

## Обновлено 31.05.2016

EXMO предоставляет доступ к API с помощью четырех способов.

### [Public API](#)

Не требует использования методов авторизации и доступен с помощью GET или POST запросов.

### [Authenticated API](#)

Требует использования авторизации и доступен только с использованием POST запросов.

### [Excode API](#)

Приватные функции доступны только с авторизацией. Для предоставления доступа к этим функциям необходимо обратиться в техподдержку.

### [Wallet API](#)

Требует использования авторизации и доступен только с использованием POST запросов.

## Public API

Этот API не требует использования авторизации и доступен с помощью GET или POST запроса.

В общем виде URL для доступа к API выглядит следующим образом [https://api.exmo.com/v1/{api\\_name}?{api\\_params}](https://api.exmo.com/v1/{api_name}?{api_params}) где api\_name - это название API к которому происходит обращение, а api\_params - входящие параметры запроса (если они необходимы)

### Описание методов:

Список сделок по валютной паре

Наименование метода:	<b>trades</b>
Тип запроса:	POST или GET
Входящие параметры:	<b>pair</b> - одна или несколько валютных пар разделенных запятой (пример BTC_USD,BTC_EUR)
Пример использования:	<a href="https://api.exmo.com/v1/trades/?pair=BTC_USD">https://api.exmo.com/v1/trades/?pair=BTC_USD</a>
Возвращаемое значение:	<pre>{   "BTC_USD": [     {       "trade_id": 3,       "type": "sell",       "price": "100",       "quantity": "1",       "amount": "100",       "date": 1435488248     }   ] }</pre>
Описание полей:	<b>trade_id</b> - идентификатор сделки <b>type</b> - тип сделки <b>price</b> - цена сделки <b>quantity</b> - кол-во по сделке <b>amount</b> - сумма сделки <b>date</b> - дата и время сделки в формате <a href="#">Unix</a>

Книга ордеров по валютной паре

Наименование метода:	<b>order_book</b>
Тип запроса:	POST или GET

Входящие параметры:	<b>pair</b> - одна или несколько валютных пар разделенных запятой (пример BTC_USD,BTC_EUR) <b>limit</b> - кол-во отображаемых позиций (по умолчанию 100, максимум 1000)
Пример использования:	<a href="https://api.exmo.com/v1/order_book/?pair=BTC_USD">https://api.exmo.com/v1/order_book/?pair=BTC_USD</a>
Возвращаемое значение:	<pre>{   "BTC_USD": {     "ask_quantity": "3",     "ask_amount": "500",     "ask_top": "100",     "bid_quantity": "1",     "bid_amount": "99",     "bid_top": "99",     "ask": [[100,1,100],[200,2,400]],     "bid": [[99,1,99]]   } }</pre>
Описание полей:	<b>ask_quantity</b> - объем всех ордеров на продажу <b>ask_amount</b> - сумма всех ордеров на продажу <b>ask_top</b> - минимальная цена продажи <b>bid_quantity</b> - объем всех ордеров на покупку <b>bid_amount</b> - сумма всех ордеров на покупку <b>bid_top</b> - максимальная цена покупки <b>bid</b> - список ордеров на покупку, где каждая строка это цена, количество и сумма <b>ask</b> - список ордеров на продажу, где каждая строка это цена, количество и сумма

Статистика цен и объемов торгов по валютным парам

Наименование метода:	<b>ticker</b>
Тип запроса:	POST или GET
Входящие параметры:	Отсутствуют
Пример использования:	<a href="https://api.exmo.com/v1/ticker/">https://api.exmo.com/v1/ticker/</a>
Возвращаемое значение:	<pre>{   "BTC_USD": {     "buy_price": "589.06",     "sell_price": "592",     "last_trade": "591.221",     "high": "602.082",     "low": "584.51011695",     "avg": "591.14698808",     "vol": "167.59763535",     "vol_curr": "99095.17162071",     "updated": "1470250973"   } }</pre>
Описание полей:	<b>high</b> - максимальная цена сделки за 24 часа <b>low</b> - минимальная цена сделки за 24 часа <b>avg</b> - средняя цена сделки за 24 часа <b>vol</b> - объем всех сделок за 24 часа <b>vol_curr</b> - сумма всех сделок за 24 часа <b>last_trade</b> - цена последней сделки <b>buy_price</b> - текущая максимальная цена покупки <b>sell_price</b> - текущая минимальная цена продажи <b>updated</b> - дата и время обновления данных

