



---

# CODEIGNITER PAYPAL PAYMENT INTEGRATION

---

KH NOUSHADUR RAHMAN



JUNE 15, 2019  
VERSION 1.0

## Contents

1. Pre-requisites: .....	2
2. Prepare your developer account and get the client ID and the secret ID .....	2
3. Integrating the library to your Codeigniter. ....	8
4. Available Functions .....	9
5. Sample Code .....	10
5.1 For a recurring payment (Subscription) .....	10
5.2 For a PayPal checkout.....	12
6. Add money to your sandbox account.....	14

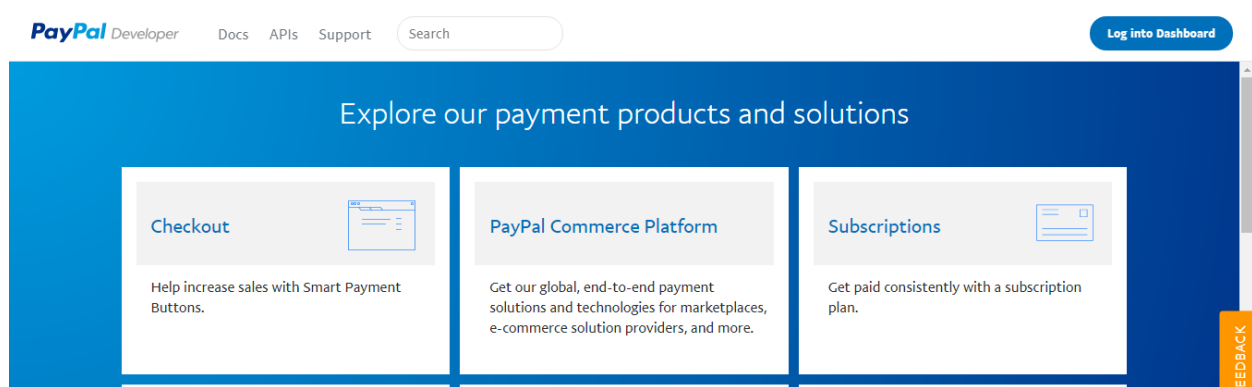
## 1. Pre-requisites:

1. A PayPal account (sandbox [<https://www.sandbox.paypal.com>] or live).
2. A PayPal developer account.
3. Client ID and Secret ID for a particular app.

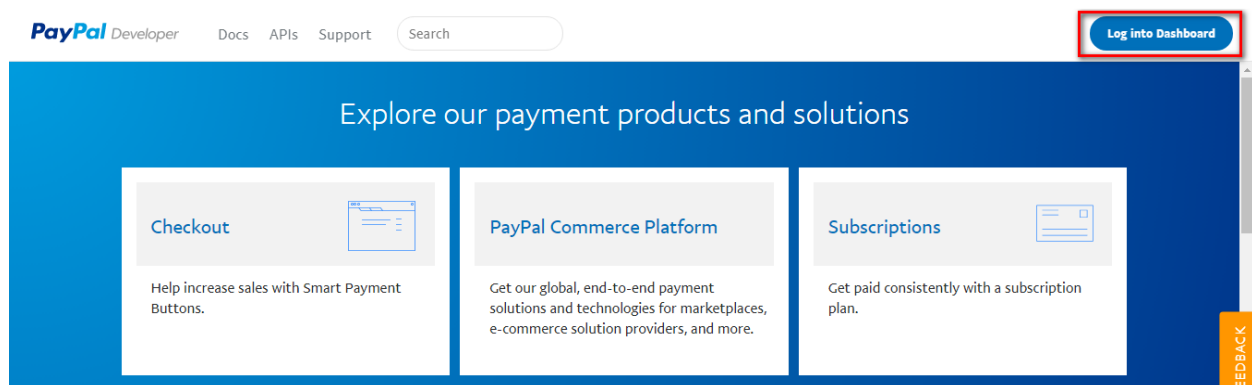
***Note:** It's always recommended to use the PayPal sandbox account (a demo PayPal account) at initial stage for testing purpose.*

## 2. Prepare your developer account and get the client ID and the secret ID

**Step 1:** Go to the <https://developer.paypal.com/>



**Step 2:** Click the button “Log into Dashboard” and log into the dashboard using your credentials.



My Account

SANDBOX

**Accounts**

Notifications

API Calls

IPN Simulator

Webhooks Events

To link your sandbox account to your  
credentials.

