Setup Guide

Product by **LivelyWorks**

Table of Contents

Installation

Files & Directories in Download Package

In the downloaded **Zip** file, you will find below files & folders:

- Source – a folder containing all the required files you need to paste in your application.

- **config.php** - a file containing all payment gateways configuration, you need to update payment gateways API and secret key.

- Setup-Guide.pdf – an Instruction manual for setup (This Document).

- Quick-Start-Guide.pdf - Basic introduction.

- Technical-Guide.pdf – Technical information may help for further customizations.

Configurations

config.php file contains all the configuration/settings for the payment gateways eg. like enable payment gateway, payment API key, Secret key, Merchant key etc. you need to update these key as per your requirement.

You will find this file in **source/example** folder.

First, you need to set base URL as per showing below.

```
/* Base Path of app

------------------------------------------------------------------ */

'base_url' =>  'Your base path to example folder exa:- hostname/example/',
```

Payment gateways can be **on** or **off** using **enable** property using gateway configurations.

```
'payments' => [
    /* Gateway Configuration key
    ------------------------------------------------------------------ */
    'gateway_configuration' => [
        'paypal' => [
            'enable'                        => false,
            'testMode'                      => true, //test mode or product mode (boolean, true or false)
            'gateway'                       => 'Paypal', //payment gateway name
            'itemName'                      => 'Test', //item name
            'paypalSandboxBusinessEmail'        => 'paypaltestbusiness@livelyworks.net', //paypal sandbox business email
            'paypalProductionBusinessEmail'     => 'Enter Your Product business email', //paypal production business email
            'orderID'               => 'ORDS' . uniqid(), //generate random order id
```

Other files in example uses

1. **index.php -** All enabled payment gateways are shown under this file. This file is containing PHP, jQuery (JavaScript) code, you will easily modify this file as per your need.

   > **Important:** Please note one should provide user details as given in $userDetails array.

2. **payment-process.php -** All these payments process through this file. It sanitize and validate user input data and if valid data are found then send this data to payment gateway service class for the processing/authorizing which further may communicate with payment gateways API.

3. **payment-response.php -** You will get all payment responses here in $requestData variable.
   Using 'paymentOption' key you can identify that which payment gateway belongs to current response data.

```php
// Get Config Data
$configData = configItem();
// Get Request Data when payment success or failed
$requestData = $_REQUEST;


// Check payment Method is paytm
if ($requestData['paymentOption'] == 'paytm') {
    // Get Payment Response instance
    $paytmResponse  = new PaytmResponse();
```

In this file, you will find paymentResponse() function. Under this function you will get payment success / failed status in **$paymentResponseData** variable. You can send an email, insert to database or redirect to any page from here.

```php
function paymentResponse($paymentResponseData) {

    // payment status success

    if ($paymentResponseData['status']) {


        // Show payment success page or do whatever you want, like send email, notify to user etc

        header('Location: '. getAppUrl('payment-success.php'));


    } else {

        // Show payment error page or do whatever you want, like send email, notify to user etc

        header('Location: '. getAppUrl('payment-failed.php'));

    }
```

4. **payment-success.php -** After successful payment, you will redirect here. You can show messages to the user or do anything after successful payment.

```html
<div class="text-center mt-5">
    <div class="col-lg-12 text-center">
        <!-- Thanks message -->
        <h3>Thanks for your payment</h3>
        <!-- Success Icon -->
        <i class="fa fa-check-square-o fa-5x text-success"></i>
        <!-- /Success Icon -->
        <h1>Payment Successfully</h1>
        <!-- /Thanks message -->
        <!-- URL for back to checkout form -->
        <a href="<?= getAppUrl() ?>" title="Back to Checkout Form">Back to Checkout Form</a>
        <!-- /URL for back to checkout form -->
    </div>
</div>
```

5. **payment-failed.php -** If payment is not completed or it get failed, then you will be redirected here. You can show error messages or do anything after an unsuccessful payment is made.

```html
<div class="text-center mt-5">
    <div class="col-lg-12 text-center">
        <!-- Error message -->
        <h3>Sorry! Payment Cancel</h3>
        <!-- /Error Icon -->
        <i class="fa fa-exclamation-circle fa-5x text-danger"></i>
        <!-- /Error Icon -->
        <h1>Payment Failed</h1>
        <!-- /Error message -->
        <!-- URL for back to checkout form -->
        <a href="<?= getAppUrl() ?>" title="Back to Checkout Form">Back to Checkout Form</a>
        <!-- /URL for back to checkout form -->
    </div>
</div>
```

6. **payment-pending.php** - If payment status is not completed / cancelled or pending then, then user will redirect here. You can show message or do anything after payment is in pending status.

```html
<body>
    <div class="text-center mt-5">
        <div class="col-lg-12 text-center">
            <!-- exclamation Icon -->
            <i class="fa fa-exclamation-triangle fa-5x text-warning"></i>
            <!-- /exclamation Icon -->
            <h1>Payment status is Pending</h1>
            <!-- URL for back to checkout form -->
            <a href="<?= getAppUrl() ?>" title="Back to Checkout Form">Back to Checkout Form</a>
            <!-- /URL for back to checkout form -->
        </div>
    </div>
</body>
```

Payment Gateways Configurations

1. Sign up for a PayPal account please visit the following link.

https://www.paypal.com (for production)

https://developer.paypal.com/developer (for sandbox)

