

Интеграция внешних платежных API с WooCommerce



- с чего начать разработку шлюза;
- с какими трудностями можно столкнуться при интеграции;
- о видах платежных шлюзов;
- как использовать токенизацию;
- что такое рекуррентные платежи, как их реализовать в WooCommerce;
- о различиях работы провайдеров с рекуррентными платежами.

Типы платежных шлюзов

Offline



<iframe> based



Form based



Direct



```
1 <?php
2
3 /**
4  * Plugin Name: Custom payment gateway fo
  Woocommerce
5  * Description: WooCommerce Payment Gateway for
  WordCamp Saint-Petersburg 2019
6  * Author: Anton Drobyshev
7  * Version: 1.0
8  */
9
10 // Не дадим открыть файл напрямую
11 if ( ! defined( 'ABSPATH' ) )
12     exit;
13
```

```
13
14 // Проверим, активирован ли WooCommerce
15 if ( in_array( 'woocommerce/woocommerce.php',
16     apply_filters( 'active_plugins', get_option( 'active_plugins' ) ) ) ) {
17
18     // Определим константы, которые могут пригодиться в процессе
19     define( 'AD_VERSION', '1.0' );
20     define( 'AD_PLUGIN_DIR', plugin_dir_path( __FILE__ ) );
21     define( 'AD_PLUGIN_URL', plugin_dir_url( __FILE__ ) );
22
23     // Добавим в массив доступных методов оплаты наш кастомный шлюз
24     add_filter( 'woocommerce_payment_gateways', 'ad_gateway_class' );
25     function ad_gateway_class( $methods ) {
26         $methods[] = 'WC_Custom_Gateway';
27         return $methods;
28     }
29
30     // Подключим файл с классом платежного шлюза
31     add_action( 'plugins_loaded', 'ad_init_wc_custom_gateway' );
32     function ad_init_wc_custom_gateway () {
33         require_once( AD_PLUGIN_DIR . 'class-custom-gateway.php' );
34     }
35
36 }
```

