

Payment Api Documentation

How to create payment method

There are several payment methods available itself and. Although sometimes you'll find yourself in the situation where you need something different, either there is no method available for your choice of payment gateway or you want some different logic. In either case, you're left with the only option: To create a new payment method module.

We'll assume that our custom payment method name is "custom". There are at least three or four files and one folder you can create in order to set up the things. Let's check the same in detail.

You need to create three file. each file are required. following are the folder structure.

1. controller -> custom.php
2. setting -> custom.php
3. view -> custom.php

You can create other one folder and one file. Folder and file are optional.following are the folder structure.

4. library -> custom
5. logo -> custom.png

#1 Controller folder

In this folder contain custom.php file. in this file contain all logic of your payment gateway. Our payment gateway send three types of request to your controller. Every request keep two parameter. They are \$settingData and \$gatewayData. You can get their sample data from Get sample data button.

Folder structure must be like this:

custom/controller/custom.php

Example for custom.php

```
<?php
class custom {
    public $title = 'Custom Payment Gateway';
    public $icon = 'assets/payment_gateway/custom.png';
    public $website = '';

    function __construct($api){ $this->api = $api; }

    public function getPaymentGatewayView($settingData,$gatewayData){
        $view = APPPATH."payment_gateway/views/custom.php";

        require $view;
    }

    public function setPaymentGatewayRequest($settingData,$gatewayData){}

    public function customCallbackFunction($settingData,$gatewayData){}

}
```

Explanation of file:

Class Name Class name must be file name

Public Property Title Name of payment gateway

Constructor must be as it is.. api variable contain this object of CI. you can used functionality of CI

public function getPaymentGatewayView This function is required. We send this request for get your custom view file to show user.

public function setPaymentGatewayRequest This function is optional. This function prepare request for send to your payment gateway.

public function customCallbackFunction This function is optional. This function accept callback from your payment gateway. You can named callback function what you want. if you want after payment completed your custom function called by your payment gateway, you must add your callback function name to \$gatewayData["callback_url"] for send to your payment gateway in getPaymentGatewayView or setPaymentGatewayRequest.

#2 Setting folder

In this folder you can add your custom setting file. The setting file name must be same with your payment mehtod name. Also setting file must be located under setting folder. Folder structure must be like this:

custom/setting/custom.php

Example for custom.php

```
<div class="form-group">
    <label class="form-control-label">Some Setting</label>
    <input class="form-
control" name="name" value="<?= $setting_data["name"] ?>" >
</div>

<div class="form-group">
    <label class="control-label" for="input-completed-
status">Completed Status</label>
    <select name="completed_status_id" id="input-completed-status" class="form-
control">
        <?php foreach ($order_status as $order_status_id => $name){
            if(isset($setting_data["completed_status_id"]))
                $selected = ($order_status_id == $setting_data["completed_status_id"])
? "selected" : "";
            else
                $selected = ($order_status_id == 1) ? "selected" : ""; ?>

            <option <?= $selected ?> value="<?= $order_status_id; ?>"><?= $name ?>
</option>
        <?php } ?>
    </select>
</div>
```

Explanation of file:

Setting file contain \$setting_data variable and this variable contain all saved setting of your payment

gateway. Some time you need to ask information from admin for example credential of payment gateway or something else. Also you can manage the status of transaction in project according to the result of the payment. \$order_status lists are shown in "Status ID And Title" section. You just need to create a input out system will auto create a setting page and save setting data.

#3 View folder

In this folder you can add your custom view file. The view file name must be same with your payment mehtod name. Also view file must be located under view folder. Folder structure must be like this:

custom/view/custom.php

Example for custom.php

```
<div class="payment-button-group">
  <button type="button" class="btn btn-
default" onclick="backCheckout()">Back</button>
  <button id="button-confirm" class="btn btn-primary">Confirm</button>
</div>
<script type="text/javascript">
$( "#button-confirm" ).click(function() {
  $this = $(this);
  $this.prop( "disabled", true );

  $.ajax({
    url: "<?= $gatewayData[ "payment_confirmation" ] ?>",
    type: "POST",
    dataType: "json",
    data: $( "[name^= 'comment' ]" ).serialize(),
    beforeSend: function() { $( "#button-confirm" ).btn( "loading" ); },
    complete: function() { $( "#button-confirm" ).btn( "reset" ); },
    success: function( json ) {
      $container = $( "#checkout-confirm" );
      $container.find( ".has-error" ).removeClass( "has-error" );
      $container.find( "span.text-danger" ).remove()

      if( json[ "errors" ] ) {
        $.each( json[ "errors" ][ "comment" ], function( ii, jj ) {
          $ele = $container.find( "#comment_textarea"+ ii );
          if( $ele ) {
            $ele.parents( ".form-group" ).addClass( "has-error" );
            $ele.after( "<span class= 'text-danger' >"+ jj + "</span>" );
          }
        } );
      }
      if( json[ "success" ] ) {
        $.ajax({
          url: "<?= $gatewayData[ 'confirm_payment' ] ?>",
          type: "POST",
          dataType: "json",
          data: {
            payment_gateway: $( "input[ name= 'payment_gateway' ]:checked" ).val()
          },
          beforeSend: function() { $this.btn( "loading" ); },
          complete: function() { $this.btn( "reset" ); },
```

```

        success:function(json){
            if(json["redirect"]){
                window.location = json["redirect"];

                if(json["warning"]){
                    alert(json["warning"]);
                },
            });
        },
    });
}
</script>

```

Explanation of file:

View file contain getPaymentGatewayView function parameter \$settingData and \$gatewayData. You can use them in view file. In view file you must send request to \$gatewayData["payment_confirmation"] like above example. After get response from that request, you have two choices. In first choice, you can directly redirect user to payment gateway page (ex.using form). In second choice, you can send request to \$gatewayData["confirm_payment"], this means that, you call setPaymentGatewayRequest method. Generally view file code structure depends on your payment gateway.

#4 Library folder

In this folder you can add your payment gateway source code if it exist. The library folder name must be same with your payment method name. If you want to call it from contoller, you can do it with this line : require_once(APPPATH."payment_gateway/library/custom/.php"). Also all payment method source code must be located under that folder. Folder structure must be like this:

custom/library/custom/

#5 Logo folder

In this folder you can add your payment gateway logo if it exist. The logo name must be same with your payment method name. Also logo must be located under logo folder. Folder structure must be like this:

custom/logo/custom.png

Payment confirmation

\$this->api->confirmPaymentGateway(\$gatewayData["id"],\$status_id,\$transaction_id,\$payment_status);
 You must call this function after payment completed. \$gatewayData["id"] must not change. You must get \$status_id from \$settingData. If you want send it directly, you can select status from "Status ID And Title" section. \$transaction_id and \$payment_status are optional.

Status ID And Title

Status ID	Title
0	Received
1	Complete
2	Total not match
3	Denied

4	Expired
5	Failed
7	Processed
8	Refunded
9	Reversed
10	Voided
11	Canceled Reversal
12	Waiting For Payment
13	Pending

How to create zip file of payment method

Zip file name must be same with payment method name. Folder structure must be like this:

custom/controller/custom.php

custom/setting/custom.php

custom/view/custom.php

custom/library/custom/

custom/logo/custom.png