

# Документация по Smsimple PHP-API

## Классы

### Класс SMSimpleException

Класс исключений. Ошибки происходящие из класса SMSimple выбрасываются в виде этого исключения.

### Класс SMSimple

SMSimple() - основной класс для работы с API.

Работа с библиотекой протекает по следующей стандартной схеме:

1. Создаём экземпляр класса SMSimple()
2. Вызываем функцию авторизации (получаем session\_id, который сохраняется как атрибут класса SMSimple)
3. Совершаем один или несколько вызовов прикладных функций (отправка SMS, проверка статуса и т.п.)
4. При возникновении ошибки класс вызывает исключение SMSimpleException с описанием ошибки (сообщение, возвращённое с сервера XMLRPC API, если ошибка произошла на стороне сервера)

Таким образом, минимальный блок работы с API выглядит примерно так:

```
require_once('./smsimple.class.php');

$sms = new SMSimple(array(
    'url' => 'http://api.smsimple.ru/',
    'username' => 'my_username',
    'password' => 'my_password',
));
try {
    $sms->connect();
```

```
$sms->call_method_1();
$sms->call_method_2();
$sms->call_method_3();

}
catch (SMSimpleException $e) {
    echo $e->getMessage();
}
```

## Методы

function \_\_construct ( \$params = array() ) ¶

Конструктор класса. Не вызывается напрямую. Вызывается интерпретатором PHP в момент создания класса `sms = SMSimple()`.

## Параметры

`$params` array()

[default = array()]

Массив параметров, обязательные: `username`, `password`, `url`; опциональные: `encoding`, `new_xmlrpc`.

`$params['username']` string

данные, указанные при регистрации (для входа в ЛК)

`$params['password']` string

данные, указанные при регистрации (для входа в ЛК)

`$params['url']` string

обычно должен быть <http://api.smsimple.ru>, но может меняться на url beta-api в случае, если Вы будете тестировать новые версии апи

`$params['encoding']` string

опционально

важный параметр, если страницы и скрипты Вашего сайта используют кодировку, отличную от `utf-8`, например, `windows-1251`.

`$params['new_xmlrpc']` string/bool

опционально

force-флаг, заставляющий при необходимости насильно использовать встроенный модуль `php5-xmlrpc`, а не написанную на PHP библиотеку `./lib/xmlrpc.inc`; может быть `true`, `false`, `'auto'`.

## Возвращает

`SMSimple`

Возвращает экземпляр класса `SMSimple`, с которым производить дальнейшие манипуляции (апи-вызовы).

### function **connect** ( ) ¶

Соединение с шлюзом SMSimple.ru.

## Параметры

Не имеет аргументов, использует `username` и `password`, заданные при создании класса `SMSimple`.

## Возвращает

`boolean`

Возвращает `true` в случае удачи, в случае неудачи выбрасывает `SMSimpleException`.

### function **send** ( \$origin\_id, \$phone, \$message, \$multiple = false ) ¶

Отправка одиночного сообщения

## Параметры

`$origin_id` int

номер подписи (номера можно узнать в личном кабинете на странице "[Подписи](#)"); можно передать `null`, тогда будет взята "Основная" подпись (см. ЛК "[Подписи](#)")

`$phone`

телефонный номер абонента (любой формат с кодом-телефоном, например `'8-916-1234567'` или `'79161234567'`) или несколько через запятую (пробелы разрешены, например `'79031234567, 89161234567'`)

`$message`

текст SMS-сообщения

`$multiple`

bool

`[default = false]` [\[опционально\]](#)

необязательный параметр, по-умолчанию `false`. Если вы отправляете сообщение сразу нескольким адресатам, то им присвоится один и тот же (`false`) или разные (`true`) номера.

## Возвращает

int

Возвращает номер (`id`) сообщения, по которому потом можно получать статус доставки/недоставки. Для `$multiple = false` вернёт одно число, для `$multiple = true` -- массив номеров в той последовательности, в которой шли телефоны абонентов.

```
function check_delivery ( $message_id ) ¶
```

Проверка статуса доставки сообщения, отправленного с помощью метода `send`.

