



# Общая информация по API

## Определения

**Продавец** (он же **мерчант**) – пользователь системы Onpay.ru (интернетмагазин, НКО, ИП и т.п.) обладающий правами получать платежи от третьих лиц через систему Onpay.ru

**Клиент** – любой пользователь системы Продавца, желающий переводить деньги Продавцу через систему Onpay.ru (покупка товаров и услуг, благотворительные пожертвования и проч.)

## Подключение API

Для включения работы по API, Вам нужно:

1. Активировать функцию **“Уведомление по API”** в разделе Настройки Магазина в Кабинете Продавца Onpay
2. Указать **APIURL** в соответствии со скриптом API на Вашем сервере. В **APIURL** возможно использование кириллицы.
3. Установить секретный ключ для **“уведомлений по API”**, который должен быть таким же, как в скрипте на Вашем сервере, чтобы позволить генерацию контрольных подписей.

Разрешается использование параметров в указываемом APIURL, по которому будут приходить уведомления. При этом накладывается ограничение в виде списка зарезервированных API Onpay параметров, которые запрещается использовать в APIURL настроек магазина. Список зарезервированных параметров прилагается:

*onpay\_id*  
*pay\_for*  
*order\_amount*  
*order\_currency*  
*balance\_amount*  
*balance\_currency*  
*exchange\_rate*  
*type*  
*comment*  
*paymentDateTime*  
*md5*

## Описание и очередность транзакций

**Onpay.ru производит 2 вида запросов к системе Продавца через API:**

1. Запрос **“check”** используется, чтобы получить разрешение от системы Продавца на прием платежа от Клиента. После удачного получения разрешения, Onpay.ru одобрит платёж. С этого момента, если Клиент действительно производит платёж, Продавец может видеть его во вкладке **“Платежи”** в Кабинете Продавца [<http://wiki.onpay.ru/doku.php?id=cabinet>].
2. Запрос **“pay”** является, по сути, уведомлением для системы Продавца, о том, что для него принят платеж. После получения уведомления, система Продавца может автоматически отправить заказанные товары или сервисы Клиенту.

### **Очередность транзакции:**

1. Клиент переходит по платежной ссылке на форму Опрау и заполняет данные(если необходимо).
2. Опрау.ru отправляет “check” запрос в адрес Продавца, удостоверяясь, что система Продавца может (и разрешает) принять платёж. Система Продавца проверяет все параметры запроса (существуют ли в системе ID Клиента и заказа, может ли Клиент платить и т.д.).
  - a. Если система Продавца не позволяет перевод (любой итог, кроме получения кода 0 от системы Продавца) – платёж не будет принят от Клиента.
  - b. Если API Продавца разрешает перевод (код 0) – Опрау разрешает Клиенту платить.
3. Клиент производит оплату
4. Опрау сохраняет платёж со статусом **“Получен”** и отправляет запрос типа “rau” в систему Продавца с теми же параметрами, что и запрос “check”, плюс ID платежа и дата/время момента, когда платёж был одобрен.
  - a. если система Продавца приняла уведомление (код 0) – Опрау.ru изменяет статус этого платежа с **“Получен”** на **“Принят”**
  - b. если система Продавца сообщает о не критичной ошибке – Опрау.ru попытается известить систему Продавца позже. Опрау.ru будет посылать извещения с возрастающими временными интервалами, до тех пор пока система Продавца не примет уведомление или пока не пройдет 72 часа.Платежи со статусами “получен” и “принят” могут быть просмотрены во вкладке **“Платежи”** в Кабинете Продавца [<http://wiki.onpay.ru/doku.php?id=cabinet>] Опрау.ru.

Продавцы могут пометить платежи, как “Принятые” вручную во вкладке **“Платежи”** если API не доступен.