Настройки валютных пар

Наименование метода:	<b>pair_settings</b>
Тип запроса:	POST или GET

Входящие параметры:	Отсутствуют
Пример использования:	<a href="https://api.exmo.com/v1/pair_settings/">https://api.exmo.com/v1/pair_settings/</a>
Возвращаемое значение:	<pre>{   "BTC_USD": {     "min_quantity": "0.001",     "max_quantity": "100",     "min_price": "1",     "max_price": "10000",     "max_amount": "30000",     "min_amount": "1"   } }</pre>
Описание полей:	<b>min_quantity</b> - минимальное кол-во по ордеру <b>max_quantity</b> - максимальное кол-во по ордеру <b>min_price</b> - минимальная цена по ордеру <b>max_price</b> - максимальная цена по ордеру <b>min_amount</b> - минимальная сумма по ордеру <b>max_amount</b> - максимальная сумма по ордеру

Список валют биржи

Наименование метода:	<b>currency</b>
Тип запроса:	POST или GET
Входящие параметры:	Отсутствуют
Пример использования:	<a href="https://api.exmo.com/v1/currency/">https://api.exmo.com/v1/currency/</a>
Возвращаемое значение:	<code>[ "USD" , "EUR" , "RUB" , "BTC" , "DOGE" , "LTC" ]</code>

# Authenticated API

Для доступа к данному API требуется авторизация и необходимо использовать POST метод.

**URL** — необходимо использовать следующий адрес [https://api.exmo.com/v1/{api\\_name}](https://api.exmo.com/v1/{api_name}) где api\_name - это наименование API метода

Авторизация осуществляется с помощью отправки на сервер следующих заголовков:

**Key** — Публичный ключ, его нужно взять в настройках профиля пользователя  
(пример: K-7cc97c89aed2a2fd9ed7792d48d63f65800c447b)

**Sign** — POST данные (param=val&param1=val1), подписанные секретным ключом методом HMAC-SHA512, секретный ключ также нужно брать в настройках профиля пользователя

Существует возможность привязать несколько ключей к одному аккаунту для доступа к API. **Для этого обращайтесь в техподдержку.**

Дополнительно во всех запросах должен находиться обязательный POST-параметр nonce с инкрементным числовым значением (>0). **Это значение не должно повторяться или уменьшаться.**

Для доступа к API можно использовать готовый программный код на следующих языках:

PHP	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/php">github.com/exmo-dev/exmo_api_lib/tree/master/php</a> или <code>composer require exmo/api</code>
Javascript	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/js">github.com/exmo-dev/exmo_api_lib/tree/master/js</a>
Nodejs	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/nodejs">github.com/exmo-dev/exmo_api_lib/tree/master/nodejs</a> или <code>npm install exmo-api</code>
C#	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/c#">github.com/exmo-dev/exmo_api_lib/tree/master/c#</a>
C++	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/c++">github.com/exmo-dev/exmo_api_lib/tree/master/c++</a>

Python	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/python">github.com/exmo-dev/exmo_api_lib/tree/master/python</a>
Objective C	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/objectivec">github.com/exmo-dev/exmo_api_lib/tree/master/objectivec</a>
Swift	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/swift">github.com/exmo-dev/exmo_api_lib/tree/master/swift</a>
Java	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/java">github.com/exmo-dev/exmo_api_lib/tree/master/java</a>
Ruby	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/ruby">github.com/exmo-dev/exmo_api_lib/tree/master/ruby</a>
Golang	<a href="https://github.com/exmo-dev/exmo_api_lib/tree/master/golang">github.com/exmo-dev/exmo_api_lib/tree/master/golang</a>

