

整合 Ping++ SDK 至 CodeIgniter 框架

以下是如何將 **Pingpp** (Ping++) 支付處理 SDK 集成到使用 CodeIgniter 框架的 PHP 應用程式的完整、逐步範例。本指南假設您已經有一個基本的 CodeIgniter 設置，並專注於集成 Pingpp 進行支付處理，例如啟用用戶通過支付寶 PC 直接支付（您可以將其適應其他渠道）。

前提條件

1. **Pingpp 帳戶**：在 Ping++ 註冊並獲取您的 **API 密鑰** 和 **應用程式 ID**。您將擁有一個測試密鑰 (`sk_test_...`) 用於開發和一個實時密鑰 (`sk_live_...`) 用於生產。
 2. **Pingpp PHP SDK**：通過 Composer 安裝 Pingpp PHP SDK：

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


確保 SDK 在您的 CodeIgniter 應用程式中自動加載（例如，將其放在 `application/libraries` 或配置 Composer 自動加載）。
 3. **CodeIgniter 設置**：您應該有一個正常運行的 CodeIgniter 應用程式，其中包含用戶和費用的控制器和數據庫模型。
-

逐步集成

1. 設置控制器 創建一個控制器（例如 `PaymentController.php`），它擴展了一個類似於 `BaseController` 的基礎控制器。在構造函數中根據您的環境（開發或生產）配置 Pingpp API 密鑰。

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

require_once APPPATH . '/libraries/REST_Controller.php';
require_once APPPATH . '/vendor/autoload.php'; // 假設 Composer 自動加載

class PaymentController extends REST_Controller {

    public $userDao;
    public $chargeDao;

    public function __construct() {
```

```

parent::__construct();

// 根據環境設置 Pingpp API 密鑰
$isLocalDebug = ENVIRONMENT === 'development'; // CodeIgniter 環境
if ($isLocalDebug) {
    \Pingpp\Pingpp::setApiKey('sk_test_your_test_key_here'); // 替換為您的測試密鑰
} else {
    \Pingpp\Pingpp::setApiKey('sk_live_your_live_key_here'); // 替換為您的實時密鑰
}

// 加載模型
$this->load->model('UserDao');
$this->userDao = new UserDao();
$this->load->model('ChargeDao');
$this->chargeDao = new ChargeDao();
}
}

```

- 替換 'sk_test_your_test_key_here' 和 'sk_live_your_live_key_here' 為您的實際 Pingpp API 密鑰。
- 確保 UserDao 和 ChargeDao 已實現以處理用戶身份驗證和費用存儲（請參見步驟 2）。

2. 創建支持模型 您需要模型來管理用戶會話並存儲費用詳細信息。

UserDao.php（簡化範例）：

```

<?php
class UserDao extends CI_Model {
    public function findUserBySessionToken($token) {
        $query = $this->db->get_where('users', ['session_token' => $token]);
        return $query->row(); // 返回用戶對象或 null
    }
}

```

ChargeDao.php（簡化範例）：

```

<?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);
}
}

```

創建相應的數據庫表（users 和 charges），並具有適當的字段。

3. 實現費用創建方法 添加一個方法來創建 Pingpp 費用並將其返回給客戶端。此範例使用支付寶 PC 直接作為支付渠道。

```

public function createChargeThenResponse($amount, $subject, $body, $metaData, $user) {
    // 生成唯一訂單號
    $orderNo = $this->generateOrderNo();

    // 根據環境設置應用程式 ID
    $isLocalDebug = ENVIRONMENT === 'development';
    $appId = $isLocalDebug ? 'app_your_test_app_id' : 'app_your_live_app_id';

    // 获取客戶端 IP 地址
    $ipAddress = $this->input->ip_address();
    if ($ipAddress === ':::1') { // 处理本地调试情况
        $ipAddress = '127.0.0.1';
    }

    // 創建費用
    try {
        $charge = \Pingpp\Charge::create([
            'order_no' => $orderNo,
            'app' => ['id' => $appId],
            'channel' => 'alipay_pc_direct', // 更改此處以使用其他渠道（例如，'wx' 表示微信）
            'amount' => $amount, // 以分計（例如，1000 = 10 CNY）
            'client_ip' => $ipAddress,
            'currency' => 'cny',
            'subject' => $subject,
            'body' => $body,
            'metadata' => $metaData,

```