[Create Account](#)

ount, [log in with PayPal](#) and provide your sandbox account

CODEIGNITER PAYPAL PAYMENT INTEGRATION

**Step 5:** Provide the all information for creating a merchant (business) account. Here, for account type, select the option “Business (Merchant Account)”. And the email section you may use your original email account or an arbitrary email address. And provide an arbitrary amount in PayPal Balance field.

### Account Type

☒ Personal (Buyer Account)

☐ Business (Merchant Account)

### Email Address

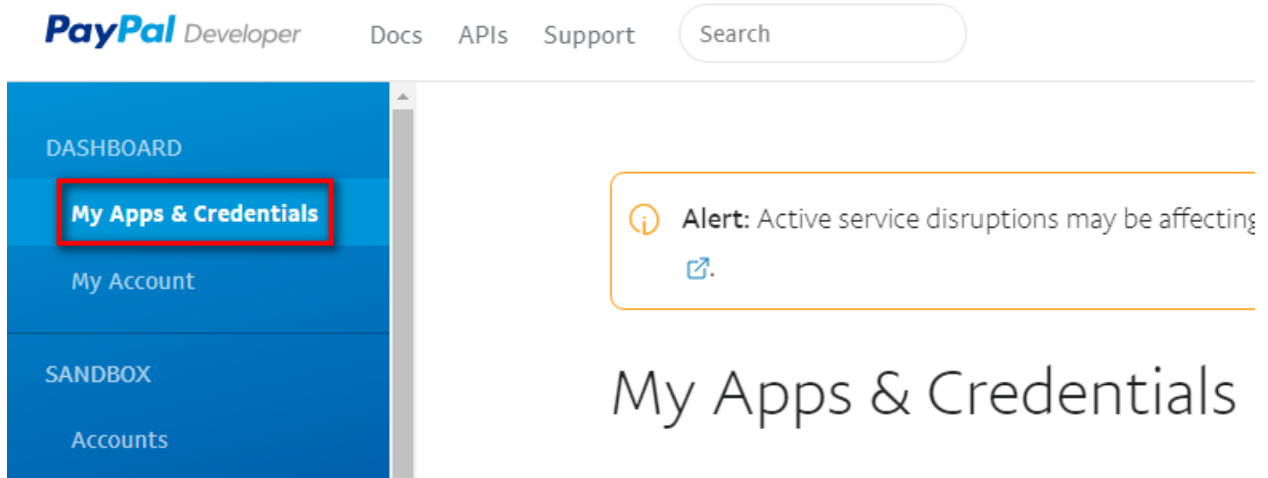
### Payment Methods

Provide an arbitrary  
value here

PayPal Balance

.00 USD

**Step 6:** Click the menu “My Apps & Credentials”.



**Step 7:** Click the button “Create App”.

## REST API apps

Create an app to receive REST API credentials for testing and live transactions.

**App Name**

**Type**



## CODEIGNITER PAYPAL PAYMENT INTEGRATION

**Step 8:** Provide an app name and select a sandbox business account (if an account has been already created successfully then it will be listed here). Next, click the create app button.

Application Details

App Name

Sandbox Business Account

brbsns@email.com (US) ▼

As a reminder, all apps created under your account should be related to your business and the type of business it conducts.


By clicking the button below, you agree to [PayPal Developer Agreement](#).

Create App

**Step 9:** After creating the app you will be redirected to the “My apps and Credentials” menu. Click the name of the app you have created just now. For an example, here my app name is “br-dev”.

## REST API apps

Create an app to receive REST API credentials for testing and live transactions.