### Описание методов:

Получение информации об аккаунте пользователя

Наименование метода:	<b>user_info</b>
Тип запроса:	POST
Входящие параметры:	Отсутствуют
Пример использования:	<b>api_query("user_info", Array());</b>
Возвращаемое значение:	<pre>{   "uid": 10542,   "server_date": 1435518576,   "balances": {     "BTC": "970.994",     "USD": "949.47"   },   "reserved": {     "BTC": "3",     "USD": "0.5"   } }</pre>
Описание полей:	<b>uid</b> - идентификатор пользователя <b>server_date</b> - дата и время сервера <b>balances</b> - доступный баланс пользователя <b>reserved</b> - баланс пользователя в ордерах

Создание ордера

Наименование метода:	<b>order_create</b>
Тип запроса:	POST
Входящие параметры:	<b>pair</b> - валютная пара <b>quantity</b> - кол-во по ордеру <b>price</b> - цена по ордеру <b>type</b> - тип ордера, может принимать следующие значения: <ul style="list-style-type: none"><li>• buy - ордер на покупку</li><li>• sell - ордер на продажу</li><li>• market_buy - ордера на покупку по рынку</li><li>• market_sell - ордер на продажу по рынку</li><li>• market_buy_total - ордер на покупку по рынку на определенную сумму</li><li>• market_sell_total - ордер на продажу по рынку на определенную сумму</li></ul>

Возвращаемое значение:	<pre>{   "result": true,   "error": "",   "order_id": 123456 }</pre>
Описание полей:	<p><b>result</b> - true в случае успешного создания и false в случае ошибки</p> <p><b>error</b> - содержит текст ошибки</p> <p><b>order_id</b> - идентификатор ордера</p>

Примеры создания ордеров на PHP:

Создание ордера на покупку 3 BTC по цене 100 USD

```
api_query("order_create", Array(
  "pair"=>"BTC_USD",
  "quantity"=>3,
  "price"=>100,
  "type"=>"buy"
));
```

Создание ордера на продажу 3 BTC по цене 100 USD

```
api_query("order_create", Array(
  "pair"=>"BTC_USD",
  "quantity"=>3,
  "price"=>100,
  "type"=>"sell"
));
```

Создание ордера на покупку по рынку 3 BTC

```
api_query("order_create", Array(
  "pair"=>"BTC_USD",
  "quantity"=>3,
  "price"=>0,
  "type"=>"market_buy"
));
```

Создание ордера на продажу по рынку 3 BTC

```
api_query("order_create", Array(
  "pair"=>"BTC_USD",
  "quantity"=>3,
  "price"=>0,
  "type"=>"market_sell"
));
```

Создание ордера на покупку BTC по рынку на сумму 100 USD

```
api_query("order_create", Array(
  "pair"=>"BTC_USD",
  "quantity"=>100,
  "price"=>0,
  "type"=>"market_buy_total"
));
```

Создание ордера на продажу BTC по рынку на сумму 100 USD

```
api_query("order_create", Array(
  "pair"=>"BTC_USD",
  "quantity"=>100,
  "price"=>0,
  "type"=>"market_sell_total"
));
```

Отмена ордера

Наименование метода:	order_cancel
Тип запроса:	POST

Входящие параметры:	<b>order_id</b> - идентификатор ордера
Пример использования:	<pre>api_query("order_cancel", Array(   "order_id"=&gt;104235 ));</pre>
Возвращаемое значение:	<pre>{   "result": true,   "error": "" }</pre>
Описание полей:	<b>result</b> - true в случае успешного создания задачи на отмену ордера и false в случае ошибки <b>error</b> - содержит текст ошибки