## Параметры

`$message_id` int

номер SMS-сообщения, возвращённый методом `send`

## Возвращает

array()

Возвращает массив вида:

```
array(
    'sms_id' => 12341234, // то, что было передано в параметра $message_id
    'sms_count' => 3, // количество SMS-сообщений, соответствующих этому id, равное <кол-во_абонентов>*<кол-во_частей_сообщения>. Если
    'sms_delivered' => 2, // количество SMS-сообщений, которые были доставлены
    'sms_failed' => 0, // количество SMS-сообщений, которые не были доставлены (то есть являются уже окончательно недоставленными по т
    'sms_delayed' => 1, // количество SMS-сообщений, статус доставки которых ещё неизвестен (доставка в процессе). Для получения стат
```

```
function addJob ( $params = array() ) ¶
```

Создание новой запланированной рассылки

## Параметры

<code>\$params</code>	array()  [ <code>default = array()</code> ] Массив параметров, обязательные: <code>groups</code> , <code>title</code> , <code>template</code> ; optionalные: <code>origin_id</code> , <code>start_date</code> , <code>start_time</code> , <code>stop_time</code> .
<code>\$params['groups']</code>	array() массив с номерами групп, по которым произвести рассылку, например <code>array(123, 365, 436, 2134)</code>
<code>\$params['title']</code>	string название рассылки (для Вас), это название будет фигурировать только в списке рассылок в ЛК
<code>\$params['template']</code>	string текст SMS-сообщения (называется template, т.к. это шаблон, и в нём могут быть использованы параметры подстановки <code>%title%</code> , <code>%custom_1%</code> , <code>%custom_2%</code> из групп контактов)
<code>\$params['origin_id']</code>	int  [ <code>optional</code> ] номер подписи (номера можно узнать в личном кабинете на странице " <a href="#">Подписи</a> "); если этот параметр отсутствует, будет взята подпись, заданная как "Основная" в ЛК
<code>\$params['start_date']</code>	string  [ <code>optional</code> ] дата старта рассылки - дата в формате YYYY-MM-DD задаёт день, в который рассылка будет отправлена (если значение <code>start_date</code> не задано, то оно будет принять равным текущему дню)
<code>\$params['start_time']</code>	string  [ <code>optional</code> ] время старта рассылки - уточняет время старта в формате HH:MM или HH:MM:SS (если не задано, то будет равно <code>00:00:00</code> )
<code>\$params['stop_time']</code>	string  [ <code>optional</code> ] время остановки рассылки - уточняет время остановки в формате HH:MM или HH:MM:SS (если не задано, то будет равно <code>23:59:59</code> ). Это время будет использовано только в случае, когда рассылка не успела завершиться. Можно рассматривать его как "аварийное" время завершения. Это может быть важно по ряду причин, в частности, информация рассылки может потерять актуальность или время рассылки слишком позднее (ночь).

## Возвращает

int

Возвращает номер (id) новой рассылки.

function **origins** ( ) 

Получение списка подписей: **все** подписи текущего аккаунта

## Параметры

Нет параметров.

## Возращает

array()

Возвращает список подписей, доступных для использования в качестве поля "отправитель" (`origin_id`), массив массивов вида

```
array(
    array(
        'id' => 12345, // номер подписи (id) для использования при отправке send() или создании рассылки addJob
        'title' => 'my_origin', // собственно подпись
        'create_date' => '2013-01-22', // дата создания подписи
        'is_default' => 1, // эта подпись является подписью по-умолчанию, которая будет использована в случае, если подпись не задана
    ),
    ...
)
```

## function **addContactToGroup** ( \$params = array() ) ¶

Добавление нового контакта в существующую группу

## Параметры

`$params`

array()

`[default = array()]`

Массив параметров, обязательные: `group_id`, `phone`; опциональные: `title`, `custom_1`, `custom_2`.