2. After creating your account update your **config.php** as per shown below.

```php
'paypal' => [
    'enable'                        => true,
    'testMode'                      => true, //test mode or product mode (boolean, true or false)
    'gateway'                       => 'Paypal', //payment gateway name
    'paypalSandboxBusinessEmail'        => 'Enter Paypal Sandbox Email', //paypal sandbox business email
    'paypalProductionBusinessEmail'     => 'Enter Paypal Business Email', //paypal production business email
    'currency'                  => 'USD', //currency
    'currencySymbol'                => '$',
    'paypalSandboxUrl'          => 'https://www.sandbox.paypal.com/cgi-bin/webscr', //paypal sandbox test mode Url
    'paypalProdUrl'             => 'https://www.paypal.com/cgi-bin/webscr', //paypal production mode Url
    'notifyIpnURl'              => 'payment-response.php', //paypal ipn request notify Url
    'cancelReturn'              => 'payment-response.php', //cancel payment Url
    'callbackUrl'               => 'payment-response.php' //callback Url after payment successful
],
```

The steps given below will take you through the setup process for **Razorpay**.

1. First, you need to visit the following link. <u>Razorpay Signup</u> and create your account, and grab your API and Secret key.
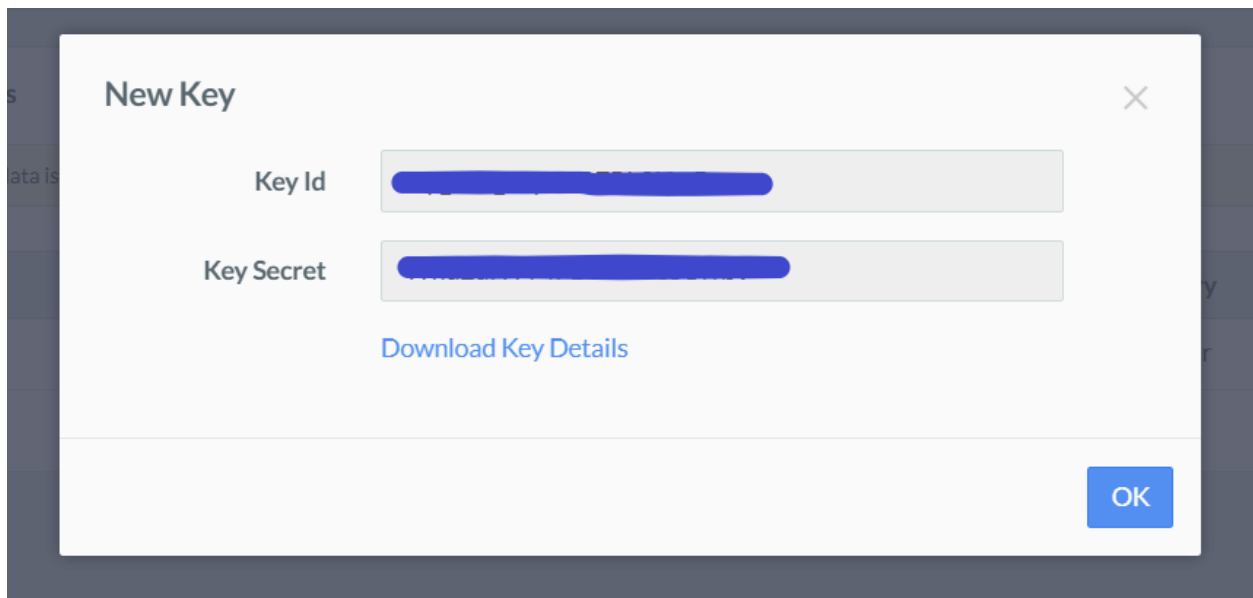
2. After getting your API and Secret key you need to paste into **config.php** file and set enable option true as per shown below.

```
'razorpay'     => [
    'enable'                      => true,
    'testMode'                    => true, //test mode or product mode (boolean, true or false)
    'gateway'                     => 'Razorpay', //payment gateway name
    'merchantname'                => 'John', //merchant name
    'themeColor'                  => '#4CAF50', //set razorpay widget theme color
    'currency'                    => 'INR', //currency
    'currencySymbol'               => '₹',
    'razorpayTestingkeyId'        => 'Enter your Test Api Key', //razorpay testing Api Key
    'razorpayTestingSecretkey'  => 'Enter your Test Secret Key', //razorpay testing Api Secret Key
    'razorpayLivekeyId'           => 'Enter your Live Api Key', //razorpay live Api Key
    'razorpayLiveSecretkey'       => 'Enter your Live Secret Key', //razorpay live Api Secret Key
    'callbackUrl'                 => 'payment-response.php' //callback Url after payment successful
],
```

3. Now access the **index.php** page inside the example folder to view all the
    payment method similarly you can access **example\razorpay-payment.php**
    to view only razorpay payment gateway.

4. Finally, after successful or unsuccessful payment you can grab a payment response in **payment-response.php** in paymentResponse() function, and you will get both (success/error) data in **$paymentResponseData** variable as an array.

Here you can send an email, redirect to any page or do whatever you want after payment completion.

```php
217  //payment response function
218  function paymentResponse($paymentResponseData) {
219      // If payment success
220      if ($paymentResponseData['status']) {
221          // Redirect to thank you page
222          header('Location: '. getAppUrl('thank-you.php'));
223      // If Payment Failed
224      } else {
225          // Redirect to failed payment page
226          header('Location: '. getAppUrl('payment-failed.php'));
227      }
```

Now we setup **Paystack** account for make a payment.