### **Типы API Опрау.ru**

1. Основное АПИ (версия 2.0). В Кабинете продавца выбран по умолчанию (поле селектора на HTTPS 2.0).
2. API 1.0. Старое апи. С этим АПИ, в частности, работают все загружаемые модули с onpaysolutions.ru
3. Специальное АПИ для Insales (его настройка описана отдельно)
4. Гибридные АПИ для упрощения миграции (переход с Робокассы и Интеркассы)

- **Описание API (2.0)**
- Старое API (1.0)
- API Insales
- Миграция с Интеркассы
- Миграция с Робокассы

## **Справочная информация**

### **Недоступность сервера магазина**

Если сервер магазина недоступен для “check” запроса или возвращает НЕположительный ответ, система Опрау не примет платёж от Клиента.

Если сервер магазина недоступен для “rau” запроса, система Опрау будет повторно отправлять запрос несколько раз в течение следующих 72 часов. Повторяющиеся запросы посылаются с увеличивающимися интервалами. При этом платеж будет зачислен на счет магазина.

# API 2.0

## Общая информация

API построено на REST архитектуре(кроме check и pay запросов). JSON возвращается в ответ на все запросы к API, в том числе и при возникновении ошибок. Имеет предсказуемые, ресурсо-ориентированные URL-адреса, использует HTTP-коды для передачи состояния ошибок, а также использует встроенные функции HTTP-аутентификации и методы GET, POST, PUT, DELETE.

Запросы отправляются на URL, указанный в личном кабинете в настройках магазина. В личном кабинете также задается пароль для формирования подписи и тип API.

## Check

### Запрос к мерчанту

Для check запроса используются параметры:

Название	Тип	Описание
type	string	Тип запроса (check)
pay_for	string	Номер заказа
amount	float	Сумма платежа в центах, если параметр mode = free, то будет передан 0
way	string	Валюта платежа
mode	string	Тип платежа, fix или free
signature	string	Контрольная подпись, SHA1 от строки - «check;pay_for;amount;way;mode;secret_key»

- «check» в строке для контрольной подписи - фиксированное слово (5 букв), а не переменная

Пример запроса:

```
{
  "type": "check",
  "pay_for": "55446",
  "amount": 500.0,
  "way": "RUR",
  "mode": "fix",
  "signature": "82f67760dbc5331963b7e00bc6df77f1"
}
```

### Ответ мерчанта

Название	Тип	Описание
status	boolean	Статус ответа, true для подтверждения, false для отказа.
pay_for	string	Номер заказа
signature	string	Контрольная подпись, SHA1 от строки - «check;status;pay_for;secret_key»

- «check» в строке для контрольной подписи - фиксированное слово (5 букв), а не переменная
- status в строке для контрольной подписи - строка («true» или «false»)

Пример:

```
{
  "status": true,
  "pay_for": "55446",
  "signature": "172de16ada92791b3753b3121d471f5c"
}
```

## Pay

### Запрос к мерчанту

Для pay запроса используются параметры:

Название	Тип	Описание
type	string	Тип запроса (pay)
signature	string	Контрольная подпись, SHA1 от строки - «pay;pay_for;payment.amount;payment.way;balance.amount;balance.way;secret_key», где pay - строковая константа «pay», «;» - символ «точка с запятой»
pay_for	string	Номер заказа
user.email	string	E-mail плательщика
user.phone	string	Телефон плательщика
user.note	string	Комментарий плательщика
payment.id	int	Номер платежа
payment.date_time	string	Дата создания платежа в формате «CCYY-MM-DDThh:mm:ssTZD» где TZD смещение часового пояса в формате [+ -]hh:mm
payment.amount	float	Сумма платежа
payment.way	string	Валюта платежа
payment.rate	float	Курс обмена между валютами balance.way/payment.way
payment.release_at	string	Время зачисления платежа, для отложенных платежей строится аналогично payment.date_time, null если уже зачислен
balance.amount	float	Сумма, зачисляемая на баланс
balance.way	string	Валюта зачисления на баланс
order.from_amount	float	Сумма из ордера, которую должен был заплатить плательщик
order.from_way	string	Валюта из ордера, в которой должен был заплатить плательщик
order.to_amount	float	Сумма из ордера, которая должна была поступить на баланс магазина
order.to_way	string	Валюта из ордера, в которой должен был пополниться баланс магазина