Получение списка открытых ордеров пользователя

Наименование метода:	<b>user_open_orders</b>
Тип запроса:	POST
Входящие параметры:	отсутствуют
Пример использования:	<pre>api_query("user_open_orders", Array());</pre>
Возвращаемое значение:	<pre>{   "BTC_USD": [     {       "order_id": "14",       "created": "1435517311",       "type": "buy",       "pair": "BTC_USD",       "price": "100",       "quantity": "1",       "amount": "100"     }   ] }</pre>
Описание полей:	<b>order_id</b> - идентификатор ордера <b>created</b> - дата и время создания ордера <b>type</b> - тип ордера <b>pair</b> - валютная пара <b>price</b> - цена по ордеру <b>quantity</b> - кол-во по ордеру <b>amount</b> - сумма по ордеру

Получение сделок пользователя

Наименование метода:	<b>user_trades</b>
Тип запроса:	POST
Входящие параметры:	<b>pair</b> - одна или несколько валютных пар разделенных запятой (пример BTC_USD,BTC_EUR) <b>offset</b> - смещение от последней сделки (по умолчанию 0) <b>limit</b> - кол-во возвращаемых сделок (по умолчанию 100, максимум 10 000)
Пример использования:	<pre>api_query("user_trades", Array(   "pair"=&gt;"BTC_USD",   "limit"=&gt;100,   "offset"=&gt;0 ));</pre>

Возвращаемое значение:	<pre>{   "BTC_USD": [     {       "trade_id": 3,       "date": 1435488248,       "type": "buy",       "pair": "BTC_USD",       "order_id": 7,       "quantity": 1,       "price": 100,       "amount": 100     }   ] }</pre>
Описание полей:	<p><b>trade_id</b> - идентификатор сделки <b>date</b> - дата и время сделки <b>type</b> - тип сделки <b>pair</b> - валютная пара <b>order_id</b> - идентификатор ордера пользователя <b>quantity</b> - кол-во по сделке <b>price</b> - цена сделки <b>amount</b> - сумма сделки</p>

Получение отмененных ордеров пользователя

Наименование метода:	<b>user_cancelled_orders</b>
Тип запроса:	POST
Входящие параметры:	<p><b>offset</b> - смещение от последней сделки (по умолчанию 0) <b>limit</b> - кол-во возвращаемых сделок (по умолчанию 100, максимум 10 000)</p>
Пример использования:	<pre>api_query("user_cancelled_orders", Array(   "limit"&gt;100,   "offset"&gt;0 ));</pre>
Возвращаемое значение:	<pre>[   {     "date": 1435519742,     "order_id": 15,     "order_type": "sell",     "pair": "BTC_USD",     "price": 100,     "quantity": 3,     "amount": 300   } ]</pre>
Описание полей:	<p><b>date</b> - дата и время отмены ордера <b>order_id</b> - идентификатор ордера <b>order_type</b> - тип ордера <b>pair</b> - валютная пара <b>price</b> - цена по ордеру <b>quantity</b> - кол-во по ордеру <b>amount</b> - сумма по ордеру</p>

Получение истории сделок ордера

Наименование метода:	<b>order_trades</b>
Тип запроса:	POST
Входящие параметры:	<b>order_id</b> - идентификатор ордера

Пример использования:	<pre>api_query("order_trades", Array(   "order_id"=&gt;12345 ));</pre>
Возвращаемое значение:	<pre>{   "type": "buy",   "in_currency": "BTC",   "in_amount": "1",   "out_currency": "USD",   "out_amount": "100",   "trades": [     {       "trade_id": 3,       "date": 1435488248,       "type": "buy",       "pair": "BTC_USD",       "order_id": 12345,       "quantity": 1,       "price": 100,       "amount": 100     }   ] }</pre>
Описание полей:	<p><b>type</b> - тип ордера <b>in_currency</b> - валюта входящая <b>in_amount</b> - кол-во входящей валюты <b>out_currency</b> - валюта исходящая <b>out_amount</b> - кол-во исходящей валюты <b>trades</b> - массив сделок, где:</p> <ul style="list-style-type: none"><li>• trade_id - идентификатор сделки</li><li>• date - дата сделки</li><li>• type - тип сделки</li><li>• pair - валютная пара</li><li>• order_id - идентификатор ордера</li><li>• quantity - кол-во по сделке</li><li>• price - цена по сделке</li><li>• amount - сумма по сделке</li></ul>

