

# Intégration du SDK Ping++ avec le Framework CodeIgniter

Voici un exemple complet, étape par étape, de l'intégration de **Pingpp** (Ping++), un SDK de traitement des paiements, dans une application PHP en utilisant le framework CodeIgniter, basé sur la classe `BaseController` fournie. Ce guide suppose que vous avez une configuration de base de CodeIgniter et se concentre sur l'intégration de Pingpp pour le traitement des paiements, comme permettre aux utilisateurs de payer via Alipay PC Direct (bien que vous puissiez l'adapter pour d'autres canaux).

---

## Prérequis

- Compte Pingpp** : Inscrivez-vous sur Ping++ et obtenez votre **clé API** et votre **ID d'application**. Vous aurez une clé de test (`sk_test_...`) pour le développement et une clé de production (`sk_live_...`).
- SDK PHP Pingpp** : Installez le SDK PHP Pingpp via Composer :

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

Assurez-vous que le SDK est chargé automatiquement dans votre application CodeIgniter (par exemple, placez-le dans `application/libraries` ou configurez le chargement automatique de Composer).

- Configuration CodeIgniter** : Vous devez avoir une application CodeIgniter fonctionnelle avec un contrôleur et des modèles de base de données pour les utilisateurs et les charges.
- 

## Intégration Étape par Étape

- Configurer le Contrôleur** Créez un contrôleur (par exemple, `PaymentController.php`) qui étend un contrôleur de base similaire à `BaseController`. Configurez la clé API Pingpp dans le constructeur en fonction de votre environnement (développement ou production).

```
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
require_once APPPATH . '/libraries/REST_Controller.php';  
require_once APPPATH . '/vendor/autoload.php'; // Supposant le chargement automatique de Composer  
  
class PaymentController extends REST_Controller {  
  
    public $userDao;
```

```

public $chargeDao;

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

    // Définir la clé API Pingpp en fonction de l'environnement
    $isLocalDebug = ENVIRONMENT === 'development'; // Environnement CodeIgniter
    if ($isLocalDebug) {
        \Pingpp\Pingpp::setApiKey('sk_test_your_test_key_here'); // Remplacez par votre clé de test
    } else {
        \Pingpp\Pingpp::setApiKey('sk_live_your_live_key_here'); // Remplacez par votre clé de production
    }

    // Charger les modèles
    $this->load->model('UserDao');
    $this->userDao = new UserDao();
    $this->load->model('ChargeDao');
    $this->chargeDao = new ChargeDao();
}
}

```

- Remplacez 'sk\_test\_your\_test\_key\_here' et 'sk\_live\_your\_live\_key\_here' par vos clés API Pingpp réelles.
- Assurez-vous que UserDao et ChargeDao sont implémentés pour gérer l'authentification des utilisateurs et le stockage des charges (voir Étape 2).

**2. Créer les Modèles de Support** Vous aurez besoin de modèles pour gérer les sessions utilisateur et stocker les détails des charges.

**UserDao.php** (exemple simplifié) :

```

<?php
class UserDao extends CI_Model {
    public function findUserBySessionToken($token) {
        $query = $this->db->get_where('users', ['session_token' => $token]);
        return $query->row(); // Retourne un objet utilisateur ou null
    }
}

```

**ChargeDao.php** (exemple simplifié) :

```

<?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);
    }
}

```

Créez des tables de base de données correspondantes (`users` et `charges`) avec les champs appropriés.

**3. Implémenter la Méthode de Création de Charge** Ajoutez une méthode pour créer une charge Pingpp et la renvoyer au client. Cet exemple utilise Alipay PC Direct comme canal de paiement.

```

public function createChargeThenResponse($amount, $subject, $body, $metaData, $user) {
    // Générer un numéro de commande unique
    $orderNo = $this->generateOrderNo();

    // Définir l'ID de l'application en fonction de l'environnement
    $isLocalDebug = ENVIRONMENT === 'development';
    $appId = $isLocalDebug ? 'app_your_test_app_id' : 'app_your_live_app_id';

    // Obtenir l'adresse IP du client
    $ipAddress = $this->input->ip_address();
    if ($ipAddress === '::1') { // Gérer le cas de débogage local
        $ipAddress = '127.0.0.1';
    }

    // Créer la charge
    try {
        $charge = \Pingpp\Charge::create([
            'order_no' => $orderNo,
            'app' => ['id' => $appId],
            'channel' => 'alipay_pc_direct', // Changez ceci pour d'autres canaux (par exemple, 'wx' pour WeChat)
        ]);
    }
}

