Integrating the Ping++ SDK with the CodeIgniter Framework

Below is a complete, step-by-step example of how to integrate **Pingpp** (Ping++), a payment processing SDK, into a PHP application using the CodeIgniter framework, based on the provided `BaseController` class. This guide assumes you have a basic CodeIgniter setup and focuses on integrating Pingpp for payment processing, such as enabling users to pay via Alipay PC Direct (though you can adapt it for other channels).

Prerequisites

1. **Pingpp Account**: Sign up at Ping++ and obtain your **API Key** and **App ID**. You'll have a test key (`sk_test_...`) for development and a live key (`sk_live_...`) for production.
2. **Pingpp PHP SDK**: Install the Pingpp PHP SDK via Composer:

   ```
   composer require pingplusplus/pingpp-php
   ```

   Ensure the SDK is autoloaded in your CodeIgniter application (e.g., place it in `application/libraries` or configure Composer autoloading).
3. **CodeIgniter Setup**: You should have a working CodeIgniter application with a controller and database models for users and charges.

Step-by-Step Integration

1. Set Up the Controller Create a controller (e.g., `PaymentController.php`) that extends a base controller similar to `BaseController`. Configure the Pingpp API key in the constructor based on your environment (development or production).

```php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require_once APPPATH . '/libraries/REST_Controller.php';
require_once APPPATH . '/vendor/autoload.php'; // Assuming Composer autoload

class PaymentController extends REST_Controller {

    public $userDao;
    public $chargeDao;
```

```php
    public function __construct() {
        parent::__construct();

        // Set Pingpp API key based on environment
        $isLocalDebug = ENVIRONMENT === 'development'; // CodeIgniter environment
        if ($isLocalDebug) {
            \Pingpp\Pingpp::setApiKey('sk_test_your_test_key_here'); // Replace with your test key
        } else {
            \Pingpp\Pingpp::setApiKey('sk_live_your_live_key_here'); // Replace with your live key
        }

        // Load models
        $this->load->model('UserDao');
        $this->userDao = new UserDao();
        $this->load->model('ChargeDao');
        $this->chargeDao = new ChargeDao();
    }
}
```

- Replace `'sk_test_your_test_key_here'` and `'sk_live_your_live_key_here'` with your actual Pingpp API keys.
- Ensure `UserDao` and `ChargeDao` are implemented to handle user authentication and charge storage (see Step 2).

2. Create Supporting Models You’ll need models to manage user sessions and store charge details.

UserDao.php (simplified example):

```php
<?php
class UserDao extends CI_Model {
    public function findUserBySessionToken($token) {
        $query = $this->db->get_where('users', ['session_token' => $token]);
        return $query->row(); // Returns user object or null
    }
}
```

ChargeDao.php (simplified example):

```php
<?php
class ChargeDao extends CI_Model {
    public function add($orderNo, $amount, $userId, $ipAddress) {
        $data = [
            'order_no' => $orderNo,
            'amount' => $amount,
            'user_id' => $userId,
            'client_ip' => $ipAddress,
            'status' => 'pending',
            'created_at' => date('Y-m-d H:i:s')
        ];
        $this->db->insert('charges', $data);
    }
}
```

Create corresponding database tables (`users` and `charges`) with appropriate fields.

3. Implement the Charge Creation Method Add a method to create a Pingpp charge and return it to the client. This example uses Alipay PC Direct as the payment channel.

```
public function createChargeThenResponse($amount, $subject, $body, $metaData, $user) {
    // Generate a unique order number
    $orderNo = $this->generateOrderNo();

    // Set App ID based on environment
    $isLocalDebug = ENVIRONMENT === 'development';
    $appId = $isLocalDebug ? 'app_your_test_app_id' : 'app_your_live_app_id';

    // Get client IP address
    $ipAddress = $this->input->ip_address();
    if ($ipAddress === '::1') { // Handle local debug case
        $ipAddress = '127.0.0.1';
    }

    // Create the charge
    try {
        $charge = \Pingpp\Charge::create([
            'order_no' => $orderNo,
            'app' => ['id' => $appId],
            'channel' => 'alipay_pc_direct', // Change this for other channels (e.g., 'wx' for WeChat)
            'amount' => $amount, // In cents (e.g., 1000 = 10 CNY)
            'client_ip' => $ipAddress,
            'currency' => 'cny',
            'subject' => $subject,
            'body' => $body,
            'metadata' => $metaData,
            'extra' => [
                'success_url' => 'http://yourdomain.com/payment/success' // Replace with your success URL
            ]
        ]);

        // Store charge details in database
        $this->chargeDao->add($orderNo, $amount, $user->id, $ipAddress);

        // Return the charge object as JSON
        $this->output
            ->set_status_header(200)
            ->set_content_type('application/json', 'utf-8')
            ->set_output(json_encode($charge));
    } catch (\Pingpp\Error\Base $e) {
        log_message('error', 'Pingpp Charge Failed: ' . $e->getMessage());
        $this->response(['error' => 'Payment creation failed'], 500);
    }
}

private function generateOrderNo() {
    return uniqid(); // Simple unique ID; replace with a more robust generator if needed
}
```

