeat' => '2 month' // Cycle repeat (cycle repeat
                 time - day, week, month, quearter, year) (optional)
            ),
            array(
                'category_name' => 'Category 1', // ID or category name
(required)
                'value' => '12345.99', // Value (required)
                'date' => '2013-12-01', // Date (optional)
            ),
            '_delete_old' => false,// Delete previous values? (optional)
            '_delete_old_details_forecast' => 1,
                   // Delete forecast only? (1) / progress (0) (optional)
            '_delete_old_details_profit' => 1,
                   // Delete profit only? (1) / cost (0) (optional)
        ),
        'is_public' => false // Is the sale public? (no by default)
        'owner_id' => 'ff3a0bdb-7348-50c8-071a-ced692fdb898',
  // user's ID, the contact owner - optional
       'notification' => 'Treść'// Notification for the contact owner
(optional)
    )
));
```

Adding many sales at once:

```
$result = $ls->call('Deal/addDeals', array(
    'deals' => array(
        0 => array(
            'name' => 'Product sale' // sale name (required)
        ),
        1 => array(
            'name' => 'Service sale' // sale name (required)
        )
    )
));
```

After a new sale is added, the informtion about this activity will appear on the wall. In order to avoid that, you need to turn this notification off, on the wall, using the *_wall switch:*

```
$result = $ls->call('Deal/addDeal', array(
    'deal' => array(
        'name' => 'Product sale' // sale name (required)
        'company' => array( // company (syntax as for the company
adding/edition)

            'name' => 'Firma'
        ),
        '_wall' => false, // do not show activity on the wall
    )
));
```

9.3 Editing

Syntax similar to adding proces, id parameter required.
Changing the sales name and description:

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'name' => 'Awesome product sale', // new sale name
        'note' => 'nowy description'
    )
));
```

Status change:

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

Marking activity according to stage:

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

9.4 Removing

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

9.5 Tags

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

9.6 Task

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'todo' => array(
            'title' => 'New task',  // title (required)
            'date' => '2013-10-10',  // date (optional)
                       format yyyy-mm-dd or yyyy-mm-dd
hh:mm            'description' => 'description',  // description (optional)
            'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8'  // id of
user to which the task will be assigned (optional)
        )
    )
));
```

9.7 Files

Adddding a file to sales' wall (under specific URL address):

```
$result = $ls->call('Deal/editDeal', array(
    'deal' => array(
        'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8',
        'file' => array(
            'url' => 'http://domain/file.ext',  // file url (required)
            'userpwd' => 'login:password', // login and password for
authorisation purposes (optional)
            'description' => 'description',  // description (optional)
        )
    )
));
```

9.8 Wall

9.8.1 Adding a post to a wall

```
$result = $ls->call('Deal/addDealNote', array(
    'deal' => array(
        'id' => 'd075b5d4-9e60-8e5b-f436-4bf9c20dfb80', // sales ID
        'note' => 'post content', // post content
        'tags' => 'tag1,tag2' // optional tags
    )
));
```

9.8.2 Getting posts from a wall

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

9.9 Searching

9.9.1 Searching by attributes

```
$result = $ls->call('Deal/getAll', array(
    // parameters - provide one value or more, divided by comas
    // arameters are optional, however you need to provide at least one
'for' parameter
    // sales name
    'names' => 'Insurance',
    // deal status - open, won, lost, outdated
    'status' => 'outdated,open',
    // sales id
    'processes' => 'c93ce2dd-aa21-1172-34b0-430af636c674',
    // id main stages and substages id
    'main_stages' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244',
    'stages' => '430af6dd-aa21-1172-34b0-c674f636c674',
    // companies and contacts id
    'companies' => '3fd46cc0-9de5-4269-4b04-85147f7cb42f',
    'contacts' => '5c480d66-7521-3fbf-f372-bdb41a393c58',
    // for parameters connected with dates, you can provide only one value
   (treated as 'from' value)or a board with 'from' and 'to' values
    // creation date
    'created' => array('from' => '2015-01-01 12:00', 'to' => '2015-01-20'),
    // creation date
    'modified' => '2015-01-15',
    // modification date
    'status_change_date' => '2015-01-15',
    // status change date
    'date_end' => '2015-01-15',
    // end date
    'last_active' => '2015-01-20',
    // loginy właścicieli rozdzielone przecinkami
    'owner_login' => 'john.smith@company.com',
    // owners' login divided by comas
    // optional - comparing the names, 'like' - example, 'equal' - exact
match, 'like' - default          'cond' => 'like',
    'cond' => 'like',
    // Tags
    'tags' => 'tag1,tag2',
    // Ways of searching by tags ('or' or 'and', default 'or')
    'tags_condition' => 'and',
    'limit' => '1000', // Number of sales (all by default)
    'offset' => '0' // Number of sales skipped initially,
0 by default)
));
```

9.9.2 Searching by phrases

Basic:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'deal', // object type
    'q' => 'my sale' // searched phrase
));
```

Advanced:

```
$result = $ls->call('Search/getResult', array(
    'object_type' => 'deal', // object type
    'q' => 'my sale' // searched phrase
    'type' => 'like', // like or equal search mode
    'condition' => 'AND', // AND or OR, the conjunction for parameters in
the following words in a phrase, decides whether in a particular object,
there have to be all words within the phrase (AND) or only one of them (OR)
    'limit' => 10, // number of results
    'offset' => 0 // offset, useful for page numbering
));
```

