

webhook-нотификации

Если Вы хотите получать статус транзакции, использовать рекуррентные или двухстадийные платежи, то Вам может быть интересна возможность получать нотификации об оплате. Для этого Вам необходимо [при создании сервиса](#) указать **URL скрипта для получения веб-хуков**.

В случае завершения платежа информация о нем будет передана Вам **POST**--запросом на данный URL, а также **GET**--запросом на URL страницы успешной покупки (или ошибки, если совершить транзакцию не удалось). Таким образом в передаче параметров нотификации участвует три поля из настроек вашего сервиса:

- URL скрипта для получения веб-хуков
- URL страницы успешной покупки
- URL страницы ошибки

В заголовках Content-Type **всегда равен** application/x-www-form-urlencoded. Набор параметров для обеих рассылок идентичен. Таким образом вы можете уведомлять пользователя о деталях оплаченного заказа после редиректа на страницу успеха, основываясь на данных, полученных из нотификации: номер заказа, стоимость, дата и т.д. Подробности ниже в таблице параметров (см. оглавление).

Если при создании сервиса было заполнено поле email, то на указанную электронную почту также будет выслана информация о платеже.



Повторы

Запросы от нашего сервера в случае первой неудачной попытки передаются со следующей периодичностью: 180, 180, 180 с.

- Версия нотификаций 1.0 (по умолчанию)
 - Таблица параметров, передаваемых на URL скрипта для получения дополнительных параметров оплаты
 - Формирование подписи check для первой версии API
 - Пример на PHP
 - Пример на Python
 - Примечания к версии 1.0
- Версия нотификаций 2.0 (по требованию)
 - Пример скрипта для расчёта

Редактирование сервиса «Демо сервис (alba@life-pay.ru)»

Тестовый

* — обязательные поля

* **Название:**
Демо сервис (alba@life-pay.ru)

В качестве продавца будет выводиться текст, указанный в этом поле

Получать дополнительные параметры оплаты

URL скрипта для получения дополнительных параметров оплаты:

Этот скрипт будет вызываться для передачи дополнительных параметров оплаты. Вы можете получать информацию о частичной оплате и т.д. [Узнать подробнее](#)

Подтверждение резервирования товара | Нет

* **URL страницы успешной покупки:**

Например: <http://www.test.ru/success.html>

* **URL страницы ошибки:**

Например: <http://www.test.ru/error.html>

Версия нотификаций 1.0 (по умолчанию)

Таблица параметров, передаваемых на URL скрипта для получения дополнительных параметров оплаты

| Параметр | Версия | Описание |
|----------|---------|---------------|
| tid | 1.0.1.1 | ID транзакции |

| | | |
|---------------------|---------------------------------|---|
| name | 1 . 0 / 1 . 1 | Название товара или услуги. Отображается на странице оплаты. |
| comment | 1 . 0 / 1 . 1 | Комментарий платежа переданный в процессе инициализации платежа . |
| partner_id | 1 . 0 / 1 . 1 | ID партнера, то есть ваш ID |
| service_id | 1 . 0 / 1 . 1 | ID сервиса |
| order_id | 1 . 0 / 1 . 1 | ID заказа |
| type | 1 . 0 / 1 . 1 | Тип платежа |
| currency | 1 . 0 / 1 . 1 | Валюта операции, поддерживается только RUB, не участвует в формировании подписи при version = (1.0 1.1) |
| cost | 1 . 0 / 1 . 1 | Общая сумма заказа, переданная при инициализации платежной операции |
| income_total | 1 . 0 / 1 . 1 | Общая сумма в рублях, заплаченная покупателем, может отличаться от income и system_income только в случае оплаты частями или при переносе комиссии на плательщика |

| | | |
|-----------------------|---------------------------------|--|
| income | 1 . 0 , 1 . 1 | Сумма в рублях, полученная от платежного инструмента по данной платежной транзакции |
| partner_income | 1 . 0 , 1 . 1 | Сумма в рублях, дохода магазина по данной платежной транзакции |
| system_income | 1 . 0 , 1 . 1 | Сумма в рублях, заплаченная покупателем по данной платежной транзакции |
| command | 1 . 0 , 1 . 1 | <p>Текущее действие:</p> <ul style="list-style-type: none"> • command=cancel – получен отказ от платежного канала, расшифровка причины в поле resultStr • command=success – вызывается при полной оплате сервиса • command=recurrent_cancel – вызывается в случае, если держатель карты отменил рекуррентные платежи. • command=recurrent_expire – вызывается когда истёк срок рекуррента. • command=refund – вызывается в результате выполнения операции отмены платежа. В поле result=ok или fail. А в resultStr – причина отказа. • command=authorize_payment – вызывается при использовании двойной авторизации при платеже • command=funds_blocked – вызывается при использовании двух этапной оплаты (BLOCK + CHARGE). Описание в разделе “Двухэтапные платежи (предавторизация)”. <p><u>ВАЖНО: в случае полной оплаты сервиса придут и success и process.</u></p> |
| result | 1 . 0 , 1 . 1 | Только для command=refund , значения 'ok' или 'fail' |
| resultStr | 1 . 0 , 1 . 1 | Текст уведомления. |
| version | 1 . 0 , 1 . 1 | Версия протокола уведомления. (На данный момент: 1.0, 1.1). Версия по умолчанию — 1.0. Переключение на другие версии производится на стороне Провайдера. |
| phone_number | 1 . 0 , 1 . 1 | Опционально, номер телефона |