- «pay» в строке для контрольной подписи - фиксированное слово (3 буквы), а не переменная
- при прямых платежах поля order.\* отсутствуют

Пример запроса:

```
{
  "type": "pay",
  "signature": "82f67760dbc5331963b7e00bc6df77f1",
  "pay_for": "55446",
  "user": {
    "email": "mail@mail.ru",
    "phone": "9631478946",
    "note": ""
  },
  "payment": {
    "id": 7121064,
    "date_time": "2013-12-05T12:07:09+04:00",
    "amount": 102.0,
    "way": "USD",
    "rate": 33.121445,
    "release_at": null
  },
  "balance": {
    "amount": 3378.39,
    "way": "RUR"
  },
  "order": {
    "from_amount": 102.0,
    "from_way": "USD",
    "to_amount": 3378.39,
    "to_way": "RUR"
  }
}
```

#### Ответ мерчанта

Название	Тип	Описание
status	boolean	Статус ответа, true для подтверждения, false для отказа (отказ не является отказом от платежа, а лишь информацией о том, что мерчант не знает о таком платеже, при этом у платежа проставится статус как «не было уведомления», и мерчант сможет активировать его вручную в личном кабинете, если такой платеж в действительности имеет место быть).
pay_for	string	Номер заказа
signature	string	Контрольная подпись, SHA1 от строки - «pay;status;pay_for;secret_key»

- «pay» в строке для контрольной подписи - фиксированное слово (3 буквы), а не переменная
- status в строке для контрольной подписи - строка («true» или «false»)

Пример:

```
{
  "status":true,
  "pay_for":"55446",
  "signature":"172de16ada92791b3753b3121d471f5c"
}
```

## Запросы от мерчанта к OnPay

### Получить данные платежа

адрес - `json_interfaces/payments/:id`, где `id` - номер платежа в системе OnPay

Используется для получения данных о прошедшем платеже. Используемые параметры:

Название	Тип	Описание
login	string	Логин сайта
signature	string	Контрольная подпись, SHA1 от строки - « <code>id;login;secret_key</code> »

Пример запроса:

```
{
  "login":"onpay",
  "signature":"1d15f90df20da53d7206e9f7db7d2c9d"
}
```

В ответ будет выдан JSON с данными:

Название	Тип	Описание
signature	string	Контрольная подпись, SHA1 от строки - « <code>payment.id;payment.amount;payment.way;balance.amount;balance.way;secret_key</code> »
user.email	string	E-mail плательщика
user.phone	string	Телефон плательщика
user.note	string	Комментарий плательщика
payment.id	int	Номер платежа
payment.date_time	string	Дата создания платежа в формате « <code>CCYY-MM-DDThh:mm:ssTZD</code> » где TZD смещение часового пояса в формате <code>[+ -]hh:mm</code>
payment.amount	bigint	Сумма платежа
payment.way	string	Валюта платежа
payment.rate	bigint	Курс обмена между валютами <code>balance.way/payment.way</code>
payment.release_at	string	Время зачисления платежа, для отложенных платежей строиться аналогично <code>payment.date_time</code> , null если зачислен мгновенно
balance.amount	bigint	Сумма, зачисляемая на баланс
balance.way	string	Валюта зачисления на баланс

Пример ответа:

```
{
  "signature":"172de16ada92791b3753b3121d471f5c",
  "user":{
    "email":"mail@mail.ru",
    "phone":"9631478946",
    "note":""
  },
  "payment":{
    "id":"7121064",
    "date_time":"2013-12-05T12:07:09+04:00",
    "amount":10200,
    "way":"USD",
    "rate":33121445,
    "release_at":null
  },
  "balance":{
    "amount":"3300",
    "way":"RUR"
  }
}
```

### Получить курс обмена

адрес - `json_interfaces/rates/:from/to/to`, где `from` и `to` - валюта из и валюта назначения соответственно