<code>\$params['group_id']</code>	int	номер группы, в которую добавляем контакт
<code>\$params['phone']</code>	string	телефонный номер абонента (контакта)

<code>\$params['title']</code>	string
	<small>опционально</small>
<code>\$params['custom_1']</code>	"Имя" контакта ( <code>%title%</code> в шаблоне)
	string
	<small>опционально</small>
<code>\$params['cusomt_2']</code>	"Доп.поле 1" контакта ( <code>%custom_1%</code> в шаблоне)
	string
	<small>опционально</small>
	"Доп.поле 2" контакта ( <code>%custom_2%</code> в шаблоне)

## Пример параметров:

```
$params = array(
    'group_id' => 1,
    'phone'     => '7-926-111-22-33',
    'title'     => 'Василий Пупкин',
    'custom_1'  => '',
    'custom_2'  => '',
);
```

## Возвращает

int

Возвращает номер (id) новой рассылки.

function **searchContact** ( \$params = array() ) ¶

Поиск контакта по телефону. Поиск может осуществляться по всем группам или по избранным (одной или нескольким).

## Параметры

<code>\$params</code>	array()
	<small>default = array()</small>
	Массив параметров, обязательные: <code>phone</code> ; опциональные: <code>group_id</code> .

<code>\$params['phone']</code>	string
--------------------------------	--------

```
$params['group_id']
```

телефонный номер абонента (контакта), который требуется найти

int|array(int\*)

опционально

номер группы (`int`) или номера групп (`array()` из `int`), в которой/которых требуется найти контакт (т.е. ограничение поиска на эти группы)

## Пример параметров:

```
$params = array(
    'phone' => '7-926-111-22-33',
    'group_id' => array(1,2,3,4), // может также быть = 1 или = null или вообще отсутствовать
);
```

## Возвращает

Возвращает список найденных контактов, массив массивов вида

```
array(
    array(
        'id'      => 12345, // номер контакта
        'phone'   => '...' // |
        'title'   => '...' // | параметры контакта
        'custom_1' => '...' // |
        'custom_2' => '...' // |
        'group_id' => 222, // номер группы
        'group_title' => 'Группа 1', // название группы
    ),
    ...
)
```

```
function deleteContact ( $params = array() ) ¶
```

Удаление контакта из группы

## Параметры

`$params` array()

[`default = array()`]

Массив параметров, два варианта вызова: 1. с обязательным полем `contact_id`; 2. с обязательным `phone` и необязательным `group_id`.

`$params['contact_id']` int|array(int\*)

заранее известный номер контакта (или массив номеров), который надо удалить. Если этот параметр задан, остальные параметры - `phone`, `group_id` - игнорируются, происходит удаление конкретных контактов по их `id`. Если этот параметр не задан, то происходит поиск (с помощью метода [searchContact](#)) по параметрам `phone`, `group_id`, затем контактов по найденным номерам (`id`)

`$params['phone']` string

телефонный номер абонента (контакта), который требуется найти и удалить

`$params['group_id']` int|array(int\*)

[`optional`]  
номер группы (`int`) или номера групп (`array()` из `int`), в которой/которых требуется найти и удалить контакт (т.е. ограничение поиска и удаления только на этих группах)

## Возвращает

int

Возвращает количество удалённых контактов.

### Внимание!

Удаление контактов из заблокированных (участвующих в незавершённых запланированных рассылках) групп невозможно, контакты из таких групп будут молча пропущены.

function **deleteGroup** ( \$params = array() ) [¶](#)

## Удаление группы

## Параметры

`$params` array()

[`default = array()`]

Массив параметров, обязательные: `group_id`.

`$params['group_id']` int  
номер группы (`int`) для удаления

## Пример параметров:

```
$params = array(  
    'id' => 1234, // номер группы  
) ;
```

## Возвращает

bool Возвращает `true`, если группа обнаружена и удалена. В остальных случаях вызывает исключения: если группы заблокирована (участвует в незавершённых запланированных рассылках) или вообще не найдена по номеру.

## function **addGroup** ( \$params = array() ) ¶

Добавление (создание) группы

## Параметры

`$params` array()  
`[default = array()]`  
Массив параметров, обязательные: `title`; опциональные: `description`, `is_blacklist`.