Подсчет в какую сумму обойдется покупка определенного кол-ва валюты по конкретной валютной паре

Наименование метода:	<b>required_amount</b>
Тип запроса:	POST
Входящие параметры:	<b>pair</b> - валютная пара <b>quantity</b> - кол-во которое необходимо купить
Пример использования:	<pre>api_query("required_amount", Array(   "pair"=&gt;"BTC_USD",   "quantity"=&gt;"11" ));</pre>
Возвращаемое значение:	<pre>{   "quantity": 3,   "amount": 5,   "avg_price": 3.66666666 }</pre>
Описание полей:	<p><b>quantity</b> - кол-во которое вы сможете купить <b>amount</b> - сумма на которую вы потратите на покупку <b>avg_price</b> - средняя цена покупки</p>

Получение списка адресов для депозита криптовалют

Наименование метода:	<b>deposit_address</b>
Тип запроса:	POST



Входящие параметры:	отсутствуют
Пример использования:	<code>api_query("deposit_address", Array());</code>
Возвращаемое значение:	<pre>{   "BTC": "16UM5DoeHkV7Eb7tMfXSuQ2ueir1yj4P7d",   "DOGE": "DEVfhgKErG5Nzas2FZJJH8Y8pjoLFVfWq4",   "LTC": "LSJFhsVJM6GCFtSgRj5hHuK9gReLhNuKFb",   "XRP": "rB2yjjyFCoJaV8QCbjlUJzMnUnQJMrkhv3S,1234" }</pre>

Создание задачи на вывод криптовалют. **ВНИМАНИЕ!!!** Эта API функция включается пользователю после запроса в техподдержку

Наименование метода:	<b>withdraw_crypt</b>
Тип запроса:	POST
Входящие параметры:	<b>amount</b> - кол-во выводимой валюты <b>currency</b> - наименование выводимой валюты <b>address</b> - адрес вывода <b>invoice</b> - дополнительный идентификатор (обязательно для XRP)
Пример использования:	<code>api_query("withdraw_crypt", Array(   "amount"=&gt;10,   "currency"=&gt;"BTC",   "address"=&gt;"16UM5DoeHkV7Eb7tMfXSu...",   "invoice"=&gt;"1234" ));</code>
Возвращаемое значение:	<pre>{   "result": true,   "error": "",   "task_id": "467756" }</pre>
Описание полей:	<b>result</b> - true в случае успешного создания задачи на вывод, и false в случае ошибки <b>error</b> - содержит описание ошибки <b>task_id</b> - идентификатор задачи на вывод

Получение ИД транзакции криптовалюты для отслеживания на blockchain

Наименование метода:	<b>withdraw_get_txid</b>
Тип запроса:	POST
Входящие параметры:	<b>task_id</b> - идентификатор задания на вывод
Пример использования:	<code>api_query("withdraw_get_txid", Array(   "task_id"=&gt;467756 ));</code>
Возвращаемое значение:	<pre>{   "result": true,   "error": "",   "status": true,   "txid": "ec46f784ad976fd7f7539089d1a129fe46..." }</pre>
Описание полей:	<b>result</b> - true в случае успешного создания задачи на вывод, и false в случае ошибки <b>error</b> - содержит описание ошибки <b>status</b> - true если вывод уже осуществлен <b>txid</b> - идентификатор транзакции по которому можно её найти в blockchain

# EXCODE API

Используя EXCODE API можно создавать и загружать купоны EXCODE. [Доступ предоставляется после обращения в техподдержку.](#)

