

# API Яндекс.Денег

Руководство разработчика

19.04.2011

**Я**ндекс

API Яндекс.Денег. Руководство разработчика. Версия 1.0

Дата сборки документа: 19.04.2011.

Этот документ является составной частью технической документации Яндекса.

Сайт справки к сервисам Яндекса: <http://help.yandex.ru>

© 2008—2011 ООО «ЯНДЕКС». Все права защищены.

## **Предупреждение об исключительных правах**

Яндексу (а также указанному им правообладателю) принадлежат исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации, используемые при разработке, поддержке и эксплуатации сервиса API Яндекс.Денег. К таким результатам могут относиться, но не ограничиваясь указанными, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования. Эти права охраняются в соответствии с Гражданским кодексом РФ и международным правом.

Вы можете использовать сервис API Яндекс.Денег или его составные части только в рамках полномочий, предоставленных вам Пользовательским соглашением сервиса API Яндекс.Денег или специального соглашения.

Нарушение требований по защите исключительных прав правообладателя влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с российским законодательством.

## **Контактная информация**

ООО «ЯНДЕКС»

<http://www.yandex.ru>

Тел.: +7 495 739 7000

Email: [pr@yandex-team.ru](mailto:pr@yandex-team.ru)

Главный офис: 119021, Россия, г. Москва, ул. Льва Толстого, д. 16

# Содержание

Введение .....	4
<b>Авторизация приложения</b> .....	5
Регистрация приложения .....	6
Получение токена .....	7
Описание протокола .....	12
<b>Получение информации о счете пользователя</b> .....	16
Метод account-info .....	16
Метод operation-history .....	16
Метод operation-details .....	19
<b>Проведение платежей</b> .....	21
Метод request-payment .....	21
Метод process-payment .....	23
Предметный указатель .....	25

## Введение

Руководство предназначено для разработчиков сервисов и приложений, использующих API Яндекс.Денег.

## Авторизация приложения

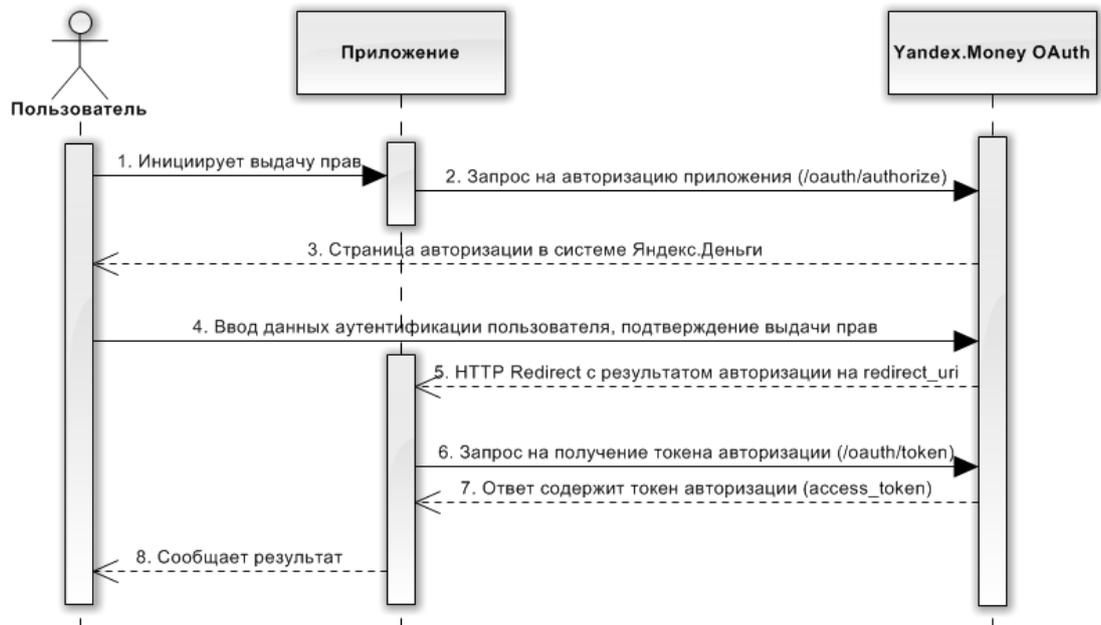
Чтобы ваше клиентское приложение могло работать с личным кошельком пользователя на сервисе Яндекс.Деньги, ему необходимо пройти авторизацию.

Протокол OAuth позволяет сделать авторизацию безопасной и удобной. При OAuth-авторизации приложению не обязательно запрашивать у пользователя его логин и пароль на Яндексе. Вместо этого можно предложить пользователю дать разрешение на доступ к своим данным непосредственно на сайте Яндекс.Денег.