1. Visit following link. <u>Paystack Signup</u> and sign up. Grab your Public and Secret key.

2. After getting your Public and Secret key you need to paste into **config.php** file and set enable option true as per shown below.

```
'paystack' => [
    'enable'                    => true,
    'testMode'                  => true, //test mode or product mode (boolean, true or false)
    'gateway'                   => 'Paystack', //payment gateway name
    'currency'                  => 'NGN', //currency
    'currencySymbol'              => 'N',
    'paystackTestingSecretKey'      => 'Enter your Test Secret Key', //paystack testing secret key
    'paystackTestingPublicKey'      => 'Enter your Test Publish Key', //paystack testing public key
    'paystackLiveSecretKey'       => 'Enter your Live Secret Key', //paystack live secret key
    'paystackLivePublicKey'       => 'Enter your Live Publish Key', //paystack live public key
    'callbackUrl'               => 'payment-response.php' //callback Url after payment successful
],
```

3. Now access the **index.php** page inside the example folder to view all the
   payment method similarly you can access **example\paystack-payment.php**
   to view only paystack payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing
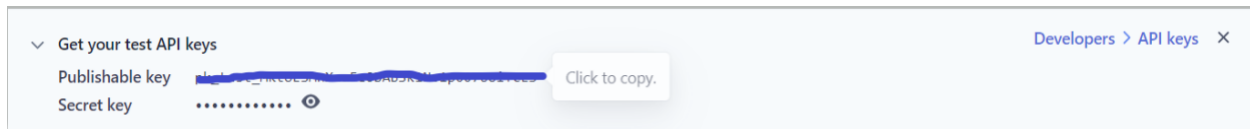
Item Name: Sample Product

○ paystack

VISA MasterCard Maestro AMERICAN EXPRESS

Proceed to Pay

The steps given below will take you through the setup process for Stripe.

1. To payment through stripe you have to visit following site. <u>Stripe</u> and get secret and publish key.

2. After getting your Public and Secret key you need to paste into **config.php** file and set enable option true as per shown below.

```php
'stripe'      => [
        'enable'                   => true,
        'testMode'                 => true, //test mode or product mode (boolean, true or
        'gateway'                  => 'Stripe', //payment gateway name
        'locale'                   => 'auto', //set local as auto
        'allowRememberMe'          => false, //set remember me ( true or false)
        'currency'                 => 'USD', //currency
        'currencySymbol'            => '$',
        'paymentMethodTypes'       => [
            // before activating additional payment methods
            // make sure that these methods are enabled in your stripe account
            // https://dashboard.stripe.com/settings/payments
            'card',
            // 'ideal',
            // 'bancontact',
            // 'giropay',
            // 'p24',
            // 'eps'
        ],
```

```
'stripeTestingSecretKey'    => 'Enter your Test Secret Key', //Stripe testing Secret Key
            'stripeTestingPublishKey'   => 'Enter your Test Publish Key', //Stripe testing Publish Key
            'stripeLiveSecretKey'       => 'Enter your Live Secret Key', //Stripe Secret live Key
            'stripeLivePublishKey'      => 'Enter your Live Publish Key', //Stripe live Publish Key
            'callbackUrl'               => 'payment-response.php', //callback Url after payment successful
            'privateItems'             => [
                'stripeTestingSecretKey',
                'stripeLiveSecretKey'
            ]
```

3. Now access the **index.php** page inside the example folder to view all the payment method similarly you can access **example\stripe-payment.php** to view only stripe payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing

Item Name: Sample Product

○ **stripe**

VISA MasterCard Maestro AMERICAN EXPRESS

Proceed to Pay

1. Create an account in Instamojo, using this link as shown in below URL,
   Note: If You are using sandbox environment the create account in this URL
   http://test.instamojo.com Or If you are using production mode environment
   then you need to create an account on http://www.instamojo.com, and grab
   your API key and auth token.

2. After getting your API key and Auth Token you need to paste into **config.php**
   file and set enable option true as per shown below.

```
'instamojo' => [
    'enable'                     => true,
    'testMode'                   => true, //test mode or product mode (boolean, true or false)
    'gateway'                    => 'Instamojo', //payment gateway name
    'currency'                   => 'INR', //currency
    'currencySymbol'               => '₹',
    'sendEmail'                  => false, //send mail (true or false)
    'instamojoTestingApiKey'         => 'Enter your Test Api Key', // instamojo testing API Key
    'instamojoTestingAuthTokenKey'   => 'Enter your Test Auth Token Key', // instamojo testing Auth token Key
    'instamojoLiveApiKey'          => 'Enter your Live Api Key', // instamojo live API Key
    'instamojoLiveAuthTokenKey'    => 'Enter your Live Auth Token Key', // instamojo live Auth token Key
    'instamojoSandboxRedirectUrl'  => 'https://test.instamojo.com/api/1.1/', // instamojo Sandbox redirect Url
    'instamojoProdRedirectUrl'     => 'https://www.instamojo.com/api/1.1/', // instamojo Production mode redirect Url
    'webhook'                    => 'http://instamojo.com/webhook/', // instamojo Webhook Url
    'callbackUrl'                => 'payment-response.php' //callback Url after payment successful
],
```

3. Now access the **index.php** page inside the example folder to view all the payment method similarly you can access **example\instamojo-payment.php** to view only Instamojo payment gateway.

1. Create an account of Paytm, using this link as shown in below URL,

   https://business.paytm.com/payment-gateway and get your Merchant ID and
   Account Secret Key.

