



DOI [10.28925/2663-4023.2022.17.6575](https://doi.org/10.28925/2663-4023.2022.17.6575)

УДК 004.77

Ступінь Андрій Петрович,

студент факультету інформаційних технологій і математики

Волинський національний університет імені Лесі Українки, м. Луцьк, Україна

stupin.andrij2018@vnu.edu.ua

Глинчук Людмила Ярославівна

кандидат фіз.-мат. наук, доцент кафедри комп'ютерних наук та кібербезпеки

Волинський національний університет імені Лесі Українки, Луцьк, Україна

ORCID ID:0000-0002-8943-9604

hlynchuk.ludmila@vnu.edu.ua

Гришанович Тетяна Олександрівна

кандидат фіз.-мат. наук, доцент кафедри комп'ютерних наук та кібербезпеки

Волинський національний університет імені Лесі Українки, Луцьк, Україна

ORCID ID:0000-0002-3595-6964

hryshanovych.tatiana@vnu.edu.ua

АЛГОРИТМ ПІДКЛЮЧЕННЯ СЕРВІСІВ ОНЛАЙН-ОПЛАТ FONDU, LIQPAY ТА ЇХ РЕАЛІЗАЦІЯ

Анотація. У роботі описано алгоритми підключення двох найпоширеніших в Україні платіжних сервісів – Fondy та LiqPay – до програмних засобів (сайтів, мобільних додатків із підключенням до Інтернет). Наразі така тематика є досить актуальною, оскільки здійснюється активний перехід не тільки вітчизняної економіки, але і світової, до безготівкових розрахунків. А це, у свою чергу, ставить виклики не тільки перед економістами, а й перед фахівцями з інформаційних технологій. Зараз важко уявити сайт-каталог або сайт послуг без онлайн-оплати придбаного товару. За допомогою безготівкових розрахунків можна оплатити практично все: від товарів до комунальних послуг та адміністративних штрафів. Метою нашого дослідження є розробка алгоритму під'єднання сервісів онлайн-оплат до веб-сайтів та мобільних додатків, підключених до Інтернету, та програмна реалізація такого алгоритму. Кожен сервіс для здійснення онлайн-оплат має документацію та набір із засобів розробки, утиліт і документації, який дозволяє створювати прикладні програми за визначеною технологією або для певної платформи (SDK). SDK, як правило, мають тестові реквізити та тестові ключі для можливості проведення тестових оплат. Аналіз схеми, за якою відбувається списання коштів з клієнта, показує, що для серверу сайту необхідно реалізувати два основних методи, а саме: створення веб-форми для переходу на наступний етап заповнення реквізитів оплати та прийом вебхуків (механізм надсилання запитів під час виникнення в системі подій) із сервера сервісу. У роботі нами описано метод для генерування веб-форми оплати та метод для опрацювання вебхуків. Запропоноване програмне рішення – це лише обгортка, яка полегшує взаємодію сервісу оплати із кодом тим, що надає можливість підключити декілька сервісів і об'єднати їх під один інтерфейс, що в свою чергу забирає пряму залежність від конкретної реалізації SDK. Також є сенс писати подібний інтерфейс і для сайтів із одним сервісом оплати, оскільки нерідко бувають замовники, які з певних причин вирішили замінити сервіс оплати на своєму сайті, а якщо там буде реалізовано подібний інтерфейс, то заміна сервісу оплати практично зводиться до реалізації декількох методів із інтерфейсу з новим SDK, що, в свою чергу, не буде нести змін в уже робочу логіку із замовленнями.

Ключові слова: сервіс онлайн-оплати, Fondy, LiqPay, вебхук, безготівкова оплата, онлайн-оплата.



ВСТУП

На думку багатьох експертів у сфері економіки та інформаційних технологій, людство переходить на безготівковий розрахунок із використанням електронних сервісів [5, 9, 10, 15]. Національний Банк України підтримує рух економіки до безготівкових розрахунків і сприяє розвитку платіжної інфраструктури. Безготівкові розрахунки – швидкий, зручний і безпечний спосіб оплатити товари та послуги. Розширення цієї сфери розрахунків сприяє прозорості платежів та економічному росту країни. В Україні правила здійснення безготівкових розрахунків встановлюються Національним Банком України. Ним також гарантується безпека таких видів платежів та їх надійність. Національний Банк встановлює правила для здійснення безготівкових розрахунків та сприяє їх безпеці та надійності. Фізичні особи та організації можуть здійснюють безготівкові розрахунки в Україні, використовуючи розрахункові документи, електронні платіжні засоби, електронні гроші та документарні операції та інструменти. [1]

