

# Ping++ SDK in das CodeIgniter Framework integrieren

Unten ist ein vollständiges, schrittweise Beispiel, wie man **Pingpp** (Ping++) eine Zahlungsverarbeitungs-SDK in eine PHP-Anwendung mit dem CodeIgniter-Framework integriert, basierend auf der bereitgestellten BaseController-Klasse. Diese Anleitung setzt voraus, dass Sie eine grundlegende CodeIgniter-Einrichtung haben und sich auf die Integration von Pingpp für die Zahlungsverarbeitung konzentriert, wie z.B. das ermöglichen von Zahlungen über Alipay PC Direct (obwohl Sie es für andere Kanäle anpassen können).

---

## Voraussetzungen

1. **Pingpp-Konto:** Registrieren Sie sich bei Ping++ und erhalten Sie Ihren **API-Schlüssel** und **App-ID**. Sie haben einen Testschlüssel (`sk_test_...`) für die Entwicklung und einen Live-Schlüssel (`sk_live_...`) für die Produktion.
2. **Pingpp PHP SDK:** Installieren Sie das Pingpp PHP SDK über Composer:

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

Stellen Sie sicher, dass die SDK in Ihrer CodeIgniter-Anwendung autogeladen wird (z.B. legen Sie sie in `application/libraries` ab oder konfigurieren Sie das Composer-Autoloading).

3. **CodeIgniter-Einrichtung:** Sie sollten eine funktionierende CodeIgniter-Anwendung mit einem Controller und Datenbankmodellen für Benutzer und Gebühren haben.
- 

## Schritt-für-Schritt-Integration

1. **Controller einrichten** Erstellen Sie einen Controller (z.B. `PaymentController.php`), der eine Basiscontroller ähnlich wie `BaseController` erweitert. Konfigurieren Sie den Pingpp API-Schlüssel im Konstruktor basierend auf Ihrer Umgebung (Entwicklung oder Produktion).

```
<?php  
defined('BASEPATH') OR exit('Kein direkter Skriptzugriff erlaubt');  
  
require_once APPPATH . '/libraries/REST_Controller.php';  
require_once APPPATH . '/vendor/autoload.php'; // Angenommen, Composer Autoload  
  
class PaymentController extends REST_Controller {
```

```

public $userDao;
public $chargeDao;

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

    // Setzen Sie den Pingpp API-Schlüssel basierend auf der Umgebung
    $isLocalDebug = ENVIRONMENT === 'development'; // CodeIgniter-Umgebung
    if ($isLocalDebug) {
        \Pingpp\Pingpp::setApiKey('sk_test_your_test_key_here'); // Ersetzen Sie durch Ihren Testschlüssel
    } else {
        \Pingpp\Pingpp::setApiKey('sk_live_your_live_key_here'); // Ersetzen Sie durch Ihren Live-Schlüssel
    }

    // Laden Sie Modelle
    $this->load->model('UserDao');
    $this->userDao = new UserDao();
    $this->load->model('ChargeDao');
    $this->chargeDao = new ChargeDao();
}
}

```

- Ersetzen Sie 'sk\_test\_your\_test\_key\_here' und 'sk\_live\_your\_live\_key\_here' durch Ihre tatsächlichen Pingpp API-Schlüssele.
- Stellen Sie sicher, dass UserDao und ChargeDao implementiert sind, um die Benutzerauthentifizierung und die Gebührenverwaltung zu handhaben (siehe Schritt 2).

**2. Unterstützende Modelle erstellen** Sie benötigen Modelle, um Benutzersitzungen zu verwalten und Gebührendetails zu speichern.

**UserDao.php** (vereinfachtes Beispiel):

```

<?php
class UserDao extends CI_Model {
    public function findUserBySessionToken($token) {
        $query = $this->db->get_where('users', ['session_token' => $token]);
        return $query->row(); // Gibt ein Benutzerobjekt oder null zurück
    }
}

```

**ChargeDao.php** (vereinfachtes Beispiel):

```

<?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);
    }
}

```

Erstellen Sie entsprechende Datenbanktabellen (`users` und `charges`) mit den entsprechenden Feldern.

**3. Gebührenerstellungsmethode implementieren** Fügen Sie eine Methode hinzu, um eine Pingpp-Gebühr zu erstellen und sie an den Client zurückzugeben. Dieses Beispiel verwendet Alipay PC Direct als Zahlungskanal.

```

public function createChargeThenResponse($amount, $subject, $body, $metaData, $user) {
    // Generieren Sie eine eindeutige Bestellnummer
    $orderNo = $this->generateOrderNo();

    // Setzen Sie die App-ID basierend auf der Umgebung
    $isLocalDebug = ENVIRONMENT === 'development';
    $appId = $isLocalDebug ? 'app_your_test_app_id' : 'app_your_live_app_id';

    // Holen Sie sich die IP-Adresse des Clients
    $ipAddress = $this->input->ip_address();
    if ($ipAddress === '::1') { // Behandeln Sie den lokalen Debug-Fall
        $ipAddress = '127.0.0.1';
    }

    // Erstellen Sie die Gebühr
    try {
        $charge = \Pingpp\Charge::create([
            'order_no' => $orderNo,
            'app' => ['id' => $appId],