Протокол авторизации, используемый в API Яндекс.Денег является реализацией следующих спецификаций:

- [The OAuth 2.0 Authorization Protocol](#)
- [The OAuth 2.0 Protocol: Bearer Tokens](#)

Упрощенная схема взаимодействия приложения с OAuth-сервером Яндекс.Денег представлена на диаграмме:



### Шаги, выполняемые разработчиком приложения

1. Разработчик регистрирует свое приложение в системе Яндекс.Деньги.

С точки зрения протокола OAuth2 — это фаза регистрации приложения (**Registration Request**). Система Яндекс.Деньги выдаст разработчику приложения `client_id` — идентификатор приложения типа `string`.

См. раздел [Регистрация приложения](#).

2. Разработчик встраивает полученный `client_id` в код приложения, объявляя его константой. Далее приложение распространяется любым удобным способом. В течение жизненного цикла приложения `client_id` не изменяется.

См. раздел [Получение токена](#).

## Шаги, выполняемые пользователем приложения

1. В приложении пользователь инициирует процесс авторизации приложения для управления своим счетом.
2. Приложение отправляет запрос **Authorization Request** на сервер системы Яндекс.Деньги.  
С точки зрения протокола OAuth2 — это фаза авторизации приложения (**Authorization Request**).
3. Сервер Яндекс.Денег перенаправляет пользователя на страницу авторизации.
4. Пользователь аутентифицируется, просматривает список запрашиваемых прав, и, подтверждает, либо отклоняет авторизацию приложения.
5. Приложению возвращается ответ в виде **HTTP Redirect**, либо со временным токеном для получения доступа, либо с кодом ошибки, см. раздел **Authorization Response** в описании стандарта OAuth2.
6. Приложение, используя временный токен полученный на предыдущем шаге, отправляет серверу Яндекс.Денег запрос на получение токена авторизации (**Access Token Request**).
7. Ответ сервера содержит токен авторизации `access_token`.
8. Приложение отображает пользователю результат авторизации.

## Требования к обеспечению безопасности

1. Все сетевые взаимодействия должны производиться **только** по **HTTPS**.
2. Приложение должно проверять корректность SSL-сертификата сервера и немедленно прекращать сессию в случае неуспеха проверки, чтобы не допустить утечку данных авторизации.
3. Рекомендуется проверять подлинность сервера Яндекс.Деньги путем валидации сертификата сервера по цепочке:
 

```
YandexExternalCA
ООО Яндекс
```
4. Не храните токен авторизации в открытом виде, в том числе в виде **cookie**.
5. **Никогда** не используйте токен авторизации в параметрах запросов (GET, POST, и пр.).

## Регистрация приложения

Для регистрации приложения требуется передать сведения о нем в систему Яндекс.Деньги. Для этого выполните следующие шаги:

1. Перейдите по ссылке <https://sp-money.yandex.ru/myservices/new.xml> на страницу **Регистрация сервиса**.
2. Укажите свойства приложения:

### **description**

Описание приложения, например "Мобильный баланс".

### **redirect\_uri**

URI для передачи события аутентификации (см. описание `redirect_uri` в стандарте [The OAuth 2.0 Authorization Protocol](#)).

### **application\_uri**

Ссылка на веб-страницу приложения или сайт разработчика.

3. Нажмите кнопку **Подтвердить**.

Откроется страница с информацией о правах доступа приложения к сервису Яндекс.Деньги.

**Опасно!**

Разработчик приложения не должен открыто публиковать где-либо свой `client_id`, поскольку это упростит "фишинговые атаки" (выпуск приложений или сайтов, получающих токены от имени вашего приложения). В случае утечки `client_id`, система Яндекс.Деньги будет считать что получает запросы от приложения, зарегистрированного с данным `client_id`.

**См. также**

[Авторизация приложения](#)

## Получение токена

Для получения токена приложение должно отправить запрос `Authorization Request` на сервер системы Яндекс.Деньги. Приложение должно открыть браузер операционной системы и отправить HTTP-запрос с его помощью.

**Совет:**

Для отправки запроса рекомендуется использовать метод `POST` — эквивалент `HTML form submit`.