Станом на травень 2022 року в Україні було здійснено 637 млн. безготівкових операцій на загальну суму 573,7 млрд. гривень, що на 5,6% більше, ніж у 2021 році [3]. І ця тенденція продовжить зростати, оскільки органами державної влади активно підтримується політика підвищення прозорості економіки. Також “зростає фінансова грамотність населення – люди хочуть розібратися в фінансових і банківських послугах, тому виграє той, хто швидше заговорить з клієнтом на їхній мові і відповість на всі питання”, – вважає маркетолог мережі терміналів самообслуговування ІВох [2]. На даний час користувачеві важко уявити сайт-каталог або сайт послуг без онлайн-оплати придбаного товару. За допомогою безготівкових розрахунків можна оплатити практично все: від товарів до комунальних послуг та адміністративних штрафів.

Постановка проблеми. Багато хто вважає, що підключення онлайн-оплати – це доволі складний і незрозумілий процес. Насправді це не так. І метою цієї статті є доведення цього факту, адже сервіси оплат самі зацікавлені в тому, щоб саме їх сервіс підключили якомога більша кількість сайтів.

Підключення онлайн-оплати на сайті дозволяє здійснювати оплату відразу в браузері через інтернет-підключення. Сьогодні існує велика кількість таких засобів, але ми звернемось до тих, які найчастіше використовуються в нашій країні – це платіжні сервіси Fondy та LiqPay. Ці сервіси представлені на ринку досить давно, швидко адаптуються під його потреби та реагують на зміни в технологіях. До того ж, на думку голови правління Української асоціації Фінтех та інноваційних компаній Ростислава Дюка, бізнес в Україні тяжіє до інтеграції – до використання перевірених засобів, які надають широкий спектр можливостей [2].

Аналіз останніх досліджень і публікацій. Сервіс LiqPay було засновано у 2008 році і зараз ним користуються понад 25 000 проєктів із 120 країн світу. Це сервіс для миттєвих оплат онлайн за допомогою карт Visa і Mastercard [11]. Він також підтримує оплату за допомогою Google Pay, Apple Pay, Privat24, MasterPass. Fondy – це платіжна платформа, основним завданням якої є надання сервісу обробки платежів онлайн на сайтах, в мобільних додатках і пристроях різних типів, підключених до Інтернет. Також Fondy пропонує хмарне white-label рішення для банків, IPSP і процесингових центрів, що надають послуги інтернет-еквайрингу [14] і фінансових розрахунків. Ця платформа підтримує інтернет-банкінг Privat24, оплату Visa, Mastercard, Apple Pay, Google Pay у всіх країнах світу, SEPA, Swift – в Європі. Протягом тривалого часу ці сервіси від ПриватБанк були найбільш поширеними. У вересні 2021 року один із співзасновників Monobank Олег Гороховський анонсував появу сервісу для оплати послуг онлайн [7]. Сервіс запрацював у 2022 році [6] і наразі із ним працює досить широкий спектр проєктів: від індивідуальних міні сайтів, наприклад, виготовлення крафтової продукції у сфері



громадського харчування, до системи донатів Stand with Ukraine (<https://helpua.bank.gov.ua/>), яка була відкрита Національним Банком України.

Метою дослідження є опис алгоритму підключення сервісів онлайн-оплат до веб-сайтів та мобільних додатків, підключених до Інтернету, та програмна реалізація такого алгоритму.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Як правило, кожен сервіс для здійснення онлайн-оплат має документацію та SDK. SDK (Software Development Kit) – набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи [12, 13]

Загалом, як відбувається онлайн-оплата з точки зору сервера сервісу та сервера сайту? В основному в кожного сервісу є схема оплат за якою відбувається списання грошей з клієнта, подібна до:

1. Клієнт наповнює кошик, замовляє послугу, тобто здійснює дію, яка вимагає онлайн-оплати.
2. Сервер сайту формує форму для онлайн-оплати.
3. Клієнт переходить на форму онлайн-оплати, яку сформував сайт, та, в свою чергу, перенаправляє клієнта безпосередньо до сервісу оплати.
4. Клієнт заповнює реквізити для оплати.
5. Сервіс для онлайн-оплати перенаправляє клієнта назад на сайт.
6. Сервіс для онлайн-оплати відправляє вебхук при зміні статусу оплати.