App Name	Type	Actions
br-dev	REST	
<div>Create App</div>		


CODEIGNITER PAYPAL PAYMENT INTEGRATION

**Step 10:** After clicking you will get your sandbox client ID. For live, click the Live and follow those procedure.

br-dev

Sandbox

Live

App display name: br-dev 

SANDBOX API CREDENTIALS

Sandbox account

brbsns@email.com

Client ID

Secret

[Show](#)

**Step 11:** Click the “Show” link under the secret section, you will get the client secret ID.

Secret

[Show](#)

Click here  
to get the  
client secret  
ID

Secret

[Hide](#)

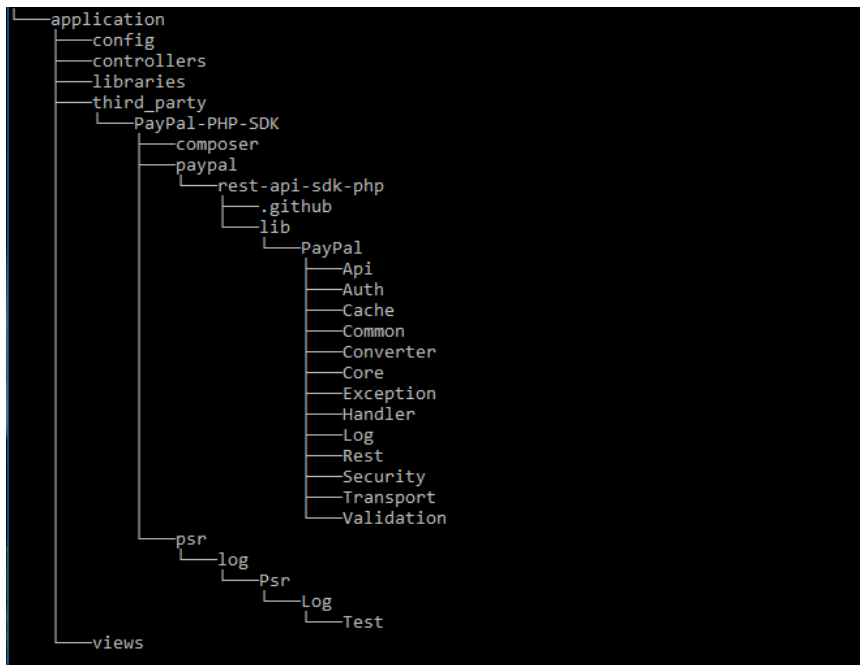
**Note:** When you generate a new secret, you still maintain the original secret. The maximum number of client secrets is two. A client secret is either in enabled or disabled state.

Created	Secret	Status	Action
May 15, 2019		Enabled	

[Generate New Secret](#)

### 3. Integrating the library to your Codeigniter.

**Step 1:** Merge the collected application folder to your Codeigniter folder or just simply copy the application folder and paste it to your Codeigniter project. The PayPal integration application folder tree looks like this:-



**Step 2:** Go to the folder “**application/config**” and open up the file **paypal.php**. Here, you will provide your client ID, secret ID, mode (sandbox or live) and the currency code you want to use. To show the list of currency code, follow the link -

<https://developer.paypal.com/docs/api/reference/currency-codes/>

```
1. $config['mode']           = "sandbox"; //for live account use the key "live"
2. $config['client_id']      = '[Your client ID]';
3. $config['client_secret'] = '[Your secret ID]';
4. $config['currency']      = 'USD';
```

**Note:** All are mandatory.

Done, your library integration successfully completed. To check it, please route to the Payment controller. For an example, if you use localhost then go to:

[http://localhost/\[yourProjectFolder\]/index.php/payment](http://localhost/[yourProjectFolder]/index.php/payment)