10 Task

10.1 Getting data

Single task:

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

Multiple tasks:

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

10.2 Adding

Very simple:

```
$result = $ls->call('Todo/addTodo', array(
  'todo' => array(
      'title' => New task',   // title (required)
    )
));
```

Simple:

```
$result = $ls->call('Todo/addTodo', array(
   'todo' => array(
      'title' => ' New task',   // title (required)
      'date' => '2013-10-10',   // date (optional)
                       format yyyy-mm-dd or yyyy-mm-dd hh:mm
'description' => ' description',   // description (optional)
      'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8'   // id of user to
which the task will be assigned (optional)
    )
));
```

More data:

```
$result = $ls->call('Todo/addTodo', array(
    'todo' => array(
        'title' => New task',   // title (required)
        'date' => '2013-10-10',   // date (optional)
                       format yyyy-mm-dd or yyyy-mm-dd hh:mm
        'description' => description',   // description (optional)
        'user_id' => 87de5630-d889-bdb1-bd09-d61335d27ff8'   // id of user
to which the task will be assigned (optional)
        'is_completed' => false,   // completion true|false, default false
                                (optional)
        'is_private' => false, // private true|false, default false
                                (optional)
        'priority' => 0, // priority 0-normal, 1-high, default 0
                       (optional)
        'status_id' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244', // id status
                                                             (optional)
        'type_id' => 'bb43bf24-b243-ad7c-bbac-1f2b33f48254', // id type
                                                             (optional)
        'dataset' => array( // additional slots' values, slots'
identifiers, to be read in Livespace settings, different for each instance
            '87de5630-d889-bdb1-bd09-d61335d27ff8' => '56e966a4-f265-39ac
                                                     7c6e-ccf5a8caebb1',
                              // for select slot type - answer ID
            '1e5d3fa6-da59-da59-4657-4998c2184e6d' => 'value_2'
                             // for text box provide full answer
        ),
        'objects' => array( // connected objects (optional)
            array(
                'type' => 'contact', // contact|company|deal|space
                'id' => '56e966a4-f265-39ac-7c6e-ccf5a8caebb1'
            ),
            array(
                'type' => 'company',
                'id' => '87de5630-d889-bdb1-bd09-d61335d27ff8'
            )
        )
    )
));
```

Adding multiple tasks simultaneously:

```
$result = $ls->call('Todo/addTodos', array(
    'todos' => array(
        0 => array(
            'title' => 'Task 1' // title (required)
        ),
        1 => array(
            'title' => 'Task 2' // title (required)
        )
    )
));
```

10.3 Editing

Syntax similar to adding proces, id parameter required.

Changing the name and description:

```
$result = $ls->call('Todo/editTodo', array(
    'todo' => array(
        'id' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244' // task id
        'title' => 'New task',  // title (required)
        'description' => 'description',  // description (optional)
    )
));
```

10.4 Removing

```
$result = $ls->call('Todo/removeTodo', array(
    'todo' => array(
        'id' => 'aa23bf90-b405-bd6a-eeac-0f0be3f08244' // id of task to
remove
    )
));
```

11 Wall

11.1 Getting data

```
$result = $ls->call('Wall/getList', array(
    // object type - contact, company, deal, space (optional)
    'object_type' => 'contact',
    // object id (optional)
    'object_id' => 'c93ce2dd-aa21-1172-34b0-430af636c674',
    // object type, to which a note is added (optional, devided with comas)
    'type_name' => 'note,email,activity,phone',
    // date from (optional)
    'date_from' => '2015-05-12',
    // data to (optional)
    'date_to' => '2016-01-23',
    // searched phrase (optional)
    'query' => 'tekst',
    'limit' => '100', // number of objects (50 by default)
    'offset' => '0' // number of contacts skipped initially (0 by default)
));
```

12 Handling forms

An example of handling form embedded in a webpage. The result will be adding a new person, company and task reminding of the contact in Livespace.

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

// Form data
$data = $_POST;

// Data validation etc…
// ...

// Save in Livespace
$ls = new LiveSpace(array(
    'api_url' => 'https://SUBDOMAIN.livespace.io',
    'api_key' => API_KEY,
    'api_secret' => API_SECRET
));

$result = $ls->call('Contact/addContact', array(
    'contact' => array(
        'name' => $data['firstname_and_lastname'], // Name
        'emails' => array( // Email address
            $data['e-mail']
        ),
        'phones' => array( // Phone number
            $data['phone number']
        ),
        'addresses' => array( // Address
            array(
                'city' => $data['city']
                'street' => $data['street']
            )
        ),
        'todo' => array( // Task
            'title' => 'start contact',
            'date' => date('Y-m-d', strtotime('+3days'))
        ),
        'company' => array( // Company data
            'name' => $data['Company'], // company name
            'nip' => $data['nip'], // VAT Identification Number
            'emails' => array( // email address
                $data['e-mail']
            ),
            '_wall' => false,
            '__check_if_exists' => false // Enforcing adding new contact,
        ),
        '__check_if_exists' => false, // Enforcing adding new contact,
        '_wall' => false
    )
));
```

```
// Saving results to logs and redirecting
if (!$result->getStatus()) { // Error
    saveLog('Error:' . $result->__toString());
    Location('/error.html');
} else { // Correct log
    $contactData = $result->getData();
    saveLog('Result:' . $result->__toString());
    Location('/success.html');
}
```