Вебхук – це механізм надсилання запитів під час виникнення в системі подій [4]. Під подією розуміють зміну стану системи. Наприклад, замовлення було оплачено успішно, отож відбудеться подія і відповідно буде надіслано запит. У випадку сервісів онлайн-оплат, це буде звичайний HTTP-запит до сервера сайту, в якого було сплачено замовлення, з інформацією про оплату.

Також кожен сервіс захищає свої дані і зазвичай для компанії видає два ключі. Перший публічний, можна показувати іншим, але не бажано, другий приватний (секретний), який в жодному випадку нікому не можна показувати. Механізм їхньої роботи схожий на алгоритм шифрування RSA [8]. SDK для сервісів оплат, як правило, мають тестові реквізити та тестові ключі для можливості проведення тестових оплат, тому завжди під час підключення та тестування використовуйте їх.

Отже, переглянувши схему, за якою відбувається списання коштів з клієнта, можна зробити висновок, що для серверу сайту необхідно реалізувати два основних методи, а саме: створення веб-форми для переходу на наступний етап заповнення реквізитів оплати та прийом вебхуків із сервера сервісу.

Для початку потрібно підключити SDK до проекту. Завантажити SDK для сервісу Fondy можна з GitHub за посиланням <https://github.com/cloudipsp/php-sdk-v2>, або слідувати інструкції підключення у документації до цього SDK <https://cloudipsp.github.io/php-docs/>. У випадку з LiqPay рекомендовано завантажити SDK з GitHub за посиланням <https://github.com/liqpay/sdk-php>.

Далі потрібно визначити інтерфейс для сервісів оплат. Будемо реалізовувати метод для генерування веб-форми оплати та метод для опрацювання вебхуків. Додатковими ж будуть методи для перевірки чи успішний статус, тобто, чи за даним статусом оплата дійсно була успішно проведена, та метод для отримання успішних статусів (рисунок 1).

Назвемо інтерфейс PaymentMethodInterface та вкажемо 4 функції:

1. `generateForm ($order_id, $amount, $description, $redirect_url, $webhook_url, $currency)` – формування веб-форми для оплати замовлення.

Параметри:

- `order_id` – номер замовлення, яке потрібно оплатити;
- `amount` – сума замовлення, копійки вказуються через кому;
- `description` – опис замовлення;
- `redirect_url` – url, на який перенаправити користувача після закінчення оплати;
- `webhook_url` – url, на який буде відправлено вебхук;
- `currency` – валюта у форматі UAH, EUR, USD та інші.

2. `webhook ($data_from_request, $callback_pay_success, $callback_pay_fail)` - метод, який буде опрацьовувати вебхуки.

Параметри:

- `data_from_request` – дані про замовлення, які були відправлені на url вебхука;
- `callback_pay_success` – функція, яку викликати у разі успішної оплати;
- `callback_pay_fail` – функція, яку викликати у разі неуспішної оплати.

3. `StatusSuccess ($status)` – перевірка чи вказаний статус оплати є успішним.

Параметри:

- `status` – статус оплати.

4. `getSuccessStatuses ()` – отримати усі статуси із успішною оплатою.

```
5 interface PaymentMethodInterface
6 {
7     public function generateForm($order_id, $amount, $description, $redirect_url, $webhook_url, $currency);
8
9     public function webhook(array $data_from_request, callable $callback_pay_success, callable $callback_pay_fail) : bool;
10
11     public function statusSuccess($status) : bool;
12
13     public function getSuccessStatuses() : array;
14
15 }
```

Рис. 1. Вигляд інтерфейсу для сервісів оплат

Наступним кроком зробимо свої власні класи (обгортки) над уже готовими SDK, щоб ті відповідали щойно створеному інтерфейсу. Але перед цим детальніше розглянемо наші SDK.

SDK `LiqPay` містить у собі декілька основних функцій, на які варто звернути увагу:

- `construct` – ініціалізація головного класу SDK, приймає публічний ключ, приватний ключ;
- `snb_form` – власне основний метод, який будемо використовувати для генерування вебформи для оплати, приймає на вхід масив даних замовлення для його формування;
- `verify_sign` – для перевірки підпису, тобто за допомогою цього методу перевіряється запит на підробку;
- `decode_params` – для декодування параметрів запиту, як правило, використовується у зв'язці з методом `verify_sign`.