Формат запроса:

```
POST /oauth/authorize HTTP/1.1
Host: m.sp-money.yandex.ru (для мобильных устройств) или sp-money.yandex.ru (для остальных устройств)
Content-Type: application/x-www-form-urlencoded
Content-Length: <content-length>

client_id=<client_id>&response_type=code
&redirect_uri=<redirect_uri>&scope=<scope>
```

Например:

```
POST /oauth/authorize HTTP/1.1
Host: sp-money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345

client_id=092763469236489593523464667&response_type=code
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
&scope=account%2Dinfo%20payment%2Dshop
```

Параметры запроса:

Параметр	Тип	Описание
<code>client_id</code>	string	Идентификатор приложения, полученный при <a href="#">регистрации</a> .
<code>response_type</code>	string	Фиксированное значение: <code>code</code> .
<code>redirect_uri</code>	string	URI, на который OAuth-сервер осуществляет передачу события результата авторизации.  Значение этого параметра при символьном сравнении должно быть идентично значению <a href="#">redirect_uri</a> , указанному при регистрации приложения. При сравнении не учитываются индивидуальные параметры

		приложения, которые могут быть добавлены в конец строки URI.
scope	string	Список запрашиваемых прав. В качестве разделителя элементов списка используется пробел.

**Примечание:**

Не допускается отправка запроса (открытие страницы) из приложения, поскольку правилами платежной системы не разрешается ввод секретов аутентификации пользователя где-либо, кроме сайта платежной системы Яндекс.Деньги.

По запросу авторизации, пользователь перенаправляется на страницы платежной авторизации системы Яндекс.Деньги, где он авторизуется сам и подтверждает, либо отклоняет запрос авторизации приложения. Также пользователь видит список предоставляемых приложению прав и лимиты платежа.

По результату действий пользователя на страницах системы Яндекс.Деньги, система отвечает HTTP-перенаправлением с результатом авторизации (HTTP 302 Redirect). Ваше приложение должно уметь обрабатывать событие HTTP Redirect.

**Внимание!**

В случае повторного вызова авторизации приложения пользователем с тем же значением параметра `client_id`, выданные ранее разрешения аннулируются.

Параметры перенаправления с результатом авторизации:

Параметр	Тип	Описание
code	string	Временный токен (authorization code), подлежащий обмену на токен авторизации. Присутствует в случае успешного подтверждения авторизации пользователем.
error	string	Код ошибки в случае ошибки или отказа в авторизации пользователем. Присутствует в случае отказа в авторизации.

В случае ошибки OAuth-сервер передает параметр

**error**

Код ошибки:

- `invalid_request` — в запросе отсутствуют обязательные параметры, либо параметры имеют некорректные или недопустимые значения.
- `unauthorized_client` — неверное значение параметра `client_id`, либо приложение не имеет право запрашивать авторизацию (например, данный `client_id` заблокирован системой Яндекс.Деньги).
- `access_denied` — пользователь отклонил запрос авторизации приложения.

Ответ системы Яндекс.Деньги при успешной авторизации:

```
HTTP/1.1 302 Found
Location: https://client.example.com/cb?code=i1WsRnluB1ehfbb37
```

Ответ системы Яндекс.Деньги при отказе в авторизации:

```
HTTP/1.1 302 Found
Location: https://client.example.com/cb?error=access_denied
```

**Примечание:**

Временный токен (значение поля `code` ответа сервера системы) подлежит немедленному обмену на токен авторизации. Время действия этого токена мало (менее 1 минуты).

Приложение должно получить и обработать ответ сервера системы Яндекс.Деньги и немедленно самостоятельно обменять временный токен на токен авторизации (см. ниже).

В случае невозможности получения ответа от сервера системы, либо утере, либо истечении времени действия временного токена, приложение должно повторить процедуру авторизации приложения пользователем.

## Получение токена авторизации

При успешном завершении фазы авторизации приложения пользователем, приложение должно немедленно обменять временный токен на токен авторизации.

Приложение должно отправить запрос, содержащий временный токен на OAuth-сервер системы. Для отправки запроса используйте метод POST:

```
POST /oauth/token HTTP/1.1
Host: m.sp-money.yandex.ru (для мобильных устройств) или sp-money.yandex.ru (для остальных устройств)
Content-Type: application/x-www-form-urlencoded
Content-Length: <content-length>

code=<code>&client_id=<client_id>&grant_type=authorization_code&redirect_uri=<redirect_uri>
```

Например:

```
POST /oauth/token HTTP/1.1
Host: sp-money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345

code=i1WsRn1uB1ehfbb37&client_id=092763469236489593523464667&grant_type=authorization_code&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
```

Параметры запроса:

Параметр	Тип	Описание
<code>code</code>	string	Временный токен ( <b>authorization code</b> ).
<code>client_id</code>	string	Идентификатор приложения, полученный при <a href="#">регистрации</a> .
<code>grant_type</code>	string	Фиксированное значение: <code>authorization_code</code> .
<code>redirect_uri</code>	string	URI, на который OAuth-сервер осуществляет передачу события результата авторизации.  Значение этого параметра при посимвольном сравнении должно быть идентично значению <code>redirect_uri</code> , ранее переданному в запросе <a href="#">authorize</a> .