2. After getting your Merchant ID and Account Secret Key you need to paste into
   config.php file and set enable option true as per shown below.

```
'paytm' => [
    'enable'                      => true,
    'testMode'                    => true, //test mode or product mode (boolean, true or false)
    'gateway'                     => 'Paytm', //payment gateway name
    'currency'                    => 'INR', //currency
    'currencySymbol'              => '₹',
    'paytmMerchantTestingMidKey'      => 'Enter your Test Mid Key', //paytm testing Merchant Mid key
    'paytmMerchantTestingSecretKey'   => 'Enter your Test Secret Key', //paytm testing Merchant Secret key
    'paytmMerchantLiveMidKey'         => 'Enter your Live Mid Key', //paytm live Merchant Mid key
    'paytmMerchantLiveSecretKey'      => 'Enter your Live Secret Key', //paytm live Merchant Secret key
    'industryTypeID'              => 'Retail', //industry type
    'channelID'                   => 'WEB', //channel Id
    'website'                     => 'WEBSTAGING',
    'paytmTxnUrl'                 => 'https://securegw-stage.paytm.in/theia/processTransaction', //paytm transaction Url
    'callbackUrl'                 => 'payment-response.php' //callback Url after payment successful or cancel payment
],
```

3. Now access the **index.php** page inside the example folder to view all the payment method similarly you can access **example\paytm-payment.php** to view only Instamojo payment gateway.

1. Create an account of Iyzipay, using this link as shown in below URL,

https://sandbox-merchant.iyzipay.com/auth/login and get your API and Secret Key.

2. After getting your Merchant ID and Account Secret Key you need to paste into **config.php** file and set enable option true as per shown below.

```
'iyzico'      => [
    'enable'                    => true,
    'testMode'                  => true, //test mode or product mode (boolean, true or false)
    'gateway'                   => 'Iyzico', //payment gateway name
    'conversation_id'           => 'CONVERS' . uniqid(), //generate random conversation id
    'currency'                  => 'TRY', //currency
    'currencySymbol'             => '₺',
    'subjectType'               => 1, // credit
    'txnType'                   => 2, // renewal
    'subscriptionPlanType'      => 1, //txn status
    'iyzicoTestingSecretkey'    => 'Enter your Test Secret Key', //iyzico testing Secret Key
    'iyzicoLiveApiKey'          => 'Enter your Live Api Key', //iyzico live Api Key
    'iyzicoLiveApiKey'          => 'Enter your Live Api Key', //iyzico live Api Key
    'iyzicoLiveSecretkey'       => 'Enter your Live Secret Key', //iyzico live Secret Key
    'iyzicoSandboxModeUrl'      => 'https://sandbox-api.iyzipay.com', //iyzico Sandbox test mode Url
    'iyzicoProductionModeUrl'   => 'https://api.iyzipay.com', //iyzico production mode Url
    'callbackUrl'               => 'payment-response.php' //callback Url after payment successful
]
```

3. Now access the **index.php** page inside the example folder to view all the payment method similarly you can access **example\iyzico-payment.php** to view only Iyzico payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing

Item Name: Sample Product

Proceed to Pay

Authorize.Net

1. Create an account as per link given below

   Note : If you want to use the sandbox account then visit
   https://developer.authorize.net/hello_world/sandbox/ or if you want to use
   the production mode then use https://www.authorize.net/sign-up/ and get
   your API login id and transaction key.

API LOGIN ID

TRANSACTION KEY

KEY

Simon

2. After getting your API login id and transaction key you need to set into
   config.php file

```php
'authorize-net'    => [
    'enable'                        => true,
    'testMode'                      => true, //test mode or product mode (boolean, true or false)
    'gateway'                       => 'Authorize.net', //payment gateway name
    'reference_id'                  => 'REF' . uniqid(), //generate random conversation id
    'currency'                      => 'USD', //currency
    'currencySymbol'                => '$',
    'type'                          => 'individual',
    'txnType'                       => 'authCaptureTransaction',
    'authorizeNetTestApiLoginId'    => 'Your Test API Login ID', //authorize-net testing Api login id
    'authorizeNetTestTransactionKey' => 'Your Test tTransaction Key', //Authorize.net testing transaction key
    'authorizeNetLiveApiLoginId'    => 'Your Live API Login ID', //Authorize.net live Api login id
    'authorizeNetLiveTransactionKey' => 'Your Live tTransaction Key', //Authorize.net live transaction key
    'callbackUrl'                   => 'payment-response.php', //callback Url after payment successful
    'privateItems'                  => [
                                        'authorizeNetTestApiLoginId',
                                        'authorizeNetTestTransactionKey',
                                        'authorizeNetLiveApiLoginId',
                                        'authorizeNetLiveTransactionKey'
                                    ]
```

3. Now access the **index.php** page inside the example folder to view all the payment method similarly you can access **example\authorize-net-payment.php** to view Authorize.Net payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