## Создание купона EXCODE

Наименование метода:	<b>excode_create</b>
Тип запроса:	POST
Входящие параметры:	<b>currency</b> - наименование валюты купона <b>amount</b> - сумма на которую создается купон
Пример использования	<pre>api_query("excode_create", Array(     "currency"=&gt;"BTC",     "amount"=&gt;10 ));</pre>
Пример ответа:	<pre>{   "result": true,   "error": "",   "task_id": "467757",   "code": "EX-CODE_9004_BTC7c3f8adc0b158658...",   "amount": "10",   "currency": "BTC",   "balances": {     "BTC": 940.994,     "USD": 949.472   } }</pre>
Описание полей:	<b>result</b> - true в случае успешного создания купона, и false в случае ошибки <b>error</b> - содержит описание ошибки <b>task_id</b> - идентификатор купона <b>code</b> - код EXCODE <b>amount</b> - сумма купона <b>currency</b> - валюта купона <b>balances</b> - баланс пользователя после создания купона

## Загрузка купона EXCODE

Наименование метода:	<b>excode_load</b>
Входящие параметры:	<b>code</b> - код купона EXCODE
Пример использования	<pre>api_query("excode_load", Array(     "code"=&gt;"EX-CODE_9004_BTC7c3f8adc0b158658..." ));</pre>
Возвращаемое значение:	<pre>{   "result": true,   "error": "",   "task_id": "467757",   "amount": "10",   "currency": "BTC",   "balances": {     "BTC": 950.994,     "USD": 949.472   } }</pre>
Описание полей:	<b>result</b> - true в случае успешной загрузки купона, и false в случае ошибки <b>error</b> - содержит описание ошибки <b>task_id</b> - идентификатор купона

**amount** - сумма купона  
**currency** - валюта купона  
**balances** - баланс пользователя после загрузки купона

## WALLET API

Этот API вызывается аналогично Authenticated API.

Количество API вызовов ограничено 10 запросами в минуту с одного IP адреса.

Получение истории wallet

Наименование метода:	<b>wallet_history</b>
Тип запроса:	POST
Входящие параметры:	<b>date</b> - дата timestamp за которую нужно получить историю (если не указан берется текущий день)
Пример использования	<pre>api_query("wallet_history", Array(   "date"&gt;1493998000 ));</pre>
Пример ответа:	<pre>{   "result": true,   "error": "",   "begin": "1493942400",   "end": "1494028800",   "history": [{     "dt": 1461841192,     "type": "deposit",     "curr": "RUB",     "status": "processing",     "provider": "Qivi (IA) [12345]",     "amount": "1",     "account": "",   },   {     "dt": 1463414785,     "type": "withdrawal",     "curr": "USD",     "status": "paid",     "provider": "EXCODE",     "amount": "-1",     "account": "EX-CODE_19371_USDda...",   } ] }</pre>
Описание полей:	<p><b>result</b> - true в случае успешного получения истории, и false в случае ошибки</p> <p><b>error</b> - содержит описание ошибки</p> <p><b>begin</b> - начало периода</p> <p><b>end</b> - конец периода</p> <p><b>history</b> - массив операций пользователя (история кошелька), где</p> <p><b>dt</b> - дата операции</p> <p><b>type</b> - тип</p> <p><b>curr</b> - валюта</p> <p><b>status</b> - статус</p> <p><b>provider</b> - провайдер</p> <p><b>amount</b> - сумма</p> <p><b>account</b> - счет</p>

## Ограничение количества запросов

Количество API вызовов ограничено 180 запросами в минуту с одного IP адреса либо от одного пользователя.

# ПОДПИСЫВАЙТЕСЬ НА НАС



Telegram  
25,468 Подписчиков



Twitter  
12,845 Подписчиков



Facebook  
4,492 Подписчиков



VK  
64,439 Подписчиков

# ПРОДУКТЫ

Кошелек

Торги

API

EXMO coin

# ИНФОРМАЦИЯ

Поддержка пользователей

EXMOWIKI

Комиссии и лимиты

Реферальная программа

# КОМПАНИЯ

Контакты

Новости

Работа в компании

О нас

# ДОКУМЕНТЫ

Пользовательское  
соглашение

Политика AML

Политика  
конфиденциальности

Политика cross-chain