В ответ на запрос, сервер системы Яндекс.Деньги возвращает токен авторизации (`access token`), который является симметричным секретом приложения, дающим право проводить операции со счетом пользователя.

Токен возвращается в виде JSON-документа, который (в зависимости от результата обмена) может содержать одно из следующих полей:

Параметр	Тип	Описание
access_token	string	Токен авторизации. Присутствует в случае успешного обмена временного токена.
error	string	Код ошибки. Присутствует в случае отказа в обмене временного токена.

В случае ошибки OAuth-сервер передает параметр

#### error

Код ошибки:

- `invalid_request` — в запросе отсутствуют обязательные параметры, либо параметры имеют некорректные или недопустимые значения.
- `unauthorized_client` — неверное значение параметра `client_id`, либо приложение не имеет право запрашивать авторизацию (например, данный `client_id` заблокирован системой Яндекс.Деньги).
- `access_denied` — пользователь отклонил запрос авторизации приложения.

Например, ответ системы Яндекс.Деньги при успешном обмене временного токена:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Cache-Control: no-store

{
  "access_token": "SlAV32hkhjhkkefhkjGHGSDcbndbjhfSHDFhjd bfvbdsj fhbvf sjhjdfbvdfhj hj
  khkwfervnrjKG"
}
```

и при ошибке:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 12345
Cache-Control: no-store

{
  "error": "access_denied"
}
```

#### Совет:

Следует учитывать что временный токен является одноразовым. В случае невозможности получения ответа от сервера за время жизни токена, потребуется повторить процедуру авторизации приложения пользователем.

---

**Примечание:**

Полученный `access_token` является симметричным секретом авторизации, поэтому разработчик приложения должен предпринять меры по его защите: хранить авторизационный токен в зашифрованном виде, предоставлять доступ к нему только после авторизации пользователя в приложении. Например, токен авторизации может быть зашифрован с помощью алгоритма 3DES, где ключ шифрования — 4-х значный ПИН-код.

---

**См. также**[Авторизация приложения](#)[Регистрация приложения](#)

## Описание протокола

### Формат запроса

Запросы осуществляются посредством протокола HTTP 1.1 с использованием SSL (HTTPS), на адрес:

```
https://money.yandex.ru/api/<имя_метода>
```

Авторизация запросов осуществляется согласно стандарту OAuth2-Bearer. HTTP-запросы должны содержать следующий заголовок:

```
Authorization: Bearer <access_token>
```

---

#### Примечание:

Указанный токен должен иметь права на исполнение запрашиваемого метода с заданным набором параметров.

---

Требования к обеспечению безопасности:

1. Все сетевые взаимодействия должны производиться **только** по **HTTPS**.
2. Приложение должно проверять корректность SSL-сертификата сервера и немедленно прекращать сессию в случае неуспеха проверки, чтобы не допустить утечку данных авторизации.
3. Рекомендуется проверять подлинность сервера Яндекс.Деньги путем валидации сертификата сервера по цепочке:

```
YandexExternalCA  
ООО Yandex
```

4. Не храните токен авторизации в открытом виде, в том числе в виде **cookie**.
5. **Никогда** не используйте токен авторизации в параметрах запросов (GET, POST, и пр.).

Для передачи аргументов запроса используется следующий формат:

- каждый аргумент указывается парой ключ/значение в виде параметра POST-запроса;
- MIME-тип: `application/x-www-form-urlencoded`;
- кодировка символов: UTF-8.

Пример запроса:

```
POST /api/request-payment HTTP/1.1  
Host: money.yandex.ru  
Content-Type: application/x-www-form-urlencoded  
Authorization: Bearer 01234567890ABCDEF01234567890  
  
param1=value1&param2=value2&param3=value3
```

### Формат ответа

Ответ сервиса представляет собой JSON-документ в кодировке UTF-8, см. [The application/json Media Type for JavaScript Object Notation \(JSON\)](#) и [официальный сайт JSON](#). Содержимое документа зависит от результата выполнения запроса.