```

        'extra' => [
            'success_url' => 'http://yourdomain.com/payment/success' // 替換為您的成功 URL
        ]
    });

    // 將費用詳細信息存儲在數據庫中
    $this->chargeDao->add($orderNo, $amount, $user->id, $ipAddress);

    // 將費用對象作為 JSON 返回
    $this->output
        ->set_status_header(200)
        ->set_content_type('application/json', 'utf-8')
        ->set_output(json_encode($charge));
    } catch (\Pingpp\Error\Base $e) {
        log_message('error', 'Pingpp Charge Failed: ' . $e->getMessage());
        $this->response(['error' => '支付創建失敗'], 500);
    }
}

private function generateOrderNo() {
    return uniqid(); // 簡單的唯一 ID；如果需要，請替換為更強大的生成器
}

```

- 替換 'app_your_test_app_id' 和 'app_your_live_app_id' 為您的 Pingpp 應用程式 ID。
- 調整 'success_url' 為您應用程式的成功頁面 URL。

4. 創建結帳方法 添加一個公共方法來處理用戶結帳，調用費用創建方法。

```

public function checkout_post() {
    // 檢查用戶是否已登錄
    $user = $this->getSessionUser();
    if (!$user) {
        $this->response(['error' => '用戶未登錄'], 401);
        return;
    }

    // 示例訂單詳細信息
    $amount = 1000; // 10 CNY (示例；在實際應用中動態計算)
    $subject = '訂單支付';
    $body = '訂單 #123 的支付';
}

```

```

$metadata = ['order_id' => '123']; // 附加訂單特定數據

// 創建費用並響應
$this->createChargeThenResponse($amount, $subject, $body, $metadata, $user);
}

private function getSessionUser() {
    $token = $this->input->get_request_header('Authorization', TRUE); // 根據您的身份驗證方法調整
    if ($token) {
        return $this->userDao->findUserBySessionToken($token);
    }
    return null;
}

```

5. 處理客戶端響應 createChargeThenResponse 方法返回一個 JSON charge 對象。對於 alipay_pc_direct，它包括一個包含支付 URL 的 credential 字段。在客戶端（例如，使用 JavaScript）處理此內容。

範例（前端 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; // 重定向到支付寶
    } else {
        console.error('支付錯誤:', charge);
    }
})
.catch(error => console.error('錯誤:', error));

```

6. 成功頁面 創建一個成功頁面（例如 payment/success），以顯示支付後的確認。如果需要，請在此手動更新訂單狀態，但建議使用 Webhook 以確保可靠性（請參見步驟 7）。

```

public function success_get() {
    // 可選地驗證支付狀態
    $this->load->view('payment_success'); // 加載視圖
}

```

7. (可選) 處理 Webhook Pingpp 通過 Webhook 發送異步支付狀態更新。設置一個端點來接收這些通知並更新您的數據庫。

```
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);  
}
```

- 向 ChargeDao 添加一個 updateStatus 方法來更新費用狀態。
- 在 Pingpp 儀表板中配置您的 Webhook URL（例如，<http://yourdomain.com/payment/webhook>）。

完整範例使用

1. **用戶操作**：用戶登錄並通過向 `/payment/checkout` 發送 POST 請求啟動結帳。
2. **服務器響應**：服務器創建費用並返回包含支付詳細信息的 JSON 對象。
3. **客戶端處理**：前端將用戶重定向到支付寶支付頁面。
4. **支付完成**：支付後，用戶將被重定向到成功 URL。
5. **Webhook 更新**：Pingpp 通知您的 Webhook 端點，您更新費用狀態。

額外說明

- **支付渠道**：將 `'alipay_pc_direct'` 替換為其他渠道（例如，`'wx'` 表示微信）以滿足需求。請參閱 Pingpp 文檔以獲取渠道特定的 `extra` 參數。
- **錯誤處理**：增強生產環境的錯誤處理（例如，重試邏輯、詳細錯誤消息）。
- **安全性**：驗證用戶輸入（`$amount` 等）並保護您的 API 端點。

此範例為將 Pingpp 集成到您的 CodeIgniter 應用程式提供了堅實的基礎，可適應各種用例，如電子商務或捐款。