| | | |
|------------------------------------|---------------------------------|--|
| email | 1 . 0 , 1 . 1 | Опционально, email |
| date_created | 1 . 0 , 1 . 1 | дата и время создания транзакции, формат 'YYYY-MM-DD HH24.MI.SS' (MSK) |
| recurrent_order_id | 1 . 0 , 1 . 1 | ID заказа (order_id), который был передан при первом вызове рекуррентного платежа (только для рекуррентных операций) |
| card | 1 . 0 , 1 . 1 | v 1.0 Маскированный номер карты, в случае если проведенный платеж является рекуррентным (только для рекуррентных операций) При v1.1 и v2.0 заполнено всегда, если оплата осуществлялась с вводом номера карты. |
| card_binding_id | 1 . 0 , 1 . 1 | уникальный токен для сохраненных данных карты. Устарело. |
| test | 1 . 0 , 1 . 1 | Значение 1 (только для тестовых платежей), См. описание |
| check | 1 . 0 , 1 . 1 | MD5 хеш от параметров: tid + name + comment + partner_id + service_id + order_id + type + cost + income_total + income + partner_income + system_income + command + phone_number + email + result + resultStr + date_created + version + card + recurrent_order_id + test + secret_key Где secret_key – секретный ключ сервиса. Пример формирования продемонстрирован ниже. |
| check (если command=refund) | 1 . 0 , 1 . 1 | MD5 хеш от параметров: 'tid'+ 'name'+ 'comment'+ 'partner_id'+ 'service_id'+ 'order_id'+ 'type'+ 'cost'+ 'command'+ 'result'+ 'resultStr'+ 'phone_number'+ 'email'+ 'date_created'+ 'version'+ 'secret_key' |
| refund_ext_id | 1 . 0 , 1 . 1 | дополнительный id возврата при осуществлении нескольких возвратов в рамках транзакции. Подробнее |



- **resultStr** – текст уведомления. Для значения параметра **command=success** он стандартный. Для **command=cancel** выводится то, что ответит платёжный шлюз.

Формирование подписи check для первой версии API



Версия меняется по запросу в [отдел технической интеграции](#)

Все параметры участвуют в формировании подписи check. Подпись формируется как md5 от всех параметров, сцепленных в строку без пробелов + добавленный в конце секретный ключ сервиса.

Пример на PHP

```
1 $param['check']==md5($param['tid'].$param['name'].$param['comment'].$param['partner_id'].$param['service_id'].$param['order_id'].$param['type'].$param['cost'].$param['income_total'].$param['income'].$param['partner_income'].$param['system_income'].$param['command'].$param['phone_number'].$param['email'].$param['result'].$param['resultStr'].$param['date_created'].$param['version'].$secretKey);
```

Формирование подписи check на примере php скрипта

```
<?php

$params = array
(
    'comment' => '',
    'phone_number' => '74952760800',
    'order_id' => '67',
    'currency' => 'RUB',
    'cost' => '511.0',
    'date_created' => '2021-01-28 21:35:49',
    'partner_id' => '250305',
    'resultStr' => '',
    'name' => '',
    'system_income' => '511.0',
    'income_total' => '511.0',
    'partner_income' => '434.35',
    'version' => '1.0',
    'command' => 'success',
    'income' => '511.0',
    'tid' => '474541305',
    'service_id' => '85494',
    'type' => 'spg_test',
    'email' => 'test@mail.ru',
    'result' => ''
);

$secretKey = 'c9264d756f170802c4eaf9405077b946';

echo rawurlencode(md5($param['tid'].$param['name'].$param['comment'].$param['partner_id'].$param['service_id'].$param['order_id'].$param['type'].$param['cost'].$param['income_total'].$param['income'].$param['partner_income'].$param['system_income'].$param['command'].$param['phone_number'].$param['email'].$param['result'].$param['resultStr'].$param['date_created'].$param['version'].$secretKey));
```