Используется для получения текущего курса валют

Название	Тип	Описание
login	string	Логин сайта
signature	string	Контрольная подпись, SHA1 от строки - «login;from;to;secret_key»

Пример запроса:

```
{
  "login": "onpay",
  "signature": "65ded5353c5ee48d0b7d48c591b8f430"
}
```

В ответ будет выдан JSON с данными:

Название	Тип	Описание
from	string	Валюта из
to	string	Валюта назначения
rate	bigint	значение курса * 10e6
signature	string	Контрольная подпись, SHA1 от строки - «from;to;rate;secret_key»

Пример ответа:

```
{
  "from": "USD",
  "to": "RUR",
  "rate": 33121445
  "signature": "4671aeaf49c792689533b00664a5c3ef"
}
```

## Купоны

Общий адрес интерфейса - [json\\_interfaces/coupons/](#)

### Создание Купона

метод - POST

адрес - [json\\_interfaces/coupons/](#)

Название	Тип	Описание
login	string	Логин сайта
code	string	Промо-код купона
paysystem	string	Тикер платежной системы
type	string	Тип купона - для величины в процентах percent, для постоянной суммы const
percent_off	bigint	Предоставляемая скидка, в процентах, используется только для типа percent (для const - 0)
max_amount	bigint	Ограничение сверху по предоставляемой сумме скидки, используется только для типа percent, указывается в центах. 0 для купона без ограничения (для const - 0)
value	bigint	Величина скидки, используется только для типа const, указывается в центах (для percent - 0)
min_amount	bigint	Ограничение снизу, купон может использоваться только для платежей выше указанной суммы, указывается в центах, используется только для типа const (для percent - 0)
max_redemptions	int	Ограничение сверху по количеству использований
expired_at	string	Дата истечения скидочной акции в формате «CCYY-MM-DDThh:mm:ssTZD» где TZD смещение часового пояса в формате [+ -]hh:mm
announced	boolean	Если true, то сервис Опрай будет распространять купон
signature	string	Контрольная подпись, SHA1 от строки - «login;paysystem;code;type;percent_off;max_amount;value;min_amount;max_redemptions;expired_at;announced;secret_key»

Пример запроса для процентной скидки:

```
{
  "login": "onpay",
  "code": "1",
  "paysystem": "RUR",
  "type": "percent",
  "percent_off": 10,
  "max_amount": 100000,
  "value": 0,
  "min_amount": 0,
  "max_redemptions": 1,
  "expired_at": "2013-12-05T12:07:09+04:00",
  "announced": true,
  "signature": "1d15f90df20da53d7206e9f7db7d2c9d"
}
```

Пример запроса для постоянной скидки:

```
{
  "login": "onpay",
  "code": "1",
  "paysystem": "WMR",
  "type": "const",
  "percent_off": 0,
  "max_amount": 0,
  "value": 10000,
  "min_amount": 100000,
  "max_redemptions": 1,
  "expired_at": "2013-12-05T12:07:09+04:00",
  "announced": true,
  "signature": "d41d8cd98f00b204e9800998ecf8427e"
}
```

В ответ будет выдан JSON с данными:

Название	Тип	Описание
code	string	Промо-код купона
type	string	Тип купона - для величины в процентах percent, для постоянной суммы const
percent_off	bigint	Предоставляемая скидка, в процентах, только для типа percent (для const - 0)
max_amount	bigint	Ограничение сверху по предоставляемой сумме скидки, только для типа percent. 0 для купона без ограничения (для const - 0)
value	bigint	Величина скидки, только для типа const, указывается в центах (для percent - 0)
min_amount	bigint	Ограничение снизу, купон может использоваться только для платежей выше указанной суммы, только для типа const (для percent - 0)
max_redemptions	int	Ограничение сверху по количеству использований
expired_at	string	Дата истечения скидочной акции в формате «CCYY-MM-DDThh:mm:ssTZD» где TZD смещение часового пояса в формате [+_]hh:mm
redemptions_count	int	Количество использований
state	string	Текущее состояние, new для купона с не исчерпанным лимитом использований, complete для использованного максимально допустимое кол-во раз, expired для купона с истекшим временем действия и не исчерпанным лимитом использований, deleted для удаленного купона
announced	boolean	Если true, то сервис Опрау будет распространять купон
signature	string	Контрольная подпись, SHA1 от строки - «code;type;redemptions_count;state;secret_key»