Пример ответа в случае успешного выполнения:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Cache-Control: no-cache

{
  "param1": "value1",
  "param2": "value2"
}

```

В ответе платежных методов присутствуют HTTP-заголовки, запрещающие прокси-серверам и локальным браузерам кэшировать содержимое.

#### Примечание:

В ответе могут также присутствовать поля, не описанные в настоящем протоколе. Приложение должно их игнорировать.

При отказе в авторизации сервер отвечает HTTP кодом 4xx (см. таблицу ниже). Отказ может быть получен, если:

- в запросе отсутствует HTTP- заголовок `Authorization`;
- в заголовке `Authorization` указан несуществующий, некорректный или просроченный токен;
- запрошена операция, на которую у токена отсутствуют необходимые права.

Ответ содержит заголовок `WWW-Authenticate` (согласно стандарту [OAuth2-Bearer](#)) в кодировке UTF-8. В ответе могут присутствовать поля, не описанные в данном протоколе.

Поля, содержащиеся в ответе при отказе в авторизации:

Параметр	Описание
<code>error</code>	Код причины отказа в авторизации.
<code>error_description</code>	Дополнительное текстовое описание причины отказа, может отсутствовать.

Возможные коды причины отказа в авторизации:

HTTP-код ответа	Значение поля <code>error</code>	Описание
400	<code>invalid_request</code>	Формат HTTP запроса не соответствует протоколу. Запрос невозможно разобрать, либо отсутствует заголовок <code>Authorization</code> , либо он имеет некорректное значение.
401	<code>invalid_token</code>	Указан несуществующий токен, либо токен просрочен или отозван.
403	<code>insufficient_scope</code>	Запрошена операция, на которую у токена отсутствуют необходимые права.

Пример ответа при отсутствующем заголовке:

```

HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer error="invalid_request"

```

Пример ответа при просроченном токене:

```

HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="The access token expired"

```

Пример ответа в случае отсутствия у токена необходимых прав:

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_scope", error_description="Платеж
запрещен параметрами авторизации приложения"
```

В случае общей технической ошибки сервер возвращает HTTP-код 500 Internal Server Error. Приложение должно повторить запрос через некоторое время с теми же параметрами.

## Права на выполнение операций

Вызов операций должен сопровождаться передачей OAuth-токена авторизации приложения. Токен авторизации должен обладать соответствующими правами. Права запрашиваются в параметре `scope` при вызове `authorize` по протоколу OAuth2.

Список возможных прав:

Название права	Описание
account-info	Получение информации о состоянии счета, см. <a href="#">метод account-info</a> .
operation-history	Просмотр истории операций, см. <a href="#">метод operation-history</a> .
operation-details	Просмотр деталей операции из истории операций, см. <a href="#">метод operation-details</a> .
payment-shop	Возможность осуществлять платежи в магазины, см. <a href="#">метод request-payment</a> и <a href="#">метод process-payment</a> .

## Типы данных

Тип данных протокола	Соответствующий JSON-тип	Описание
string	string	Текстовая строка в кодировке UTF-8.
amount	number	Сумма. Число с фиксированной точкой, два знака после точки.
boolean	string	Логическое значение: true, false.
int	number	32-битное знаковое целое число.
datetime	string	Временная метка согласно стандарту <a href="#">RFC3339</a> в следующем формате (см. расшифровку ниже): YYYY-MM-DDThh:mm:ss.fZZZZZ

Расшифровка формата datetime:

- YYYY — год, всегда 4 цифры;
- MM — месяц, всегда 2 цифры (01=Январь, и т. д.);
- DD — день месяца, точно 2 цифры (от 01 до 31);
- T — латинский символ «Т», должен быть в верхнем регистре;
- hh — часы, всегда 2 цифры (24-часовой формат, от 00 до 23);
- mm — минуты, всегда 2 цифры (от 00 до 59);
- ss — секунды, всегда 2 цифры (от 00 до 59);
- f — дробная часть секунды (до шести цифр), может отсутствовать, в этом случае следует опускать и точку-разделитель;
- ZZZZZ- Описатель временной зоны, обязательный параметр. Может принимать значения:
  - Z - UTC, символ "Z" должен быть в верхнем регистре;
  - +hh:mm или -hh:mm – смещение относительно UTC (показывает, что указано локальное время, которое на данное число часов и минут опережает или отстает от UTC).

Пример:

2004-03-24T19:00:00.000+03:00 – 19 часов 24 марта 2004 года, часовой пояс Санкт-Петербурга (Москвы) – UTC+3 часа.

**См. также**

[Date and Time on the Internet: Timestamps](#)

## Получение информации о счете пользователя

- [Метод account-info](#)
- [Метод operation-history](#)
- [Метод operation-details](#)

### Метод account-info

#### Описание

Получение информации о текущем состоянии счета пользователя.

Требуемые права токена: [account-info](#)

#### Входные параметры

Нет.

#### Возвращает

В случае успеха возвращает JSON-документ со следующим содержимым:

Параметр	Тип	Описание
account	string	Номер счета пользователя.
balance	amount	Остаток на счете пользователя.
currency	string	Код валюты счета пользователя. В текущей реализации всегда 643 (рубль РФ по стандарту ISO 4217).