Відтак реалізуємо клас (обгортку) для сервісу `LiqPay`. Для цього створимо новий файл, назвемо його `LiqPayService` та наслідуємо інтерфейс `PaymentMethodInterface` (рисунок 2).

```
33 use InvalidArgumentException;  
34 use LiqPay\LiqPay;  
35 use Log;  
36  
37 class LiqPayService implements PaymentMethodInterface  
38 {
```

Рис. 2. Лістинг класу для сервісу LiqPay

Реалізуємо функцію для формування веб-форми для оплати з інтерфейсу PaymentMethodInterface, де з SDK LiqPay використовуємо функцію `cnb_form`.

Варто зауважити, що для цього необхідно створити інстанс класу LiqPay з SDK та викликати метод із створеного інстансу. Під час створення інстансу з SDK необхідно буде вказати публічний та приватний ключі.

Відповідно до входних параметрів методу `generateForm()` із інтерфейсу PaymentMethodInterface виклик набуде вигляду зображеного на рисунку 3.

```
297 public function generateForm($order_id, $amount, $description, $redirect_url, $webhook_url, $currency = 'UAH')  
298 {  
299     $liqPaySDK = new LiqPay('your public key', 'your private key');  
300     return $liqPaySDK->cnb_form(array(  
301         'action'      => 'pay',  
302         'amount'     => $amount,  
303         'currency'   => $currency,  
304         'description' => $description,  
305         'order_id'  => $order_id,  
306         'version'   => '3',  
307         'result_url' => $redirect_url,  
308         'server_url' => $webhook_url,  
309     ));  
310 }  
311 }
```

Рис. 3. Реалізація методу `generateFrom` для класу `LiqPayService`

Перед реалізацією методу для обробки вебхуків (webhook із PaymentMethodInterface), необхідно реалізувати методи `getSuccessStatuses` та `statusSuccess` із цього ж інтерфейсу.

З документації підключення LiqPay можна побачити, що успішний статус має назву `success`. Окрім нього є ще один статус із назвою `wait_ассерт`, який означає, що кошти з клієнта списані, але магазин ще не пройшов перевірку, тобто при цьому статусі із клієнта гроші списані, отже необхідно його опрацювати як успішний. У випадку, якщо магазин не пройде перевірку, кошти будуть повернуті і на сервер будуть приходити вебхуки із новими статусами замовлень.

Отож, метод `getSuccessStatuses()` буде повертати масив усіх статусів, де з клієнта списало кошти. А метод `statusSuccess()` буде перевіряти чи статус належить масиву із методу `getSuccessStatuses()` (рисунок 4).

```
353 public function getSuccessStatuses() : array  
354 {  
355     return ['success', 'wait_ассерт'];  
356 }  
357  
358 public function statusSuccess($status) : bool  
359 {  
360     return in_array($status, $this->getSuccessStatuses());  
361 }  
362 }
```

Рис. 4. Реалізація методів `getSuccessStatuses()` і `statusSuccess()`

Тепер реалізуємо функцію для обробки вебхуків (webhook із PaymentMethodInterface). Для цього, так як і в попередній реалізації, нам необхідний інстанс класу LiqPay з SDK. Але спочатку згадаємо, які параметри приймає ця функція:

- data_from_request – дані про замовлення, які були відправлені на url вебхука;
- callback_pay_success – функція, яку слід викликати у разі успішної оплати;
- callback_pay_fail – функція, яку викликати у разі неуспішної оплати.

За допомогою цього інстансу перевіряємо вхідні параметри, які знаходяться у змінній data_from_request, на наявність підроблення функцією verify_sign. Після чого розкодуємо параметри за допомогою методу decode_params(). У результаті після розкодування параметрів можна легко отримати статус оплати і звірити його із статусами, де відбулося списання коштів, метод isSuccess() із інтерфейсу PaymentMethodInterface, та викликати одну з переданих функцій, змінні callback_pay_success і callback_pay_fail, в залежності від результату методу isSuccess. Для виклику функцій у перемінних callback_pay_success і callback_pay_fail, можна застосувати php функцію call_user_func() (рисунок 5)

SDK Fondy наділений більшою кількістю можливостей, але нас будуть цікавити лише три класи:

1. Cloudipsp\Configuration – для вказання приватного та публічного ключа;
2. Cloudipsp\Checkout – для створення вебформи для онлайн-оплати;
3. Cloudipsp\Result – для обробки запитів, що приходять із серверів Fondy.