```

```

'amount' => $amount, // En centimes (par exemple, 1000 = 10 CNY)
'client_ip' => $ipAddress,
'currency' => 'cny',
'subject' => $subject,
'body' => $body,
'metadata' => $metaData,
'extra' => [
    'success_url' => 'http://yourdomain.com/payment/success' // Remplacez par votre URL de succès
]
]);

// Stocker les détails de la charge dans la base de données
$this->chargeDao->add($orderNo, $amount, $user->id, $ipAddress);

// Retourner l'objet de charge en 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', 'Échec de la charge Pingpp : ' . $e->getMessage());
    $this->response(['error' => 'La création du paiement a échoué'], 500);
}
}

private function generateOrderNo() {
    return uniqid(); // ID unique simple ; remplacez par un générateur plus robuste si nécessaire
}

```

- Remplacez 'app\_your\_test\_app\_id' et 'app\_your\_live\_app\_id' par vos IDs d'application Pingpp.
- Ajustez 'success\_url' à l'URL de la page de succès de votre application.

**4. Créer une Méthode de Caisse** Ajoutez une méthode publique pour gérer le passage en caisse de l'utilisateur, en appelant la méthode de création de charge.

```

public function checkout_post() {
    // Vérifier si l'utilisateur est connecté
    $user = $this->getSessionUser();
    if (!$user) {
        $this->response(['error' => 'Utilisateur non connecté'], 401);
    }
}

```

```

    return;
}

// Exemple de détails de commande
$amount = 1000; // 10 CNY (exemple ; calculez dynamiquement dans une application réelle)
$subject = 'Paiement de commande';
$body = 'Paiement pour la commande #123';
$metaData = ['order_id' => '123']; // Attacher des données spécifiques à la commande

// Créer une charge et répondre
$this->createChargeThenResponse($amount, $subject, $body, $metaData, $user);
}

private function getSessionUser() {
    $token = $this->input->get_request_header('Authorization', TRUE); // Ajustez en fonction de votre méthode
    if ($token) {
        return $this->userDao->findUserBySessionToken($token);
    }
    return null;
}

```

**5. Gérer la Réponse du Client** La méthode `createChargeThenResponse` retourne un objet JSON `charge`. Pour `alipay_pc_direct`, il inclut un champ `credential` avec une URL de paiement. Gérez cela côté client (par exemple, avec JavaScript).

#### Exemple (JavaScript Frontend) :

```

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; // Rediriger vers Alipay
    } else {
        console.error('Erreur de paiement :', charge);
    }
})
.catch(error => console.error('Erreur :', error));

```

**6. Page de Succès** Créez une page de succès (par exemple, payment/success) pour afficher une confirmation après le paiement. Mettez à jour le statut de la commande manuellement ici si nécessaire, bien que les webhooks soient recommandés pour la fiabilité (voir Étape 7).

```
public function success_get() {  
    // Optionnellement vérifier le statut du paiement ici  
    $this->load->view('payment_success'); // Charger une vue  
}
```

**7. (Optionnel) Gérer les Webhooks** Pingpp envoie des mises à jour de statut de paiement asynchrones via des webhooks. Configurez un point de terminaison pour recevoir ces notifications et mettre à jour votre base de données.

```
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);  
}
```

- Ajoutez une méthode updateStatus à ChargeDao pour mettre à jour le statut de la charge.
- Configurez votre URL de webhook dans le tableau de bord Pingpp (par exemple, <http://yourdomain.com/payment/>)

---

## Exemple d'Utilisation Complète

- Action de l'Utilisateur** : Un utilisateur se connecte et initie un passage en caisse via une requête POST à /payment/checkout.
- Réponse du Serveur** : Le serveur crée une charge et retourne un objet JSON avec les détails de paiement.
- Gestion Client** : Le frontend redirige l'utilisateur vers la page de paiement Alipay.
- Achèvement du Paiement** : Après le paiement, l'utilisateur est redirigé vers l'URL de succès.
- Mise à Jour du Webhook** : Pingpp notifie votre point de terminaison de webhook, et vous mettez à jour le statut de la charge.

---

## Notes Supplémentaires

- **Canaux de Paiement** : Remplacez 'alipay\_pc\_direct' par d'autres canaux (par exemple, 'wx' pour WeChat) selon les besoins. Consultez la documentation Pingpp pour les paramètres `extra` spécifiques au canal.
- **Gestion des Erreurs** : Améliorez la gestion des erreurs pour la production (par exemple, logique de nouvelle tentative, messages d'erreur détaillés).
- **Sécurité** : Validez les entrées utilisateur (`$amount`, etc.) et sécurisez vos points de terminaison API.

Cet exemple fournit une base solide pour intégrer Pingpp dans votre application CodeIgniter, adaptable à divers cas d'utilisation comme le commerce électronique ou les dons.