## 4. Available Functions

All available functions you will get in “application/libraries/Paypal.php”. And everything is well described at the beginning or inside of the each function like the code below:-

```
1.  /* Creates a billing plan. In the JSON request body, include the plan details. A plan must include at least one regular payment definition and, optionally, a trial payment definition. Each payment definition specifies a billing period, which determines how often and for how long the customer is charged. A plan can specify a fixed or infinite number of payment cycles. A payment definition can optionally specify shipping fee and tax amounts. The default state of a new plan is CREATED. Before you can create an agreement from a plan, you must activate the plan by updating its state to ACTIVE.
2.
3.  name string required
4.  The plan name.
5.  Maximum length: 128.
6.
7.  description string required
8.  The plan description. Maximum length is 127 single-byte alphanumeric characters.
9.  Maximum length: 127.
10.
11. type enum required
12. The plan type. Indicates whether the payment definitions in the plan have a fixed number of or infinite payment cycles. Value is:
13. FIXED. The plan has a fixed number of payment cycles.
14. INFINITE. The plan has infinite, or 0, payment cycles.
15. Allowed values: FIXED, INFINITE.
16.
17. Maximum length: 20.
18. */
19.
20. public function set_plan( $plan_name, $plan_description, $plan_type ){
21.
22.     $this->plan = new Plan();
23.     $this->plan->setName($plan_name)
24.         ->setDescription($plan_description)
25.         ->setType($plan_type);
26.
27.     return;
28.
29. }
```

## 5. Sample Code

### 5.1 For a recurring payment (Subscription)

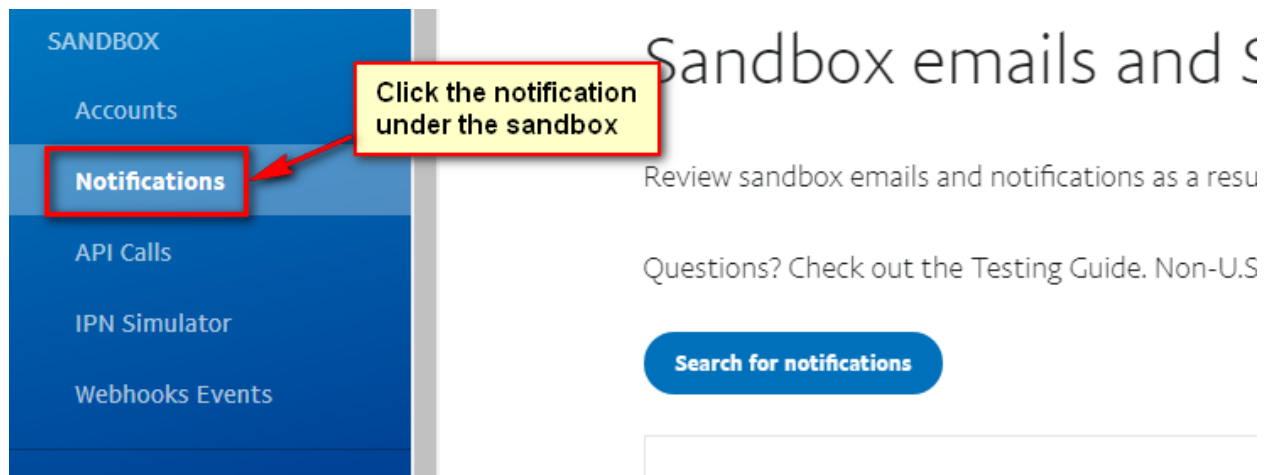
Create a controller named **Payment.php**

```
1. <?php
2. class Payment extends CI_Controller {
3.
4.     public function __construct(){
5.         parent::__construct();
6.         $this->load->library("paypal");
7.     }
8.
9.     public function subscribe(){
10.
11.         if ( !empty($_POST["plan_name"]) && !empty($_POST["plan_description"]) ) {
12.
13.             $this->paypal->set_api_context();
14.
15.             $this->paypal-
16. >set_plan( $_POST["plan_name"], $_POST["plan_description"], "INFINITE" );
17.
18.             $definition = "Regular Payments";
19.             $type = "REGULAR";
20.             $frequency = "MONTH";
21.             $frequency_interval = '1';
22.             $cycles = 0;
23.             $price = "49";
24.
25.             $this->paypal-
26. >set_billing_plan_definition( $definition, $type, $frequency, $frequency_interval, $cycles, $price );
27.
28.             $returnurl = base_url()."index.php/payment/success";
29.             $cancelurl = base_url()."index.php/payment/cancel";
30.
31.             $this->paypal->set_merchant_preferences( $returnurl, $cancelurl );
32.
33.             $line1 = "Street - 1, Sector - 1";
34.             $city = "Dhaka";
35.             $state = "Dhaka";
36.             $postalcode = "12345";
37.             $country = "AU";
38.
39.             $this->paypal-
40. >set_shipping_address( $line1, $city, $state, $postalcode, $country );
41.
42.             $agreement_name = "Payment Agreement Name";
43.             $agreement_description = "Payment Agreement Description";
44.
45.             $this->paypal-
46. >create_and_activate_billing_plan( $agreement_name, $agreement_description );
47.         }
```

