

Публикация на тему

Платёжка Checkout

Внедряем Checkout на сайты PHP и Laravel

Анотация

Платформа Checkout.com предназначена для ускорения роста бизнеса, изучения более полных и подробных данных и доступа к новым продуктам. В данной публикации я покажу как использовать данную систему для проведения платежей с сайтов, написанных на PHP и Laravel

The Checkout.com platform is designed to accelerate business growth, explore richer, more detailed data, and access new products. In this publication I will show how to use this system to make payments from sites written in PHP and Laravel

Автор

[Михалькевич Александр Викторович](#)

Публикация

Наименование Платёжка Checkout

Автор А.В.Михалькевич

Специальность Внедряем Checkout на сайты PHP и Laravel,

Анотация

Платформа Checkout.com предназначена для ускорения роста бизнеса, изучения более полных и подробных данных и доступа к новым продуктам. В данной публикации я покажу как использовать данную систему для проведения платежей с сайтов, написанных на PHP и Laravel

Anotation in English

The Checkout.com platform is designed to accelerate business growth, explore richer, more detailed data, and access new products. In this publication I will show how to use this system to make payments from sites written in PHP and Laravel

Ключевые слова checkout, php, laravel

Количество символов 9522

Содержание

[Введение](#)

1 [Что такое checkout.com?](#)

2 [Ключевые особенности новой платформы](#)

3 [Быстрая настройка](#)

4 [Тестовый аккаунт и запросы оплаты](#)

[Заклучение](#)

[Список использованных источников](#)

[Приложения](#)

Введение

Если, при прочтении данной публикации у вас возникнут вопросы по терминологии, интеграции и инструментах для управления своей учетной записью, обратитесь к документации [checkout.com](https://www.checkout.com) (<https://www.checkout.com/docs/previous/upgrade-to-our-new-platform>). . Если вам нужна дополнительная поддержка, свяжитесь с нашей службой поддержки по адресу Platformupgrade@checkout.com.

1 Что такое checkout.com?

Checkout.com предлагает глобальную платформу, прозрачное ценообразование и решения в области платежей.



2 Ключевые особенности новой платформы

После установки или обновления данной платформы, мы получаем доступ к подробной финансовой отчётности и широкому спектру инструментов бизнес-аналитики, в том числе:

- API-интерфейсы отчетов и финансовых действий - [Reports and Financial actions APIs](#);
- Статистика, которая поможет анализировать производительность - [Dashboard](#).

Особенности платформы Checkout.com

- обновленный набор webhook-ов, которые позволяют отслеживать события на протяжении всего жизненного цикла платежа - [webhooks](#).
- полная видимость комиссий, взимаемых каждый месяц, с помощью единого счета в формате PDF для каждой валюты - [invoice](#).
- отчет о балансе, чтобы понять притоки и оттоки валюты - [balance report](#).

3 Быстрая настройка

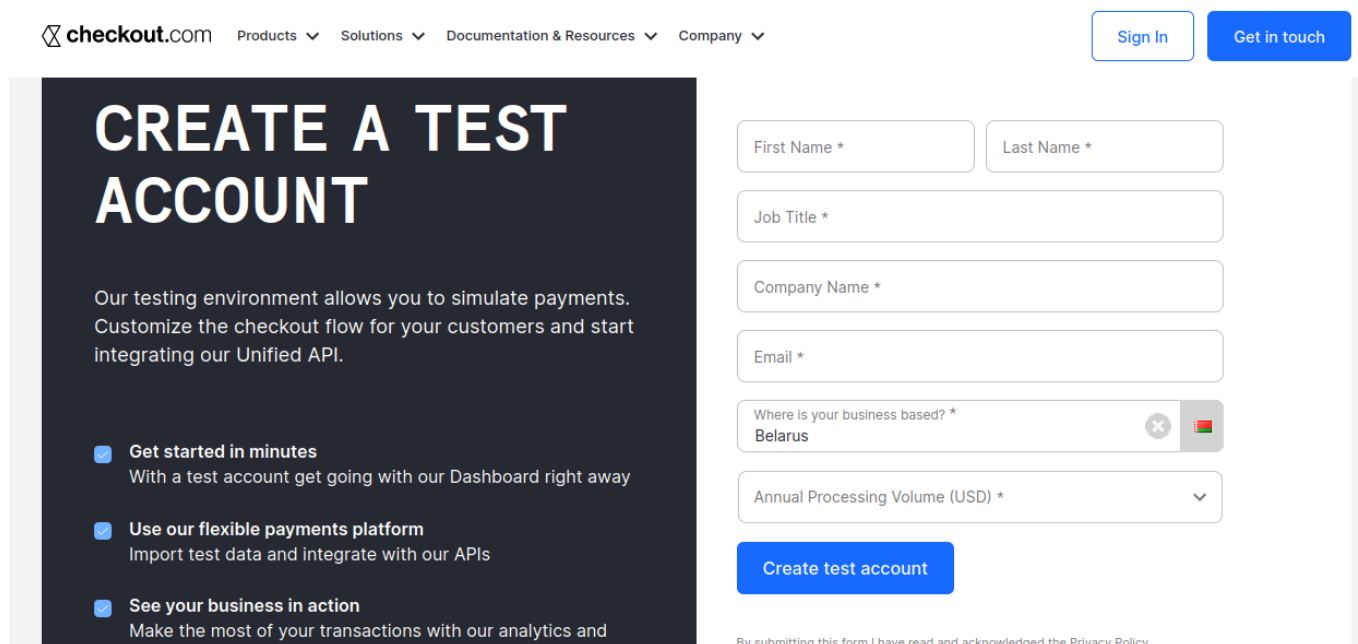
С помощью конфигурации на уровне клиента вы можете обновить определенные настройки для всей учетной записи:

- добавлять, обновлять или отзывать доступ пользователей ко всем вашим объектам одновременно - [user access](#);
- блокировать обнаружение мошенничества (риск) и изменять правила для всей учетной записи без необходимости настраивать это для каждого направления бизнеса - [Fraud Detection](#).

4 Тестовый аккаунт и запросы оплаты

Прежде чем приступать к разработке и подключению платежной системы, лучше создать тестовый аккаунт и протестировать оплату.

Зарегистрировать тестовый аккаунт можно по ссылке - <https://www.checkout.com/get-test-account> (регистрация для жителей России и Беларуси доступна)



После регистрации тестового или реального (live) аккаунта переходим по ссылке <https://dashboard.checkout.com>

Для входа вводим указанный при регистрации email, потом пароль. После авторизации происходит перенаправление на страницу <https://dashboard.sandbox.checkout.com>. Далее переходим по ссылке Developers (<https://dashboard.sandbox.checkout.com/developers/get-started>). На этой странице после генерации ключей и получения токена, получим тестовый номер карты Visa и пример request-запроса оплаты.

Итак, после того, как получим `public_key`, `secret_key`, `token` и данные по Visa, система сгенерирует **Request payment**. Пример запроса:

```
curl -i -X POST \
  https://api.sandbox.checkout.com/payments \
  -H 'Authorization: Bearer Значение_Bearer' \
```

```

-H 'Content-Type: application/json' \
-d '{
  "source": {
    "type": "token",
    "token": "Значение_token"
  },
  "amount": 6540,
  "currency": "USD",
  "processing_channel_id": "pc_значение_pc",
  "reference": "get started guide",
  "description": "Set of 3 masks",
  "capture": true,
  "customer": {
    "email": "brucewayne@gmail.com",
    "name": "Bruce Wayne",
    "phone": {
      "country_code": "+1",
      "number": "415 555 2671"
    }
  },
  "shipping": {
    "address": {
      "address_line1": "Checkout.com",
      "address_line2": "90 Tottenham Court Road",
      "city": "London",
      "state": "London",
      "zip": "W1T 4TJ",
      "country": "GB"
    },
    "phone": {
      "country_code": "+1",
      "number": "415 555 2671"
    }
  },
  "3ds": {
    "enabled": false,
  },
  "success_url": "http://example.com/payments/success",
  "failure_url": "http://example.com/payments/fail",
  "payment_ip": "значение_ip",
  "metadata": {
    "coupon_code": "NY2018",
    "partner_id": 123989
  }
}'

```

Если вы используете Laravel, то в экшне оплаты, для отправки запроса можем использовать фасад Http. Пример:

```

$response = Http::withHeaders([
  'authorization' => $this->credentials['secret_key'],

```

```

])->post('https://api.checkout.com/hosted-payments', [
    'amount' => (float) $amount * 100,
    'currency' => 'USD',
    'reference' => $transactionNumber,
    'description' => 'Subscription To receive services of ' .
config('app.url'),
    'billing' => [
        'address' => [
            'country' =>strtoupper(optional(auth()->user()->country) ?? 'UA',
        ],
    ],
    'customer' => [
        'name' => auth()->user()->name,
        'email' => auth()->user()->email,
    ],
    'success_url' => url('/dashboard'),
    'failure_url' => url('/price'),
    'cancel_url' => url('/price'),
]);
$responseData = $response->json();
dd($responseData);

```

Помимо маршрута оплаты также необходимо реализовать еще 2 маршрута: success_url (при успешной оплате происходит перенаправление на этот url) и failure_url (для перехвата ошибок)

Заключение

Список использованных источников

Приложения