Пример ответа для процентной скидки:

```
{
  "code": "1",
  "paysystem": "RUR",
  "type": "const",
  "percent_off": 10.0,
  "max_amount": 100000,
  "value": 0.0,
  "min_amount": 0,
  "max_redemptions": 1,
  "expired_at": "2013-12-05T12:07:09+04:00",
  "redemptions_count": 0,
  "state": "new",
  "announced": true,
  "signature": "172de16ada92791b3753b3121d471f5c"
}
```

Пример ответа для постоянной скидки:

```
{
  "code": "1",
  "paysystem": "WMR",
  "type": "const",
  "percent_off": 0.0,
  "max_amount": 0,
  "value": 10000.0,
  "min_amount": 100000,
  "max_redemptions": 1,
  "expired_at": "2013-12-05T12:07:09+04:00",
  "redemptions_count": 0,
  "state": "new",
  "announced": true,
  "signature": "172de16ada92791b3753b3121d471f5c"
}
```

### Получение купона

метод - GET

адрес - json\_interfaces/coupons/:code, где code - уникальный код купона

Используется для просмотра текущего состояния купона.

Название	Тип	Описание
login	string	Логин сайта
signature	string	Контрольная подпись, SHA1 от строки - «login;code;get;secret_key»

- «get» в строке для контрольной подписи - фиксированное слово (3 буквы), а не переменная

Пример запроса:

```
{
  "login": "onpay",
  "signature": "1d15f90df20da53d7206e9f7db7d2c9d"
}
```

Ответ будет аналогичен как и у запроса для создания купона.

### Удаление купона

метод - DELETE

адрес - json\_interfaces/coupons/:code, где code - уникальный код купона

Используется для отключения купона до истечения его срока годности или исчерпания количества использований.

Название	Тип	Описание
login	string	Логин сайта
signature	string	Контрольная подпись, SHA1 от строки - «login;code;delete;secret_key»

- «delete» в строке для контрольной подписи - фиксированное слово (6 букв), а не переменная

Пример запроса:

```
{
  "login": "onpay",
  "signature": "1d15f90df20da53d7206e9f7db7d2c9d"
}
```

Ответ будет аналогичен как и у запроса для создания купона.

## Ошибки

В любом ответе могут присутствовать данные об ошибке, они записываются в конец сообщения в следующем формате:

Название	Тип	Описание
params	string	Перечисляются все ошибочные параметры
params.code	string	Код ошибки
params.message	string	Текст ошибки
params.name	string	Название параметра
type	string	Тип ошибки
message	string	Текст сообщения

JSON вида:

```
{
  "error": {
    "params": [
      {
        "code": "required",
        "message": "Description cannot be blank.",
        "name": "description"
      }
    ],
    "type": "invalid_param_error",
    "message": "Invalid data parameters"
  }
}
```

В случае возникновения ошибки общего характера, не привязанной к определенному параметру (пример - внутренняя ошибка сервера, недоступность БД), ключ 'params' можно не передавать

Тогда JSON будет вида:

```
{
  "error": {
    "type": "invalid_param_error",
    "message": "Invalid data parameters"
  }
}
```



## Примечания

1) Разделителем целой и дробной части типа float является символ «.» (точка).

Следующие числа будут выглядеть следующим образом:

123 → 123.0

123.0 → 123.0

123.00 → 123.0

123.001 → 123.0

В этом же представлении числа должны быть использованы при подсчете подписи.

2) Методы запросов:

Запрос	метод
check	POST
pay	POST
state	GET
rate	GET
Создание купона	POST
Получение купона	GET
Удаление купона	DELETE