`$params['title']` string  
название группы, заголовок, который будет виден в списке групп

`$params['description']` string  
 `[опционально]`  
более подробное описание (для комментариев); значение по-умолчанию `''`

`$params['is_blacklist']` bool  
 `[опционально]`  
будет ли эта группа использоваться как "чёрный список"; значение по-умолчанию `false`

## Пример параметров:

```
$params = array(
    'title'          => 'Название новой группы',
    'is_blacklist'  => false, // это будет группа для "чёрного списка"?
    'description'   => 'Поясняющее описание к группе',
);
```

## Возвращает

int

Возвращает номер вновь созданной группы.

function **getContactCount** ( \$params = array() ) ¶

Объём группы (количество контактов в группе)

## Параметры

`$params` array()

`[default = array()]`

Массив параметров, обязательные: `[id]`.

`$params['id']`

int

номер группы

## Пример параметров:

```
$params = array(
    'id' => 1234, // номер группы
);
```

## Возвращает

int

Возвращает количество контактов в группе.

```
function getContacts ( $params = array() ) ¶
```

Получение контактов из группы, возможно постранично

## Параметры

\$params array()

[default = array()]

Массив параметров, обязательные: `id`; опциональные: `offset`, `limit`;

`$params['id']`

int

номер группы

`$params['offset']`

int

[опционально]

начинать с этого контакта (по аналогии с SQL-командой `OFFSET`); по-умолчанию `0`

`$params['limit']`

int

[опционально]

ограничить объём выдачи (по аналогии с SQL-командой `LIMIT`); по-умолчанию `0`, что означает "не ограничивать"

## Пример параметров:

```
$params = array(
    'id' => 1234,      // номер группы
    'offset' => 100,   // - для постраничного получения
    'limit' => 20,     // - для постраничного получения
);
```

## Возвращает

array()

Возвращает контакты из группы, в виде массивов

```
array(
    array(
        'id' => 1,
        'title' => 'Вася',
        'phone' => '79031234567',
        'custom_1' => '',
        'custom_2' => '',
    ),
    ...
)
```

## function **get\_profile()** ¶

Получение информации о профиле

### Параметры

Не принимает параметров.

### Возращает

array()

Возвращает доступные данные по Вашему аккаунту в виде массива, например

```
array(
    'id' => 52579,
    'username' => 'test',
    'title' => 'Василий Васильевич Пупкин',
    'ussd_enabled' => 1,
    'ussd_balance' => -673,
    'ussd_leverate' => 0,
    'phone' => '03',
    'email' => 'vasya@pupkin.ru',
    'billing_type' => 'Предоплата',
    'balance' => 966.911,
    'leverate' => 0.0,
```

```
)
```

```
function ussd_session_start ( $phone, $optional = null, $encoding = 'GSM-7', $app_id = null ) ¶
```

Старт USSD сессии, инициирует начало обмена USSD-сообщениями. Эта функция ещё ничего не посыпает на телефон абонента, первая отправка USSD-меню осуществляется Вашим скриптом обратного вызова согласно идентификатору Вашего приложения.

## Параметры

`$phone` string

номер абонента, которому отправляется USSD

`$optional` string

`[default = null]`

любые данные для дополнительной идентификации, сопровождают сессию до закрытия, обычно ваш внутренний идентификатор сессии

`$encoding` string

`[default = 'GSM-7']`

`'GSM-7'` для меню на латинице (~120 символов) или `'UCS2'` для меню с русскими буквами (~60 символов)

`$app_id` string

`[default = null]`

идентификатор USSD-приложения согласно вашему ЛК (см. ["USSD-приложения"](#) в меню, пункт доступен, если у Вас открыт доступ к USSD). Идентификатор можно не задавать, если у Вас одно USSD-приложение (оно и вызовется), если у Вас несколько USSD-приложений и `app_id` не указан, будет возвращена ошибка

## Пример параметров:

```
$this->ussd_session_start(  
    '79991234567', // phone.  
    'AF13BC57234', // optional. например, это id сессии, для идентификации диалога на стороне Вашего приложения  
    'GSM-7', // encoding. при этом значении Ваши скрипты обратного вызова должны отдавать меню в латинице  
    'my_ussd_app' // app_id. идентификатор USSD-приложения, задаётся Вами самостоятельно в ЛК  
) ;
```

## Возвращает

int/bool

Если старт сессии прошёл удачно, возвращает номер (`id`) сессии (по нумерации SMSimple), иначе `false` (например, неверный телефон или не установлено/не отвечает корректно скрипт обратного вызова).

```
function ussd_session_abort ( $ussd_session_id ) ¶
```

Прерывание USSD сессии. Оставлена для обратной совместимости, ничего реально не делает.