Стандартные платежные шлюзы

WC_Gateway_BACS

Прямой банковский перевод



WC_Gateway_COD

Оплата при доставке



WC_Gateway_Cheque

Чековые платежи

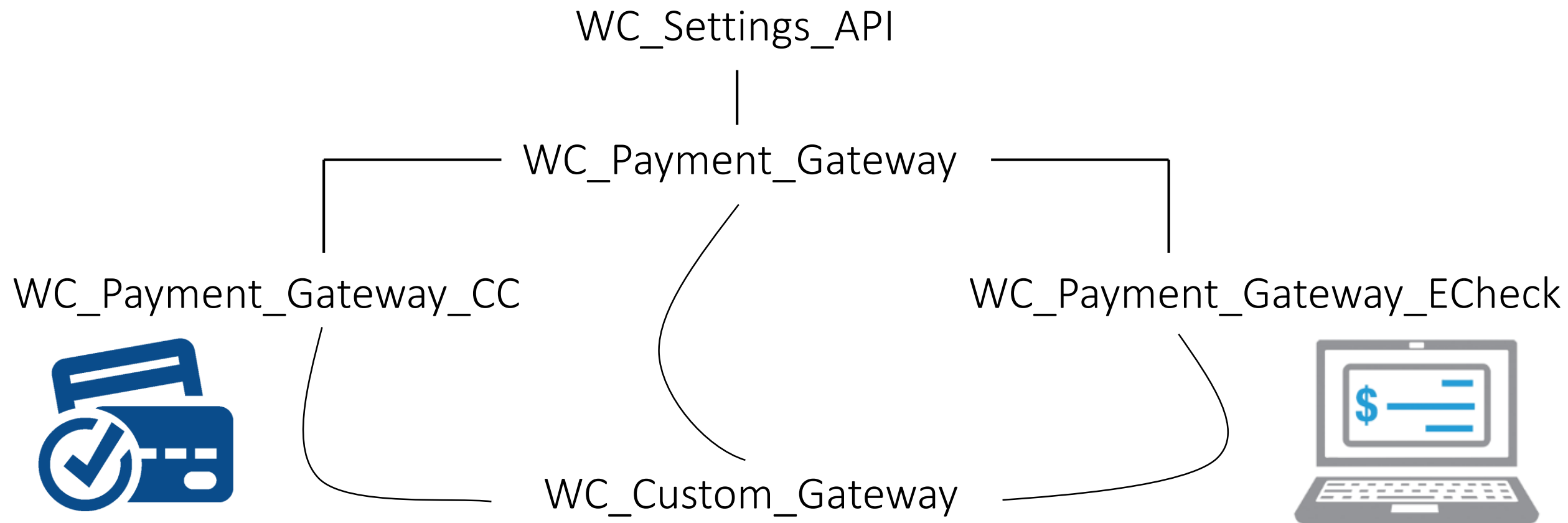


WC_Gateway_Paypal

Оплата через PayPal



Пишем класс шлюза



WC_Payment_Gateway_CC

WC_Payment_Gateway_ECheck

Card number *

Routing number *

Expiry (MM/YY) *

Card code *

Account number *


```
3 class WC_Custom_Gateway extends WC_Payment_Gateway_CC {
4
5     public function __construct () {
6         // Уникальный ID платежного шлюза
7         $this->id = 'custom_gateway';
8         // Иконка платежного шлюза рядом с его названием
9         $this->icon = AD__PLUGIN_URL . 'assets/images/visa-mastercard.png';
10        // Отобразить поля ввода банковской карты
11        $this->has_fields = true;
12        // Название и описание шлюза на странице настроек
13        $this->method_title = 'Custom Pay';
14        $this->method_description = 'Custom Pay Gateway for WordCamp Saint-Petersburg
15        2019';
16        // Какой функционал поддерживает наш платежный шлюз
17        $this->supports = [
18            'products', // Поддержка покупки товаров
19            'subscriptions', // Поддержка покупки подписки
20            'refunds', // Поддержка возврата денег
21            'tokenization' // Поддержка токенизации
22        ];
23    }
24 }
```

```
22
23 // Метод, в котором мы далее распишем опции страницы настроек
24 $this->init_form_fields();
25 // Данный метод достает из бд все сохраненные настройки нашего шлюза
26 $this->init_settings();
27 // Включить или выключить шлюз
28 $this->enabled = $this->get_option( 'enabled' );
29 // Название шлюза, которое отобразим в чекауте
30 $this->title = $this->get_option( 'title' );
31 // Описание шлюза, которое отобразим в чекауте
32 $this->description = $this->get_option( 'description' );
33 // Тестовый режим: да или нет
34 $this->testmode = 'yes' === $this->get_option( 'testmode' );
35 // Секретный ключ
36 $this->private_key = $this->get_option( 'private_key' );
37 // Публичный ключ
38 $this->public_key = $this->get_option( 'public_key' );
39 // Сохраним введенные данные на странице настроек
40 add_action( 'woocommerce_update_options_payment_gateways_' . $this->id, array( $this, '
    process_admin_options' ) );
```

```
/**
 * Определим поля на странице настроек шлюза
 */

public function init_form_fields() {

    $this->form_fields = array(
        'enabled' => array(
            'title'      => 'Включить / Выключить',
            'label'      => 'Включить Custom Pay',
            'type'       => 'checkbox'
        ),
        'title' => array(
            'title'      => 'Название',
            'type'       => 'text',
            'description' => 'Данное название появится на странице чекаута.',
            'default'    => 'Visa / Mastercard',
            'desc_tip'   => true,
        ),
    ),
```

Методы нашего класса

Валидация формы

validate_fields()

```
if ( empty( $_POST['credit_card'] ) ) {  
    wc_add_notice( 'Мне нужна ваша карта!', 'error' );  
    return false;  
}  
  
return true;
```

Списываем деньги

process_payment()

```
// Редирект на страницу успешного заказа  
return array(  
    'result' => 'success',  
    'redirect' => $this->get_return_url( $order )  
);
```