## CODEIGNITER PAYPAL PAYMENT INTEGRATION

```
48. //After successfully create an agreement we will be redirected to this function
49. public function success(){
50.
51.     if ( !empty($_GET['token']) ) {
52.
53.         $token = $_GET['token'];
54.         $this->paypal->execute_agreement( $token );
55.
56.     }
57.
58.     return;
59.
60. }
61. }
```

After successfully executing all of those you will be notified on your account. To check notification on developer account follow these steps:-



You will find a notification like this

To	Subject	Date
brbsns@email.com	You have a new automatic payment profile for Khandakar Noushadur Rahman	11 Jun 2019 13:26:44

## 5.2 For a PayPal checkout


Create a controller named **Payment.php**

```
1. <?php
2. class Payment extends CI_Controller {
3.
4.     public function __construct(){
5.         parent::__construct();
6.         $this->load->library("paypal");
7.     }
8.
9.     public function checkout_payment(){
10.
11.         $this->paypal->set_api_context();
12.
13.         $payment_method = "paypal";
14.         $return_url      = base_url()."index.php/payment/success_payment";
15.         $cancel_url      = base_url()."index.php/payment/cancel";
16.         $total           = 10;
17.         $description     = "Paypal product payment";
18.         $intent          = "sale";
19.
20.         $this->paypal->create_payment( $payment_method, $return_url, $cancel_url,
21.         $total, $description, $intent );
22.
23.         return;
24.
25.     }
26.
27.     //After creating a payment successfully we will be redirected here
28.     public function success_payment(){
29.
30.         if ( !empty( $_GET['paymentId'] ) && !empty( $_GET['PayerID'] ) ) {
31.
32.             $this->paypal->execute_payment( $_GET['paymentId'], $_GET['PayerID'] );
33.
34.         }
35.
36.         return;
37.
38.     }
39. }
40.
```

To check it, go to your PayPal account (sandbox or live). After logging in you will observe that your balance have been subtracted by the price you provided in code and a notification have been arrived in recent activity. Here is the example of a sandbox account.

## Good morning, Khandakar Noushadur

**PayPal balance**




**\$915.62**

Available

[Transfer Money](#)