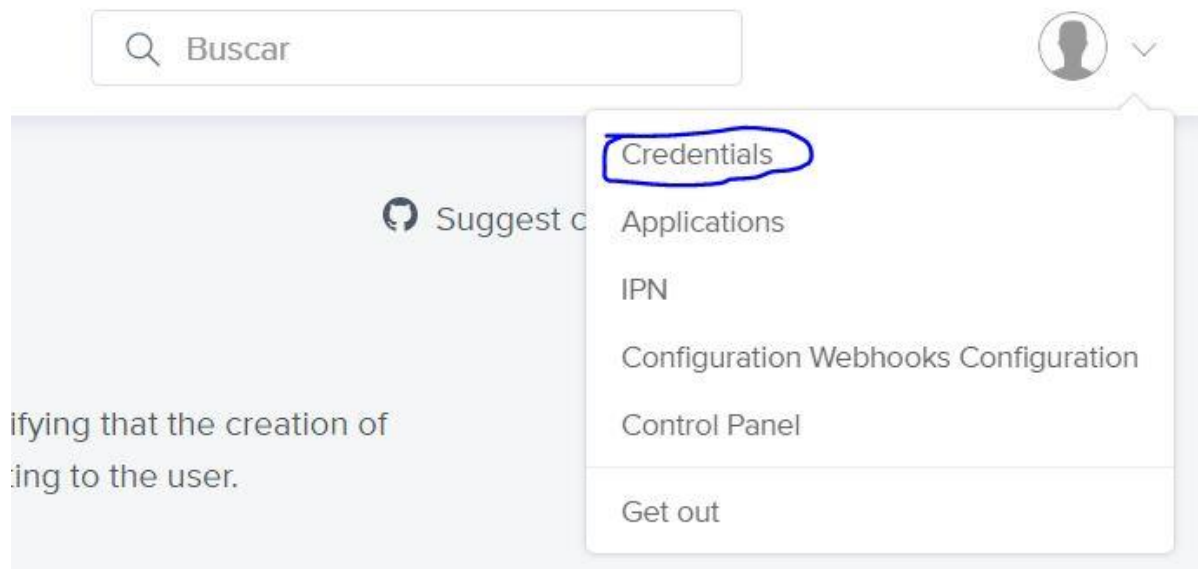**Additional info:** Lorem ipsum dolor sit amet, constetur adipisicing

Item Name: Sample Product

○ Authorize.Net

MasterCard VISA AMERICAN EXPRESS DISCOVER

Proceed to Pay

1. Create your Mercado Pago account on link given below.
   https://www.mercadopago.com.ar/registration-mp

2. Grab your access token from credential section as per below.

3. If you are using sandbox account then copy the access token from sandbox mode or you are using the production mode then copy access token from production mode.

4. After getting your access token paste it into **config.php** as per given below.

```
'mercadopago' => [
    'enable'              => true,
    'testMode'            => true, //test mode or product mode (boolean, true or false)
    'currency'            => 'USD', //currency
    'currencySymbol'      => '$', //currency Symbol
    'testAccessToken'     => 'Your Test Access Token',
    'liveAccessToken'     => 'Your Live Access Token',
    'callbackUrl'         => 'payment-response.php', //callback Url after payment successful
    'privateItems'        => ['testAccessToken', 'liveAccessToken']
]
],
```

5. Now access the index.php page inside the example folder to view all the payment method. Similarly, you can access **example\ mercadopago-payment.php** to view Mercado Pago payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing

Item Name: Sample Product

Proceed to Pay

![PayUmoney logo]

1. Create an account on link given below

https://onboarding.payumoney.com/app/account/

2. Get your Merchant test key and salt from your account.

Here are your Integration Credentials

Live Credentials   **Test Credentials**

Merchant Key   Test Key
�█████████   ████████

Merchant Salt   Test Salt
████████   █████████

Auth Header   Test Auth Header
you can click here to generate header   you can click here to generate header

                                           Test card details

3. After getting your key and salt you need to paste it into config.php as per given below.

```
'payumoney' => [
    'enable'                    => true,
    'testMode'                  => true, //test mode or product mode (boolean, true or false)
    'gateway'                   => 'PayUmoney', //payment gateway name
    'currency'                  => 'INR', //currency
    'currencySymbol'            => '₹', //currency Symbol
    'txnId'                     => "Txn" . rand(10000,99999999),
    'merchantTestKey'           => 'Your Test Merchant Key',
    'merchantTestSalt'          => 'Your Test Salt Key',
    'merchantLiveKey'           => 'Your Live Merchant Key',
    'merchantLiveSalt'          => 'Your Live Salt Key',
    'callbackUrl'               => 'payment-response.php', //callback Url after payment successful
    'checkoutColor'             => 'e34524',
    'checkoutLogo'              => 'http://boltiswatching.com/wp-content/
        uploads/2015/09/Bolt-Logo-e14421724859591.png',
    'privateItems'              => ['merchantTestKey', 'merchantTestSalt', 'merchantLiveKey', 'merchantLiveSalt']
],
```

4. Now access the **index.php** page inside the example folder to view all the payment method. Similarly, you can access **example\ payumoney-payment.php** to view PayUMoney payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing

Item Name: Sample Product

○ PayU'money

MasterCard VISA AMERICAN EXPRESS DISCOVER AMERICAN EXPRESS DISCOVER NETWORK 🏛

Proceed to Pay

mollie

1. Create an account on link given below https://www.mollie.com/dashboard/signup

2. Get your API key from your account.

Use the Test API key to test your integration, and the Live API key to create real payments. Please keep them private as much as possible. Read more about integrating our API.
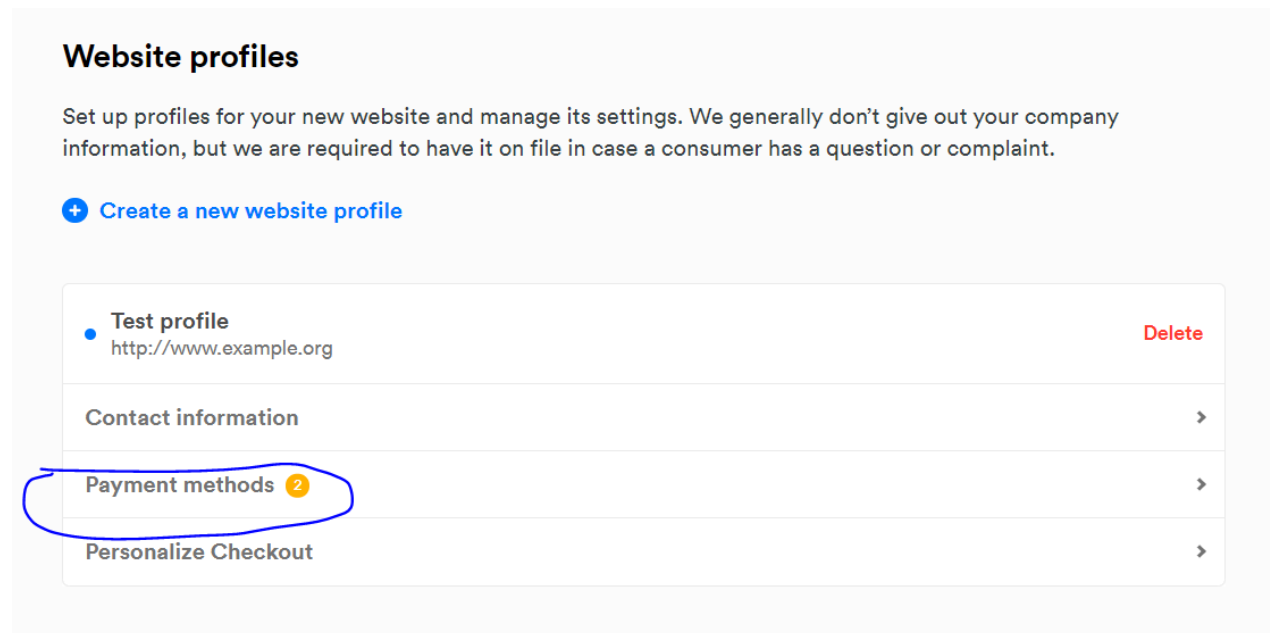
For access to the more advanced organization-level APIs, you can create organization access tokens.

Test profile
http://www.example.org

● **Live API key**	Visible when your website profile has been verified.		
● **Test API key**	████████████████████████	Copy	Reset
● **Profile ID**	████████████		Copy

4. After getting your key you need to paste it into config.php as per given below.

```php
'mollie' => [
    'enable'                => true,
    'testMode'              => true, //test mode or product mode (boolean, true or false)
    'gateway'               => 'mollie', //payment gateway name
    'currency'              => 'EUR', //currency
    'currencySymbol'        => '€', //currency Symbol
    'testApiKey'            => 'Your Test API Key',
    'liveApiKey'            => 'Your Live API Key',
    'callbackUrl'           => 'payment-response.php', //callback Url after payment successful
    'privateItems'          => ['testApiKey', 'liveApiKey']
],
],
```

5. After that click on **Settings** from sidebar menu and select **Payment Methods** from the setting page.

And select which payment method you want to use in **mollie** payment gateway.

6. Now access the **index.php** page inside the example folder to view all the payment method. Similarly, you can access **example\ mollie-payment.php** to view Mollie payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing
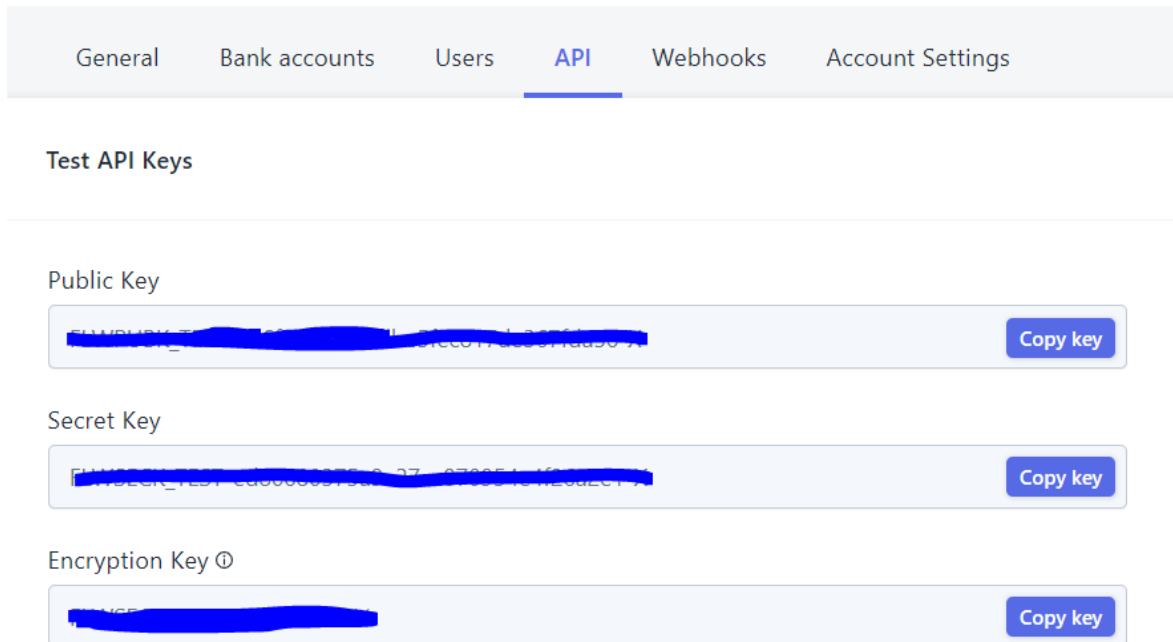
Item Name: Sample Product

○ mollie

Proceed to Pay

First setup the **Ravepay** account for making the payment through it as per follows

1. To create the Ravepay account visit following link <u>Ravepay Signup</u> and sign up. Grab your Public and Secret key.