Пример запроса:

```
POST /api/account-info HTTP/1.1
Host: money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer 01234567890ABCDEF01234567890
Content-Length: 0
```

Пример ответа:

```
{
  "account" : "4100123456789",
  "balance" : 1000.00,
  "currency" : "643"
}
```

### Метод operation-history

#### Описание

Метод позволяет просматривать историю операций (полностью или частично) в постраничном режиме. Записи истории выдаются в обратном хронологическом порядке.

Требуемые права токена: [operation-history](#).

## Входные параметры

Параметр	Тип	Описание
<b>Необязательные</b>		
type	string	Перечень типов операций (см. <a href="#">таблицу</a> ), которые требуется отобразить. Типы операций перечисляются через пробел. В случае, если параметр отсутствует, выводятся все операции.
start_record	amount	Порядковый номер первой записи в выдаче. По умолчанию 1 (см. <a href="#">замечание</a> ).
records	string	Количество запрашиваемых записей истории операций. Допустимые значения: от 1 до 100, по умолчанию 30.

Типы операций:

Тип	Описание
deposition	Пополнение счета (приход).
payment	Платеж со счета (расход).

## Возвращает

В случае успеха возвращает следующие параметры:

Параметр	Тип	Описание
error	string	Код ошибки, присутствует при ошибке выполнения запроса.
next_record	string	Порядковый номер первой записи на следующей странице истории операций. Присутствует, только если следующая страница существует (см. <a href="#">замечание</a> ).
operations	array	Список операций.

Параметры операции:

Параметр	Тип	Описание
<b>Обязательные</b>		
operation_id	string	Идентификатор операции.
datetime	datetime	Дата и время совершения операции.
title	string	Краткое описание операции (название магазина или источник пополнения).
<b>Необязательные</b>		
pattern_id	string	Идентификатор шаблона платежа, по которому совершен платеж. Присутствует только для платежей.
direction	amount	Направление движения средств. Может принимать значения: <ul style="list-style-type: none"> <li>in (приход);</li> <li>out (расход).</li> </ul>
amount	string	Сумма операции.

В случае ошибки выполнения операции возвращается ее код:

### error

Код ошибки:

- `illegal_param_type` — неверное значение параметра `type`.
- `illegal_param_start_record` — неверное значение параметра `start_record`.
- `illegal_param_records` — неверное значение параметра `records`.

Все прочие значения: техническая ошибка, повторите вызов операции позднее.

#### Примечание:

При значительном количестве записей в истории список операций выдается постранично. По умолчанию выдается первая страница истории. Если есть хотя бы одна последующая страница, то в ответе присутствует параметр `next_record`, определяющий порядковый номер ее первой записи.

Для запроса следующей страницы истории повторите запрос с теми же параметрами, добавив параметр `start_record` и указав в нем порядковый номер первой записи следующей страницы, полученный ранее из параметра `next_record`.

### Запрос полной истории

Запрос:

```
POST /api/operation-history HTTP/1.1
Host: money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Authorization: Bearer 01234567890ABCDEF01234567890

type=deposition%20payment&records=3
```

Ответ:

```
{
  "next_record" : "4",
  operations : [
    {
      "operation_id" : "1234567",
      "pattern_id" : "2904",
      "direction" : "out",
      "amount" : "500.00",
      "datetime" : "2011-03-11T20:43:00.000+03:00",
      "title" : "Оплата ADSL-доступа компании XXX",
    },
    {
      "operation_id" : "1234568",
      "pattern_id" : "2901",
      "direction" : "out",
      "amount" : "300.00",
      "datetime" : "2011-03-10T20:43:00.000+03:00",
      "title" : "Прямое пополнение счета телефона YYY",
    },
    {
      "operation_id" : "1234569",
      "direction" : "in",
      "amount" : "1000.00",
      "datetime" : "2011-03-10T20:40:00.000+03:00",
      "title" : "Банк ZZZ, пополнение",
    }
  ]
}
```

Ответ при неверно заданном параметре:

```
{
  "error" : "illegal_param_type"
}
```

## Запрос последующих страниц истории платежей

```
POST /api/operation-history HTTP/1.1
Host: money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Authorization: Bearer 01234567890ABCDEF01234567890

type=payment&records=20&start_record=120
```

## Метод operation-details

### Описание

Получение детальной информации по операции из истории.

Требуемые права токена: [operation-details](#).

### Входные параметры

Параметр	Тип	Описание
<b>Обязательные</b>		
operation_id	string	Идентификатор операции, полученный из ответа метода <a href="#">operation-history</a> .