- Replace `'app_your_test_app_id'` and `'app_your_live_app_id'` with your Pingpp App IDs.
- Adjust `'success_url'` to your application’s success page URL.

4. Create a Checkout Method Add a public method to handle user checkout, calling the charge creation method.

```
public function checkout_post() {
    // Check if user is logged in
    $user = $this->getSessionUser();
    if (!$user) {
        $this->response(['error' => 'User not logged in'], 401);
        return;
    }
    // Example order details
    $amount = 1000; // 10 CNY (example; calculate dynamically in a real app)
    $subject = 'Order Payment';
    $body = 'Payment for order #123';
    $metaData = ['order_id' => '123']; // Attach order-specific data

    // Create charge and respond
    $this->createChargeThenResponse($amount, $subject, $body, $metaData, $user);
}

private function getSessionUser() {
    $token = $this->input->get_request_header('Authorization', TRUE); // Adjust based on your auth method
    if ($token) {
        return $this->userDao->findUserBySessionToken($token);
    }
    return null;
}
```

5. Handle the Client-Side Response The `createChargeThenResponse` method returns a JSON `charge` object. For `alipay_pc_direct`, it includes a `credential` field with a payment URL. Handle this on the client side (e.g., with JavaScript).

Example (Frontend JavaScript):

```
fetch('/payment/checkout', {
    method: 'POST',
    headers: { 'Authorization': 'your-session-token' }
})
.then(response => response.json())
.then(charge => {
    if (charge.credential && charge.credential.alipay_pc_direct) {
        window.location.href = charge.credential.alipay_pc_direct; // Redirect to Alipay
    } else {
        console.error('Payment error:', charge);
    }
})
.catch(error => console.error('Error:', error));
```

6. Success Page Create a success page (e.g., `payment/success`) to display a confirmation after payment. Update the order status manually here if needed, though webhooks are recommended for reliability (see

Step 7).

```
public function success_get() {
    // Optionally verify payment status here
    $this->load->view('payment_success'); // Load a view
}
```

7. (Optional) Handle Webhooks Pingpp sends asynchronous payment status updates via webhooks. Set up an endpoint to receive these notifications and update your database.

```
public function webhook_post() {
    $rawData = file_get_contents('php://input');
    $event = json_decode($rawData, true);

    if ($event['type'] === 'charge.succeeded') {
        $charge = $event['data']['object'];
        $orderNo = $charge['order_no'];
        $this->chargeDao->updateStatus($orderNo, 'paid');
    }

    $this->response(['status' => 'ok'], 200);
}
```

- Add an `updateStatus` method to `ChargeDao` to update the charge status.
- Configure your webhook URL in the Pingpp dashboard (e.g., `http://yourdomain.com/payment/webhook`).

Full Example Usage

1. **User Action**: A user logs in and initiates a checkout via a POST request to `/payment/checkout`.
2. **Server Response**: The server creates a charge and returns a JSON object with payment details.
3. **Client Handling**: The frontend redirects the user to the Alipay payment page.
4. **Payment Completion**: After payment, the user is redirected to the success URL.
5. **Webhook Update**: Pingpp notifies your webhook endpoint, and you update the charge status.

Additional Notes

- **Payment Channels**: Replace `'alipay_pc_direct'` with other channels (e.g., `'wx'` for WeChat) as needed. Check the Pingpp documentation for channel-specific `extra` parameters.

- **Error Handling**: Enhance error handling for production (e.g., retry logic, detailed error messages).
- **Security**: Validate user input (`$amount`, etc.) and secure your API endpoints.

This example provides a solid foundation for integrating Pingpp into your CodeIgniter application, adaptable to various use cases like e-commerce or donations.