2. After getting your Public and Secret key you need to paste it into **config.php** file and set enable option true as per shown below.

```
'ravepay' => [
    'enable'                       => true,
    'testMode'                     => true, //test mode or product mode (boolean, true or false)
    'gateway'                      => 'Ravepay', //payment gateway name
    'currency'                     => 'NGN', //currency
    'currencySymbol'               => '₦', //currency Symbol
    'txn_reference_id'             => 'REF' . uniqid(), //generate random conversation id
    'testPublicApiKey'             => 'Your Test Public API Key',
    'testSecretApiKey'             => 'Your Test Secret API Key',
    'livePublicApiKey'             => 'Your Live Public API Key',
    'liveSecretApiKey'             => 'Your Live Secret API Key',
    'callbackUrl'                  => 'payment-response.php', //callback Url after payment successful
    'sandboxVerifyPaymentUrl'      => 'https://ravesandboxapi.flutterwave.com/flwv3-pug/getpaidx/api/v2/verify', //sandbox
    staging server verify payment url.
    'productionVerifyPaymentUrl'   => 'https://api.ravepay.co/flwv3-pug/getpaidx/api/v2/verify', //production staging
    server verify payment url.
    'privateItems'                 => ['testSecretApiKey', 'liveSecretApiKey']
],
```

3. Now access the **index.php** page inside the example folder to view all the
   payment methods. Similarly, you can access **example\ravepay-payment.php**
   to view only Ravepay payment gateway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing

Item Name: Sample Product

○ rave BY FLUTTERWAVE

MasterCard VISA AMERICAN EXPRESS DISCOVER

Proceed to Pay

To setup a **PagSeguro** account to making the payment through it follow the procedure

1. Create your PagSeguro account on the link given below.

   https://sandbox.pagseguro.uol.com.br

2. Grab your access token and email from the credentials section on the sidebar link as per below.

3. If you are using a sandbox account then copy the email and token from sandbox mode and when you are using the production mode then copy email and token from production mode.

Testing Vendor

Credentials

Use the information below to access the PagSeguro Sandbox APIs.

Your email is: ▓▓▓▓▓▓▓▓▓▓▓	**copy**
Your token is: ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓	**copy**

4. After getting your access token paste it into **config.php** as per given below.

```
'pagseguro' => [
    'enable'            => true,
    'testMode'          => true, //test mode or product mode (boolean, true or false)
    'gateway'           => 'Pagseguro', //payment gateway name
    'environment'       => 'sandbox', //production, sandbox
    'currency'          => 'BRL', //currency
    'currencySymbol'    => 'R$', //currency Symbol
    'reference_id'      => 'REF' . uniqid(), //generate random refrence id
    'email'             => 'Your Email', //your pagseguro email id for create account credentials
    'testToken'         => 'Your Test Sandbox Token', //your sandbox pagseguro token for create account
    credentials
    'liveToken'         => 'Your Live Production Token', //your production pagseguro token for create
    account credentials
    'callbackUrl'       => 'payment-response.php', //callback Url after payment successful
    'notificationUrl'   => 'payment-response.php', //notification url when payment successfull user collect
    notfication data
    'privateItems'      => ['live_token']
],
```

5. Now access the index.php page inside the example folder to view all the payment methods. Similarly you can access example\ pagseguro-payment.php to view only **PagSeguro** payment gateway.

   Note - If any doubts, please refer docs as per link on given below,

   https://dev.pagseguro.uol.com.br/docs

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing
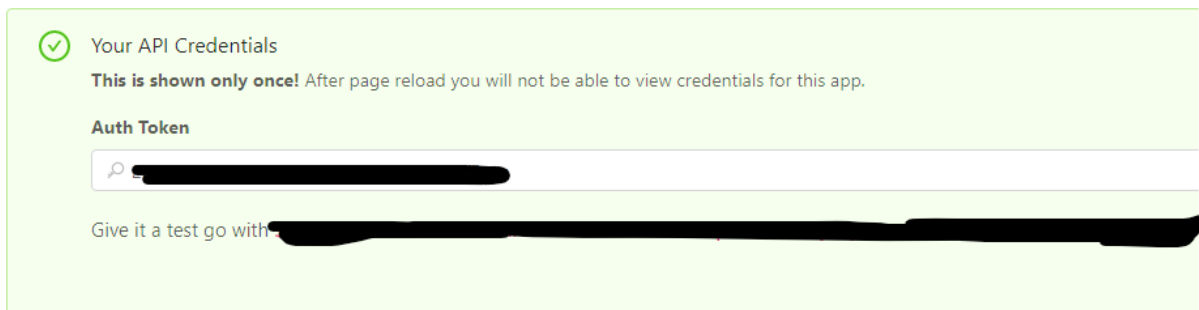
Item Name: Sample Product

○ *PagSeguro*

Proceed to Pay

The steps given below will take you through the setup process for **Coingate**.