### Возвращает

В случае успеха возвращает следующие параметры:

Параметр	Тип	Описание
error	string	Код ошибки, присутствует при ошибке выполнения запроса.
operation_id	string	Идентификатор операции.
pattern_id	string	Идентификатор шаблона платежа, по которому совершен платеж. Присутствует только для платежей.
direction	amount	Направление движения средств. Может принимать значения: <ul style="list-style-type: none"> <li>in (приход);</li> <li>out (расход).</li> </ul>
amount	string	Сумма операции.
datetime	datetime	Дата и время совершения операции.
title	string	Краткое описание операции (название магазина или источник пополнения).
details	string	Детальное описание платежа. Строка произвольного формата, может содержать любые символы и переводы строк.

В случае ошибки выполнения операции возвращается ее код:

#### error

Код ошибки:

- illegal\_param\_operation\_id — неверное значение параметра operation\_id.

Все прочие значения: техническая ошибка, повторите вызов операции позднее.

Пример запроса:

```
POST /api/operation-details HTTP/1.1
Host: money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Authorization: Bearer 01234567890ABCDEF01234567890

operation_id=1234567
```

**Пример ответа:**

```
{
  "operation_id" : "1234567",
  "pattern_id" : "2904",
  "amount" : "500.00",
  "direction" : "out",
  "datetime" : "2011-03-11T20:43:00.000+03:00",
  "title" : "Оплата ADSL-доступа компании Мой-Провайдер",
  "details" : "Предоплата услуг ADSL-доступа в интернет компании ООО \"XXX\" \nНомер лицевого счета абонента: \n1234567/89\nЗачисленная сумма: 500.00\nНомер транзакции: 2000002967767"
}
```

**Пример ответа при запросе несуществующей операции:**

```
{
  "error" : "illegal_param_operation_id"
}
```

## Проведение платежей

Сценарий платежа в системе Яндекс.Деньги с точки зрения пользователя:

1. Платежи проводятся на основе *шаблона платежа* с заполнением пользовательских параметров. Приложение должно отобразить пользователю форму с запросом пользовательских параметров платежа (сумма, номер телефона, номер договора, и т. п.).
2. Приложение отправляет *запрос платежа*, содержащий идентификатор шаблона платежа и пользовательские параметры платежа, введенные пользователем (вызывается метод [request-payment](#)). Сервер Яндекс.Денег проверяет параметры платежа, проверяет возможность проведения платежа в магазине, возвращает идентификатор и текст контракта платежа.
3. Приложение отображает пользователю контракт платежа, пользователь может подтвердить платеж или отказаться от него.
4. В случае подтверждения платежа приложение отправляет запрос на подтверждение платежа (вызов метода [process-payment](#)) с указанием идентификатора запроса, полученного с помощью вызова метода [request-payment](#).

### Примечание:

1. Средства со счета пользователя списываются при вызове метода [process-payment](#).
2. При повторном вызове [process-payment](#) с теми же параметрами метод возвратит состояние ранее проведенного платежа.
3. В случае обрыва связи или иных проблем, приложение должно повторить вызов метода с теми же параметрами.

- [Метод request-payment](#)
- [Метод process-payment](#)

## Метод request-payment

### Описание

Запрос платежа.

Требуемые права токена: [payment-shop](#).

### Входные параметры

Параметр	Тип	Описание
<b>Обязательные</b>		
pattern_id	string	Идентификатор шаблона платежа.
*	string	Любые пользовательские параметры шаблона платежа.

## Возвращает

В случае успеха возвращает следующие параметры:

Параметр	Тип	Описание
<b>Обязательные</b>		
status	string	Код результата выполнения операции (см. <a href="#">таблицу</a> ).
<b>Необязательные</b>		
error	string	Код ошибки при проведении платежа (пояснение к полю status). Присутствует только при ошибках.
error_description	string	Дополнительное текстовое пояснение к ошибке.
request_id	string	Идентификатор запроса платежа, сгенерированный системой. Присутствует только при успешном выполнении метода.
contract	string	Текст описания платежа (контракт). Присутствует только при успешном выполнении метода.

Код результата выполнения операции:

Код	Описание
sucess	Успешное выполнение.
refused	Отказ в проведении платежа, объяснение причины отказа содержится в поле error. Это конечное состояние платежа.

В случае ошибки выполнения операции возвращается ее код:

### error

Код ошибки:

- `illegal_params` — отсутствуют или имеют недопустимые значения обязательные параметры платежа;
- `payment_refused` — магазин отказал в приеме платежа (например, пользователь попытался пополнить счет мобильного телефона, указав несуществующий номер).

Все прочие значения: техническая ошибка, повторите платеж через несколько минут.