Пример на Python

checker_md5

```
import hashlib
from urllib.parse import parse_qs
import pprint

secretkey = '262eb24f12d0c3fdd990eae096016055'

params_list = [
    'tid',
    'name',
    'comment',
    'partner_id',
    'service_id',
    'order_id',
    'type',
    'cost',
    'income_total',
    'income',
    'partner_income',
    'system_income',
    'command',
    'phone_number',
    'email',
    'result',
    'resultStr',
    'date_created',
    'version',
    'secretKey'
]

# webhook-, URL :
qs = 'comment=&phone_number=79165483580&order_id=00000015&cy=RUB&cost=75.0&date_created=2022-03-29+22%3A38%3A08&partner_id=250305&check=66b522b5749bfe713ac089a55a013725&resultStr=%D1%82%D1%80%D0%B0%D0%BD%D0%B7%D0%B0%D0%BA%D1%86%D0%B8%D1%8F+%D0%BE%D0%BF%D0%BB%D0%B0%D1%87%D0%B5%D0%BD%D0%B0+%D1%87%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%BE&name=Acquiring+lifepay+00000015&system_income=75.0&income_total=75.0&partner_income=63.75&version=1.0&command=process&income=75.0&tid=491789584&service_id=87875&type=ipsp_test_cards_01&email=awa77%40mail.ru'
params = parse_qs(qs)
pp = pprint.PrettyPrinter()
pp.pprint(params)
values = []

for i in params_list:
    values.append(params.get(i, [''])[0])

md5_str = ''.join(values)
md5_str += secretkey
print(md5_str)
md5 = hashlib.md5(md5_str.encode('UTF-8')).hexdigest()
print(' MD5', md5)
print('MD5    ',params['check'][0])
```

Примечания к версии 1.0

Для рекуррентных платежей добавляются два поля **card** и **recurrent_order_id** и строка для подписи MD5 формируется следующим образом:

tid + name + comment + partner_id + service_id + order_id + type + cost + income_total + income + partner_income + system_income + command + phone_number + email + resultStr + date_created + version + **card** + **recurrent_order_id** + secret_key

Версия уведомлений 2.0 (по требованию)

В расчёте участвуют все параметры, возвращающиеся на ваш URL (см. **поля доступные для заполнения** в статье [настройки сервиса](#)), за исключением значения параметра check, с которым вы будете сравнивать результат своего расчёта.



Прежде чем приступать к сравнению подписей, убедитесь в работе рассылки уведомлений по версии 2.0. Подпись check всегда присутствует в теле уведомления.

Пример подписи check по первой версии (MD5). Все символы в нижнем регистре: `d8828f9f58be88385a67d9cc48e3c99e`

Пример подписи check по второй версии: `2u0kGxj9a91qfzC8To51RB3Az/wCBhOXjOqgDE8Dwrk=`

К перечисленным выше параметрам в уведомление добавляются следующие:

| Параметр | Версия | Описание |
|------------|--------|--|
| cardholder | 2.0 | Имя держателя карты если присутствует в транзакции |
| paid_date | 2.0 | Дата и время оплаты транзакции (подтверждения оплаты платежным каналом), формат 'YYYY-MM-DD HH24.MI.SS' (MSK) |
| version | 2.0 | Меняет своё значение с 1.0 на 2.0. Для переключения в этот режим обратитесь в отдел технической интеграции . |

Для написания своего скрипта расчёта подписи обратитесь к статье [Алгоритм формирования подписи для API v 2.0](#).

Пример скрипта для расчёта

Python3

```
from urllib.parse import urlparse, parse_qs, quote
import hashlib
import base64
import hmac

def sign_by_url(method, url, params, secret_key,
               exclude=['check', 'mac'], exclude_port=True):
    o = urlparse(url)
    host = o.hostname if exclude_port else o.netloc
    return sign(method, host, o.path, params, secret_key, exclude)

def sign(method, host, uri, params, secret_key, exclude=['check', 'mac']):
    """
    HTTP
    """
    keys = [param for param in params if param not in exclude]
    keys.sort()

    result = []
    for key in keys:
        value = quote(
            str(params.get(key) or '').encode('utf-8'),
            safe='~'
        )
        result.append('{}={}'.format(key, value))

    data = "\n".join([
        method,
        host,
        urlparse(uri).path, # QUERY_STRING, uri
        "&".join(result)
    ])

    data = data.encode('utf-8')
    secret_key = secret_key.encode('utf-8')

    digest = hmac.new(
        secret_key,
        data,
        hashlib.sha256
    ).digest()
    signature = base64.b64encode(digest)

    return signature

#
qs = "comment=&phone_number=0&order_id=0&cy=RUB&cost=100.0&date_created=2022-06-30+11%3A46%3A22&partner_id=250305&check=nsxegvtGyPnZ4iE4GXe5iPKRjKjhi5%2FejN2sfErAewE%3D&card=220138XXXXX0013&resultStr=%D1%82%D1%80%D0%B0%D0%BD%D0%B7%D0%B0%D0%BA%D1%86%D0%B8%D1%8F+%D0%BE%D0%BF%D0%BB%D0%B0%D1%87%D0%B5%D0%BD%D0%B0+%D0%BF%D0%BE%D0%BB%D0%BD%D0%BE%D1%81%D1%82%D1%8C%D1%8E&name=Life+Pay&system_income=100.0&income_total=100.0&paid_date=2022-06-30+11%3A46%3A41.355627&partner_income=96.6&version=2.0&command=success&cardholder=TEST+TEST&income=100.0&tid=491825313&service_id=67279&type=spg_test&email="

params = {k: v[0] for k, v in parse_qs(qs, keep_blank_values=True).items()}
secret_key = "262eb24f12d0c3fdd990eae096016055" #
url = "https://96d8-109-63-129-14.eu.ngrok.io" # URL -

print(" ", params['check'].encode('UTF-8'))
signature = sign_by_url('POST', url, params, secret_key)
print(" ", signature) # !
```