1. First, you need to visit the following link. Coingate Signup, create your account, and grab your API token.

2. After getting your token key you need to paste into **config.php** file and set enable option true as per shown below.

```
'coingate' ⇒ [
    'enable'                    ⇒ true,
    'testMode'                  ⇒ true, //test mode or product mode (boolean, true or false)
    'gateway'                   ⇒ 'Coingate', //payment gateway name
    'environment'               ⇒ 'sandbox', //production, sandbox
    'currency'                  ⇒ 'USD', //currency
    'currencySymbol'            ⇒ '$', //currency Symbol
    'reference_id'              ⇒ 'REF' . uniqid(), //generate random reference id
    'testToken'                 ⇒ 'Your Sandbox Token',//your sandbox coingate token for create account
credentials
    'liveToken'                 ⇒ 'Your Live Production Token', //your production coingate token for
create account credentials
    'callbackUrl'               ⇒ 'payment-response.php', //callback Url after payment successful
    'notificationUrl'           ⇒ 'payment-response.php', //notification url when payment successfully
user collect notification data
    'privateItems'              ⇒ ['liveToken', 'testToken']
],
```

3. Now access the **index.php** page inside the example folder to view all the payment methods similarly you can access **example\coingate-payment.php** to view only Coingate payment gateway.

s

4. Finally, after successful or unsuccessful payment you can grab a payment response in **payment-response.php** in the paymentResponse() function, and you will get both (success/error) data in **$paymentResponseData** variable as an array.
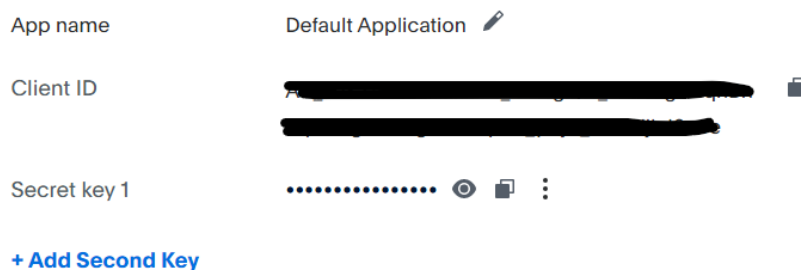
Here you can send an email, redirect to any page or do whatever you want after payment completion.

PayPal

The steps given below will take you through the setup process for **PayPal Checkout**.

1. First, you need to visit the following link. <u>PayPal Signup</u>, create your account, and grab your API Client ID and Secret key.

API credentials

App name	Default Application ✏
Client ID	
Secret key 1	••••••••••••••• ⊙ ⧉ ⋮

+ Add Second Key

2. After getting your API and Secret key you need to paste into **config.php** file and set enable option true as per shown below.

```
'paypal-checkout' ⇒ [
    'enable'                    ⇒ true,
    'testMode'                  ⇒ true, //test mode or product mode (boolean, true or false)
    'gateway'                   ⇒ 'Paypal Checkout', //payment gateway name
    'paypalTestingClientKey'    ⇒ 'Enter your Test Client Key', //paypal Testing Client key
    'paypalTestingSecretKey'    ⇒ 'Enter your Test Secret Key', //paypal Testing Secret key
    'paypalLiveClientKey'       ⇒ 'Enter your Live Client Key', //paypal Live Client key
    'paypalLiveSecretKey'       ⇒ 'Enter your Live Secret Key', //paypal Live Secret key
    'currency'                  ⇒ 'USD', //currency
    'currencySymbol'            ⇒ '$',
    'paypalSandboxUrl'          ⇒ 'https://api-m.sandbox.paypal.com', //paypal sandbox test mode Url
    'paypalProdUrl'             ⇒ 'https://api-m.paypal.com', //paypal production mode Url
    'successUrl'                ⇒ 'payment-response.php', //paypal ipn request notify Url
    'cancelReturn'              ⇒ 'payment-response.php', //cancel payment Url
    'callbackUrl'               ⇒ 'payment-response.php', //callback Url after payment successful
    'privateItems'              ⇒ [
        'paypalTestingSecretKey',
        'paypalLiveSecretKey'
    ]
```

3. Now access the **index.php** page inside the example folder to view all the payment methods similarly you can access **example\paypal-checkout-payment.php** to view only PayPal Checkout payment gat0eway.

Complete your payment

This is just an example, you can integrate these payment gateway easily using this example implementation.

Customer Details

Name: John Doe

Email: sample@domain.com

Address: 3234 Godfrey Street Tigard, OR 97223

City: Tigard

Country: United States

Additional info: Lorem ipsum dolor sit amet, constetur adipisicing

Item Name: Sample Product

Check out with PayPal

$ 100 USD

4. Finally, after successful or unsuccessful payment you can grab a payment response in **payment-response.php** in the paymentResponse() function,

and you will get both (success/error) data in **$paymentResponseData** variable as an array.

Here you can send an email, redirect to any page or do whatever you want after payment completion.

Thank You

We have tried our best to explain the functionality as possible as via this document, Our System also help you while interacting with application.
We hope this guide will help you lot.

=[you if you have any questions relating to this product. No Guarantees, but We will do our best to assist.

We are the team of Designer & Developers, here are some of the Great tools and technologies we use day2day for the development:

HTML5, CSS3, jQuery, PHP, Laravel, Angular, AngularJS etc.

We also build **iOS & Android Application.**

If you need any Customization Work / Support on this Product or having any **Project** for us.

Please feel free to email us at contact@livelyworks.net

Our New Service

We are happy to inform you that we just launched our new service called ServePHP for managing Servers and Applications for PHP.

ServePHP will help you in creating, provisioning and managing server tuned for PHP. It helps in performing many important tasks like Attaching Domains, Applications, Cron Jobs, Firewall Rules, Auto Deployments and many more without writing single line in Terminal, these tasks are just a click away using simple interface.

Many Advance features like Auto Deployments using Git Push requests, these deployments can be configured conditionally to avoid unnecessary commands executions.

Get started with free account without any credit card information and you can quick sign up via the Google, GitHub, Bitbucket, DigitalOcean, Slack, Gitlab to get hassle free ride to functions & features of site.

To know more about this application please visit our site at the servephp.com

Thank You & Good Luck

livelyworks.net | contact@livelyworks.net net | @livelyworks | https://github.com/livelyworks