**Recent activity**



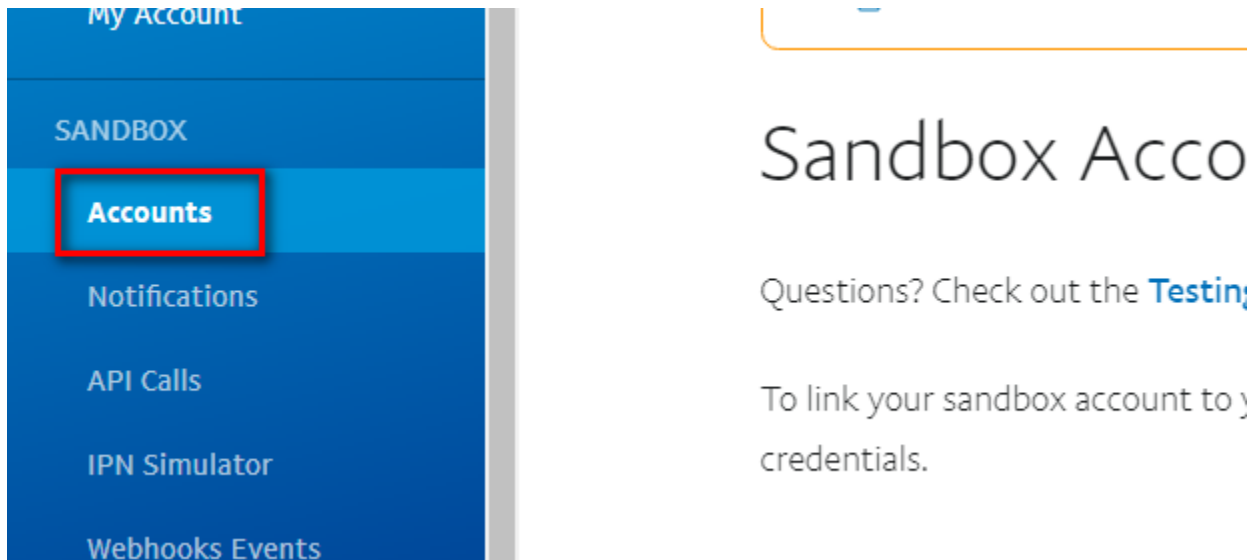
JUN 15	Noushadur Rahman's Test Store Preapproved Payment	-\$4.00
<hr/>		
JUN 15	Noushadur Rahman's Test Store <span>PENDING</span> Preapproved Payment	-\$49.00 AUD
<hr/>		

And if you observe your merchant account profile then you will see the balance has been increased.

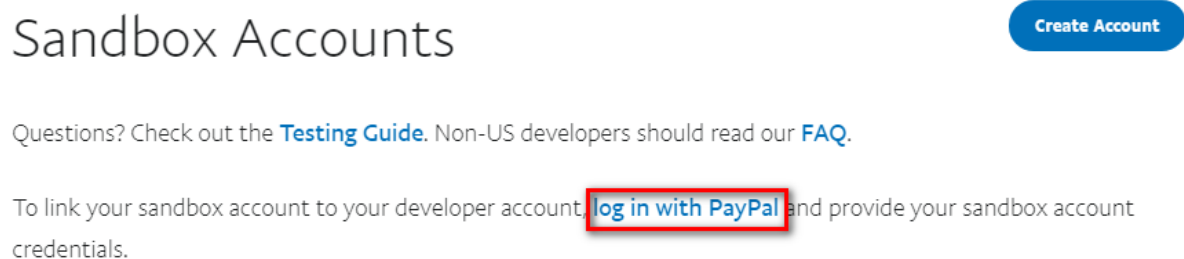
## 6. Add money to your sandbox account

**Step 1:** Go to your developer account (<https://developer.paypal.com/>)

**Step 2:** Click the Accounts menu under the sandbox section.



**Step 3:** Click the link “log in with PayPal” and complete the further procedure.



After that, you will see your sandbox account has been listed successfully.

**Step 4:** Click on the account and click the profile link.

<input type="checkbox"/>	Email Address	Type	Country	Date Created	Status	Actions
<input type="checkbox"/>	kh.noushadur@gmail.com	PERSONAL	US	10 Jun 2019	complete	...
	<a href="#">Profile</a> <a href="#">Notifications</a>					

**Step 5:** Click the funding tab

Account Details

Profile **Funding** Settings

First Name:  
Khandakar Noushadur

**Step 6:** Click the edit button and enter the amount you want to add then save it.

Profile Funding **Settings**

Credit Card

Credit Card Number:  
49290000000006

Credit Card Type:  
VISA

Expiration Date:  
04/2021

PayPal

Balance:  
915.62 USD

**Edit** Close

**Step 7:** Go to your sandbox account (<https://www.sandbox.paypal.com>) and check the balance.

**To know more about PayPal API:**

<https://developer.paypal.com/docs/api/overview/>