Возвращаем деньги

process_refund()

```
if ( $response['response']['code'] === 200 ) {  
    return true;  
} else {  
    return false;  
}
```

```
// Собираем данные о заказе и карте
```

```
$param = array(  
    'posEntryMode' => '012',  
    'pan' => $_POST['epay_ccNo'],  
    'expdate' => $expdate,  
    'amount' => $amount,  
    'track2Data' => '',  
    'cvv2' => $_POST['epay_cvv'],  
    'paymentgwIP' => '190.149.69.135',  
    'shopperIP' => antondrob_get_user_ip(),  
    'merchantServerIP' => antondrob_get_merchant_ip(),  
    'messageType' => '0200',  
    'additionalData' => ''  
);
```

```
// Soap запрос
```

```
$params = array( array( 'AuthorizationRequest' => $param ) );  
ini_set( 'default_socket_timeout', $this->timeout );  
$client = new SoapClient( $this->url, array( 'connection_timeout' => $this->timeout ) );  
try {  
    $result = $client->__soapCall( 'AuthorizationRequest', $params );
```

process_payment()

process_payment()

```
$order->payment_complete();  
$order->add_order_note( 'Деньги пришли!', false );  
WC()->cart->empty_cart();  
return array(  
    'result'    => 'success',  
    'redirect' => $this->get_return_url( $order )  
);
```

process_refund()

```
383
384 // Собираем данные о платеже
385 $param = array(
386     'posEntryMode' => '012',
387     'paymentgwIP' => '190.149.69.135',
388     'shopperIP' => antondrob_get_user_ip(),
389     'merchantServerIP' => antondrob_get_merchant_ip(),
390     'messageType' => '0202',
391     'additionalData' => '',
392     'auditNumber' => get_post_meta( $order_id, 'auditNumber', true )
393 );
394 // Soap запрос
395 $params = array( array( 'AuthorizationRequest' => $param ) );
396 ini_set( 'default_socket_timeout', $this->timeout );
397 $client = new SoapClient( $this->url, array( 'connection_timeout' => $this->timeout ) );
398 try {
399     $result = $client->__soapCall( 'AuthorizationRequest', $params );
400
```

process_refund()

```
// Проверяем ответ
if ( $result->response->responseCode === '00' ) {
    $order->add_order_note( 'Деньги ушли!', false );
    return true;
} else {
    $order->add_order_note( 'Деньги не ушли. Код ошибки: ' . $result->response->
        responseCode, false );
    return false;
}
```


Токенизация в WooCommerce

Dashboard 


Orders 

Downloads 

Addresses 

Payment methods 

Account details 

Logout 

Edit

Method	Expires	
Visa ending in 1234	12/18	Delete

[Add payment method](#)

Токенизация в WooCommerce

Прямой банковский перевод

Чековые платежи

Оплата при доставке

Visa / Mastercard




Visa ending in 1234 (expires 12/18)

Use a new payment method

Токенизация в WooCommerce

1. Подключим поддержку токенизации.
2. Добавим метод `add_payment_method()` нашему классу

```
// Какой функционал поддерживает наш платежный шлюз
$this->supports = [
    'products', // Поддержка покупки товаров
    'subscriptions', // Поддержка покупки подписки
    'refunds', // Поддержка возврата денег
    'tokenization' // Поддержка токенизации
];
```



```
public function add_payment_method () {

    $token = new WC_Payment_Token_CC();
    $token->set_token( $token_string );
    $token->set_gateway_id( $this->id );
    $token->set_card_type( 'visa' );
    $token->set_last4( '1234' );
    $token->set_expiry_month( '12' );
    $token->set_expiry_year( '2018' );
    $token->set_user_id( get_current_user_id() );
    $token->save();

}
```

Токенизация в WooCommerce

3. Получим токен и сделаем необходимые проверки.

```
if ( isset( $_POST['wc-simplify_commerce-payment-token'] ) &&
    'new' !== $_POST['wc-simplify_commerce-payment-token'] ) {
    // Получим токен
    $token_id = wc_clean( $_POST['wc-simplify_commerce-payment-token'] );
    $token     = WC_Payment_Tokens::get( $token_id );
    // Проверим является ли пользователь владельцем токена
    if ( $token->get_user_id() !== get_current_user_id() ) {
        wc_add_notice( 'No no no...', 'error' );
        return;
    }
}
```