```

```

'channel' => 'alipay_pc_direct', // Ändern Sie dies für andere Kanäle (z.B. 'wx' für WeChat)
'amount' => $amount, // In Cent (z.B. 1000 = 10 CNY)
'client_ip' => $ipAddress,
'currency' => 'cny',
'subject' => $subject,
'body' => $body,
'metadata' => $metaData,
'extra' => [
    'success_url' => 'http://yourdomain.com/payment/success' // Ersetzen Sie durch Ihre Erfolg-URL
]
);

// Speichern Sie die Gebührendetails in der Datenbank
$this->chargeDao->add($orderNo, $amount, $user->id, $ipAddress);

// Geben Sie das Gebührenobjekt als JSON zurück
$this->output
    ->set_status_header(200)
    ->set_content_type('application/json', 'utf-8')
    ->set_output(json_encode($charge));
} catch (\Pingpp\Error\Base $e) {
    log_message('error', 'Pingpp Charge Failed: ' . $e->getMessage());
    $this->response(['error' => 'Erstellung der Zahlung fehlgeschlagen'], 500);
}
}

private function generateOrderNo() {
    return uniqid(); // Einfache eindeutige ID; ersetzen Sie durch einen robusteren Generator, wenn nötig
}

```

- Ersetzen Sie 'app\_your\_test\_app\_id' und 'app\_your\_live\_app\_id' durch Ihre Pingpp App-IDs.
- Passen Sie 'success\_url' an Ihre Anwendungs-Erfolgseite-URL an.

**4. Checkout-Methode erstellen** Fügen Sie eine öffentliche Methode hinzu, um den Benutzer-Checkout zu verarbeiten, indem Sie die Gebührenerstellungsmethode aufrufen.

```

public function checkout_post() {
    // Überprüfen Sie, ob der Benutzer angemeldet ist
    $user = $this->getSessionUser();
    if (!$user) {

```

```

    $this->response(['error' => 'Benutzer nicht angemeldet'], 401);

    return;
}

// Beispiel-Bestelldetails

$amount = 1000; // 10 CNY (Beispiel; berechnen Sie dies dynamisch in einer echten App)
$subject = 'Bestellzahlung';
$body = 'Zahlung für Bestellung #123';
$metaData = ['order_id' => '123']; // Fügen Sie bestellenspezifische Daten hinzu

// Erstellen Sie eine Gebühr und antworten Sie
$this->createChargeThenResponse($amount, $subject, $body, $metaData, $user);
}

private function getSessionUser() {
    $token = $this->input->get_request_header('Authorization', TRUE); // Passen Sie dies basierend auf Ihrem ...
    if ($token) {
        return $this->userDao->findUserBySessionToken($token);
    }
    return null;
}

```

**5. Clientseitige Antwort verarbeiten** Die Methode `createChargeThenResponse` gibt ein JSON `charge`-Objekt zurück. Für `alipay_pc_direct` enthält es ein `credential`-Feld mit einer Zahlungs-URL. Verarbeiten Sie dies auf der Clientseite (z.B. mit JavaScript).

### Beispiel (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; // Weiterleitung zu Alipay
    } else {
        console.error('Zahlungsfehler:', charge);
    }
})
.catch(error => console.error('Fehler:', error));

```

**6. Erfolgseite** Erstellen Sie eine Erfolgseite (z.B. payment/success), um eine Bestätigung nach der Zahlung anzuzeigen. Aktualisieren Sie den Bestellstatus hier manuell, falls erforderlich, obwohl Webhooks für die Zuverlässigkeit empfohlen werden (siehe Schritt 7).

```
public function success_get() {  
    // Optional: Überprüfen Sie hier den Zahlungsstatus  
    $this->load->view('payment_success'); // Laden Sie eine Ansicht  
}
```

**7. (Optional) Webhooks verarbeiten** Pingpp sendet asynchrone Zahlungsstatusaktualisierungen über Webhooks. Richten Sie ein Endpunkt ein, um diese Benachrichtigungen zu empfangen und Ihre Datenbank zu aktualisieren.

```
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);  
}
```

- Fügen Sie eine updateStatus-Methode zu ChargeDao hinzu, um den Gebührenstatus zu aktualisieren.
- Konfigurieren Sie Ihre Webhook-URL im Pingpp-Dashboard (z.B. <http://yourdomain.com/payment/webhook>).

---

## Vollständiges Beispiel zur Verwendung

1. **Benutzeraktion:** Ein Benutzer meldet sich an und initiiert einen Checkout über eine POST-Anfrage an /payment/checkout.
2. **Serverantwort:** Der Server erstellt eine Gebühr und gibt ein JSON-Objekt mit Zahlungsdetails zurück.
3. **Client-Verarbeitung:** Die Frontend-App leitet den Benutzer zur Alipay-Zahlungsseite weiter.
4. **Zahlungsabschluss:** Nach der Zahlung wird der Benutzer zur Erfolg-URL weitergeleitet.
5. **Webhook-Aktualisierung:** Pingpp benachrichtigt Ihren Webhook-Endpunkt, und Sie aktualisieren den Gebührenstatus.

## Zusätzliche Hinweise

- **Zahlungskanäle:** Ersetzen Sie 'alipay\_pc\_direct' durch andere Kanäle (z.B. 'wx' für WeChat) nach Bedarf. Überprüfen Sie die Pingpp-Dokumentation für kanalbezogene extra-Parameter.
- **Fehlerbehandlung:** Verbessern Sie die Fehlerbehandlung für die Produktion (z.B. Wiederholungslogik, detaillierte Fehlermeldungen).
- **Sicherheit:** Validieren Sie die Benutzereingaben (\$amount usw.) und sichern Sie Ihre API-Endpunkte.

Dieses Beispiel bietet eine solide Grundlage für die Integration von Pingpp in Ihre CodeIgniter-Anwendung, die für verschiedene Anwendungsfälle wie E-Commerce oder Spenden angepasst werden kann.