

# Платёжная кнопка

В настоящий момент функционал временно недоступен.

Простой способ обеспечить приём платежей на любом сайте - поместить рядом с описанием товара или услуги кнопку, которая откроет платёжную форму в модальном окне (iframe). Пример использования платёжной кнопки вы можете посмотреть на [этой странице](#).

Вы можете использовать данные [тестовых карт](#), чтобы посмотреть, как это работает.

## Подробные сведения

### Предварительные действия

Перед началом подготовки своего сайта вы должны получить у технической поддержки специальный токен: алфавитно-цифровое значение, которое следует использовать в коде сайта для создания платёжной кнопки. На настоящем сайте для примера будет использоваться токен YRF3C5RFICWISEWFR6GJ.

### Изменения в теге <head> страницы магазина

Чтобы встроить платёжную кнопку на страницу магазина, вы должны внести изменения в содержимое тега <head> страницы, на которой будет размещена платёжная кнопка.

Страница вашего сайта, на которой будет использоваться платёжное решение, должна быть сохранена в кодировке UTF-8, а также на ней должен присутствовать meta-тег <meta name=<viewport> content=<width=device-width>>.

В тег <head> страницы добавьте:

- путь к сценарию JavaScript (`ipay.js`), размещённым на сервере платёжного шлюза;
- токен продавца, полученный у технической поддержки.

```
<head>
    <title>Тестовая страница</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <script
```

```

src="https://securecardpayment.ru/payment/docsite/assets/js/ipay.js"></script>
<script>
    var ipay = new IPAY({api_token: 'YRF3C5RFICWISEWFR6GJ'});
</script>
</head>

```

Ссылки ведут на тестовый стенд. По окончании тестирования замените ссылки, чтобы они вели на основной стенд.

## Изменения в теле страницы магазина

В тело страницы необходимо добавить ссылку на сценарий JavaScript (ipay-modal.js), в котором можно указать следующие параметры.

Название	Описание	Обязательно	Тип
amount	Сумма платежа. Позволяет указать сумму в рублях с копейками, при этом в качестве разделителя используется точка, например: 499.99 (четыреста девяносто девять рублей, девяносто девять копеек). Если не указывать сумму платежа, покупатель сможет ввести произвольную сумму самостоятельно.	Нет	N..12
currency	Валюта платежа. Можно использовать буквенные или цифровые коды валют по стандарту ISO 4217, например, RUB.	Да	N3
order_number	Номер заказа. Если не указывать значение параметра, номер заказа будет сгенерирован автоматически платёжным шлюзом.	Нет	ANS..32
description	Описание заказа. Обычно это наименование товара или товаров.	Нет	ANS..128

Примеры кода приведены ниже:

- [указана точная цена](#);
- [точная цена не указана](#).

### Указана точная цена

```

<a onclick="ipayCheckout({
    amount:499.99,
    currency:'RUB',
    order_number:'',
    description: 'А. С. Пушкин. Избранное (подарочное издание')},
    function(order) { showSuccessfulPurchase(order) },
    function(order) { showFailurefulPurchase(order) }"

```

```
    class="btn btn-xs btn-outline btn-primary">Купить  
  </a>
```

Ссылки ведут на тестовый стенд. По окончании тестирования замените ссылки, чтобы они вели на основной стенд.

## Точная цена не указана

```
<a onclick="ipayCheckout({  
  currency: 'RUB',  
  order_number: '',  
  description: 'Н. В. Гоголь. Вечера на хуторе близ Диканьки'),  
  function(order) { showSuccessfulPurchase(order) },  
  function(order) { showFailurefulPurchase(order) }"  
  class="btn btn-xs btn-outline btn-primary">Купить  
  </a>
```

Ссылки ведут на тестовый стенд. По окончании тестирования замените ссылки, чтобы они вели на основной стенд.

## Проверка завершения оплаты

При использовании платёжной кнопки взаимодействие с платёжным шлюзом происходит незащищённым способом - через элементы встраиваемые в сайт магазина. В таком случае покупатель может изменить свойства платежа, например, сумму. Поэтому важно после завершения оплаты проверить, что проведённый платёж соответствует изначальным требованиям.

### Ручная проверка завершения оплаты

Для ручной проверки оплаты не требуется навыков программирования. Проверить завершение оплаты вручную можно двумя способами:

- проверив параметры фактической оплаты в [личном кабинете](#);
- свериться с письмом-уведомлением, полученным от платёжного шлюза по факту оплаты.

### Автоматическая проверка завершения оплаты

Автоматическая проверка завершения оплаты предполагает проверку цифровой подписи в

[уведомлении обратного вызова на сервере магазина.](#)

Для реализации автоматической проверки завершения оплаты требуются навыки программирования.

После того как плательщик завершит оплату, будет вызвана одна из двух функций: `successCallback(order)` или `failureCallback(order)`, которые вам нужно реализовать для выполнения необходимых действий в связи с оплатой (отправить товар на доставку и т. п.). В параметр вызова функций `successCallback` и `failureCallback` передаётся набор ключевых данных о платеже и подпись, сформированная с использованием закрытого ключа, выданного при регистрации (`secretToken`).

Подпись формируется по следующему алгоритму: значения всех атрибутов объекта `order` соединяются в одну строку в следующем порядке: `status`, `formattedAmount`, `currency`, `approvalCode`, `orderNumber`, `panMasked`, `refNum`, `paymentDate`, `formattedFeeAmount`. Затем в конец добавляется закрытый ключ и от полученной строки вычисляется хэш по алгоритму SHA-256. Полученное значение передаётся в свойстве `digest` объекта `order`.

Для проверки подлинности используется алгоритм HMAC. Полученные данные можно проверить на ресурсах, например: <http://beautifytools.com/hmac-generator.php>

Например, для объекта `order` вида

```
{  
    "approvalCode": "123456",  
    "orderNumber": "310fe4-ec02-4d7f-af04-091f908bf0",  
    "panMasked": "411111**1111",  
    "refNum": "111111111111",  
    "digest": "8f76a4caffaee8f41480ddfc38b2d47edaeb89c298c19cce22a07b9626d612",  
    "currency": "643",  
    "paymentDate": "28.04.2016 20:58:24",  
    "formattedAmount": "500.00",  
    "formattedFeeAmount": "0.00",  
    "status": "DEPOSITED"  
}
```

значение поля `digest` вычислялось по строке

DEPOSITED500.0064312345631840fe4-ec02-4d7f-  
af04-09611f908bf0411111\*\*111111111111112016-04-28  
20:58:240.00j64h6ksafvm200ecpc2l1u4b9k;

где `j64h6ksafvm200ecpc2l1u4b9k` - это пример закрытого ключа.

Никому не сообщайте и не публикуйте закрытый ключ. Проверка подписи должна быть реализована на сервере магазина. Реализация проверки подписи на JavaScript на фронтенде

магазина недопустима.

Проверить факт оплаты можно также с помощью [API платёжного шлюза](#), сделав запрос `getOrderStatusExtended` (через интерфейс [REST](#) или [WSDL](#)).