Рекуррентные платежи в WooCommerce



```
// Какой функционал поддерживает наш платежный шлюз
$this->supports = [
    'products', // Поддержка покупки товаров
    'refunds', // Поддержка возврата денег
    'tokenization', // Поддержка токенизации
    'subscriptions', // Поддержка покупки подписки
    'subscription_cancellation', // Аннулирование
    'subscription_suspension', // Приостановка
    'subscription_reactivation', // Возобновление
    'subscription_amount_changes', // Изменение суммы
    'subscription_date_changes', // Изменение даты
    'subscription_payment_method_change', // Изменение метода оплаты
    'multiple_subscriptions', // Возможность покупки нескольких
    подписок в одном заказе
];

// Вешаем событие на крон
add_action('woocommerce_scheduled_subscription_payment_' . $this->id,
    array($this, 'scheduled_subscription_payment'), 10, 2);
```

Но...что если нет токенов?

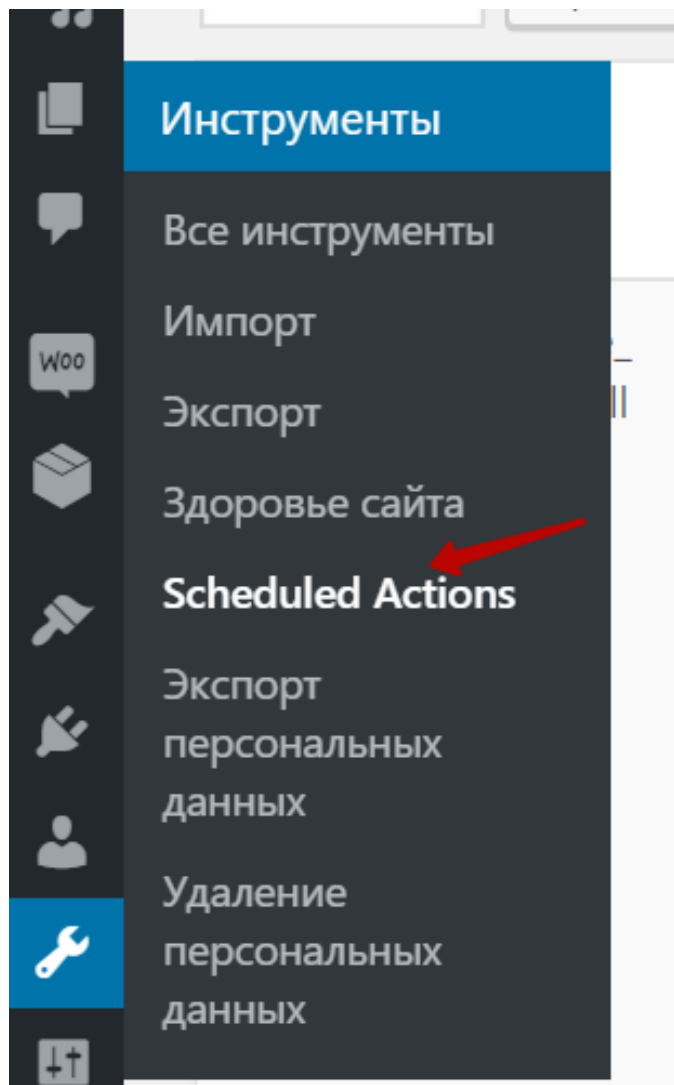
Есть IPN (Instant Payment Notification)? Есть callback!

```
// В конструкторе класса добавим callback
add_action( 'woocommerce_api_callback', array( $this, 'ad_callback_handler' ) );

// Метод обработчик callback
public function ad_callback_handler () {
    // Обработка сигнала от провайдера...
}
```

Тестируем рекуррентный платеж

1.



2.

Scheduled Actions

All (19) | Complete (11) | Pending (6) | Canceled (2)

Действия ▾

Применить



Hook

Status

Arguments

3.



woocommerce_scheduled_subscription_payment

Pending

```
'subscription_id'  
=> 71
```

Run | Cancel



Hook

Status

Arguments

G

Действия ▾

Применить

The end