### Примечание:

Следует учитывать, что при выполнении запроса платежа система Яндекс.Деньги связывается с сервером магазина и в некоторых случаях время ответа метода может составлять до 30 секунд, так как определяется временем получения ответа от сервера магазина. На время работы метода `request-payment` приложение должно отображать пользователю сообщение об ожидании ответа от магазина.

#### Пример запроса:

```
POST /api/request-payment HTTP/1.1
Host: money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Authorization: Bearer 01234567890ABCDEF01234567890

pattern_id=2904&phone-prefix=921&phone-number=9538416&sum=300.00
```

#### Пример ответа (успешное завершение):

```
{
  "status": "success",
  "request_id": "1234567"
  "contract": "Оплата услуг ОАО Мегафон Северо-Западный филиал, номер +7 9xx xxx xxxx,
  сумма 300.00 руб"
}
```

Пример ответа (отказ):

```
{
  "status": "refused",
  "error": "payment_refused",
  "error_description": "Абонент не существует"
}
```

См. также  
[Метод process-payment](#)

## Метод process-payment

### Описание

Подтверждение платежа.

Требуемые права токена: [payment-shop](#).

### Входные параметры

Параметр	Тип	Описание
<b>Обязательные</b>		
request_id	string	Идентификатор запроса, полученный с помощью метода <a href="#">request-payment</a> .

### Возвращает

В случае успеха возвращает следующие параметры:

Параметр	Тип	Описание
<b>Обязательные</b>		
status	string	Код результата выполнения операции (см. <a href="#">таблицу</a> ).
<b>Необязательные</b>		
error	string	Код ошибки при проведении платежа (пояснение к полю status). Присутствует только при ошибках.
payment_id	string	Идентификатор проведенного платежа. Присутствует только при успешном выполнении метода.

Код результата выполнения операции:

Код	Описание
sucess	Успешное выполнение (платеж проведен). Это конечное состояние платежа.
refused	Отказ в проведении платежа, объяснение причины отказа содержится в поле error. Это конечное состояние платежа.

В случае ошибки выполнения операции возвращается ее код:

**error**

Код ошибки:

- `contract_not_found` — отсутствует выставленный контракт с заданным `request_id`;
- `not_enough_funds` — недостаточно средств на счете плательщика;
- `limit_exceeded` — превышен лимит на сумму операции или сумму операций за период времени для выданного токена авторизации. Приложение должно отобразить соответствующее диалоговое окно.

Все прочие значения: техническая ошибка, приложению следует провести **новый** платеж через несколько минут.

Пример запроса:

```
POST /api/process-payment HTTP/1.1
Host: money.yandex.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Authorization: Bearer 01234567890ABCDEF01234567890

request_id=1234567
```

Пример ответа (успешное завершение):

```
{
  "status": "success",
  "payment_id": "2ABCDE123456789"
}
```

Пример ответа (отказ):

```
{
  "status": "refused",
  "error": "not_enough_funds",
}
```

**См. также**  
[Метод request-payment](#)

# Предметный указатель

## #

3DES 11

## A

access token 9  
Access Token Request 6  
access\_token 6, 10, 11  
account 16  
account-info 14, 16  
amount 14, 17, 19  
API 4  
Authorization 13  
authorization code 8, 9  
Authorization Request 6, 7  
Authorization Response 6  
authorization\_code 9  
authorize 14

## B

balance 16  
boolean 14

## C

client\_id 5, 7–10  
code 7–9  
contract 22  
cookie 6, 12  
currency 16

## D

datetime 14, 17, 19  
deposition 17  
details 19  
direction 17, 19

## E

error 8, 10, 13, 17, 19, 22, 23  
error\_description 13, 22

## G

grant\_type 9

## H

HTTP Redirect 6, 8

## I

in 17, 19  
insufficient\_scope 13  
int 14  
invalid\_request 13

invalid\_token 13

## N

next\_record 17

## O

OAuth 5  
operation-details 14, 19  
operation-history 14, 16  
operation\_id 17, 19  
operations 17  
out 17, 19

## P

pattern\_id 17, 19, 21  
payment 17  
payment-shop 14, 21, 23  
payment\_id 23  
process-payment 23

## R

records 17  
redirect\_uri 6, 7, 9  
refused 22, 23  
Registration Request 5  
request-payment 21, 22  
request\_id 22–24  
response\_type 7

## S

scope 8, 14  
start\_record 17  
status 22, 23  
string 14  
sucess 22, 23

## T

title 17, 19  
type 17

## W

WWW-Authenticate 13

## Л

логин 7

## П

пароль 7

## Т

токен 7

## Я

Яндекс.Деньги 4



API Яндекс.Денег  
Руководство разработчика

19.04.2011