## Параметры

В качестве параметра `$ussd_session_id` сюда надо было передавать `id` сессии, полученный вызовом [`ussd\_session\_start`](#).

```
function sms_statistic ( $year_begin, $month_begin, $day_begin, $year_end, $month_end, $day_end, $count = false, $start = 0, $limit = 1000 ) ¶
```

Получение полной статистики SMS-сообщений за указанный период.

### Внимание!

Внимание! Использование этой функции временно отключено.

```
function get_balance ( ) ¶
```

Запрос баланса.

## Параметры

Без параметров.

## Возвращает

float

Возвращает только баланс, по конечному результату эквивалентно вызову `get_profile()['balance']`, но выгоднее, если Вам нужен только Ваш SMS-баланс, т.к. не передаёт ничего лишнего.

function **originOrder** ( \$new\_origin ) 

Заказ на создание новой подписи. Запрос на создание новой подписи проходит обязательную премодерацию менеджером и может быть одобрен или отклонён.

## Параметры

`$new_origin` string

какую подпись Вы желаете заказать, например `'vasyapupkin'`

## Возвращает

int

Возвратит номер заказа. Его можно использовать для проверки статуса заказа с помощью [originOrderStatus](#).

function **originOrderStatus** ( \$order\_id ) 

Проверка статуса заказа на создание новой подписи.

## Параметры

`$order_id` int

id заказа, полученный от метода [originOrder](#)

## Возвращает

int/null/string

Возвращает:

- `0` при отказе
- `null` при ожидании (возможен возврат пустой строки `''`)

- ненулевое число - `id` новой подписи, если заказ одобрен и подпись создана

```
function getAmount ( $message ) ¶
```

## Параметры

`$message` string  
Текст смс

## Возвращает

int Возвращённое число равно количеству отдельных SMS, которые необходимы для передачи всего текста на мобильный телефон абонента.

```
function getSymbolsCount ( $message ) ¶
```

## Параметры

`$message` string  
Текст смс

## Возвращает

int Возвращённое число равно количеству символов, которые будут задействованы при передаче SMS-сообщения. Логика работы примерно такая: если в тексте встречается хотя бы один символ не-латиницы, то все символы SMS будут занимать по 2 символа в конечном PDU, если только латиница - то по одному символу PDU на каждый символ исходного текста.

## Примечание

Под латиницей понимаются также цифры и знаки препинания.

```
function getPrice ( $origin_id, $phone, $message ) ¶
```

Функция совершает подсчёт символов в сообщении (аналогично [getSymbolsCount](#)), затем по ним считает количество частей (PDU), на которое, возможно, придётся разбить сообщение для отправки, затем выбирает наилучшую цену из Ваших пакетов и умножает цену одного PDU на их количество и на количество контактов, указанных в `$phone`.

## Параметры

`$origin_id` int

`id` подписи, `0` для случайно-цифровой

`$phone` string

номер абонента, можно использовать несколько номеров через запятую, например, `'89031234567, 79161234567, 89261324354'`

`$message` string

Текст сообщения

## Возвращает

array()

Возвращает массив вида

```
array(
    'cost' => 1.88, // суммарная стоимость всех SMS, с учётом количества абонентов
    // и количества частей сообщений
    'sms_count' => 2, // количество всех SMS = количество абонентов * количество частей сообщения
    'sms_out_of_balance' => 0 // если баланса достаточно для отправки всех этих сообщений, то
    // тут будет 0; в противном случае тут будет количество SMS,
    // на которых не хватит баланса; например, если баланс=0, то тут
    // будет такое же число, как и в 'sms_count', что значит, что ни
    // одно сообщение не может быть отправлено.
)
```

## Примечание

Цена SMS зависит от подписи.