Спочатку реалізуємо клас (обгортку) для сервісу Fondy. Для цього створимо новий файл, назвемо його FondyService та наслідуюмо інтерфейс PaymentMethodInterface() (рисунок 6).

Далі проводимо аналогічні дії, які проводили для LiqPayService. Реалізуємо метод generateForm, але з відмінністю, що тут замість створення одного інстансу класу, як це було у випадку LiqPay, ми викликаємо статичні методи у класів із SDK Fondy. За допомогою статичних методів setMerchantId() і setSecretKey(), викликаних до класу Configuration, вказуємо публічний (merchant_id) та приватний ключі (secret_key). І повертаємо результат виконання статичного методу form класу Checkout, де передаємо масив із параметрами для формування оплати (рисунок 7)

```
119 public function webhook(array $data from request, callable $callback pay success, callable $callback pay fail) : bool
120 {
121     $liqPaySDK = new LiqPay('your public key', 'your private key');
122
123
124     if(isset($data from request['data']) && isset($data from request['signature'])) {
125         // check is sign ok
126         $signOk = $liqPaySDK->verify_sign($data from request['data'] ?? null, $data from request['signature'] ?? null);
127     } else {
128         // if not set the keys data or signature return false
129         return false;
130     }
131
132
133     if($signOk){
134         // getting payment params
135         $info = $liqPaySDK->decode_params($data from request['data']);
136         $status = $info['status'];
137
138         if( $this->isSuccess($status) ) {
139             // liqpay webhook status SUCCESS
140             call_user_func($callback pay success, $info['order_id'], $status, $info, $data from request['signature']);
141         } else {
142             // liqpay webhook status FAIL
143             call_user_func($callback pay fail, $info['order_id'], $status, $info, $data from request['signature']);
144         }
145
146         return true;
147     } else {
148         return false;
149     }
150 }
```

Рис. 5. Реалізація методу webhook() для класу LiqPayService

```
33 use InvalidArgumentException;
34 use Log;
35
36
37 use Cloudipsp\Checkout;
38 use Cloudipsp\Configuration;
39 use Cloudipsp\Order;
40 use Cloudipsp\Result\Result;
41
42 class FondyService implements PaymentMethodInterface
43 {
44
```

Рис. 6. Вигляд класу для сервісу Fondy

```
45 public function generateForm($order_id, $amount, $description, $redirect_url, $webhook_url, $currency = 'UAH')
46 {
47     Configuration::setMerchantId('your public key (merchant id)');
48     Configuration::setSecretKey('your private key (secret key)');
49     return Checkout::form([
50         'order_id' => $order_id,
51
52         'response_url' => $redirect_url,
53         'server_callback_url' => $webhook_url,
54
55         'lifetime' => 60*30,
56         'order_desc' => $description,
57         'currency' => $currency,
58         'amount' => $amount*100,
59     ]);
60 }
```

Рис. 7. Реалізація методу generateFrom() для класу FondyService

Також знаходимо у документації назви статусів із успішними платежами для Fondy – це статус із назвою approved і реалізуємо методи getSuccessStatuses() і statusSuccess() аналогічно тим, що були у LiqPayService, лиш із відмінністю у назвах статусів (рисунок 8).

```
78 public function getSuccessStatuses() : array
79 {
80     return ['approved'];
81 }
82
83 public function statusSuccess($status) : bool
84 {
85     return in_array($status, $this->getSuccessStatuses());
86 }
87
88
89 }
```

Рис. 8. Реалізація методів getSuccessStatuses() і statusSuccess()

Не забуваємо про вебхук і його також реалізуємо. Оскільки SDK Fondy доволі обширне і містить багато різних функцій та класів, можна дуже просто перевірити на підробку вхідні дані вебхука за допомогою класу Result. Отож, створюємо новий інстанс цього класу, де першим параметром передаємо вміст запиту та другий приватний (секретний) ключ. Далі за допомогою методу isValid ми перевіряємо коректність та

підробку вхідних даних. На завершення аналогічна схема з перевіркою статусу оплати як і у класі LiqPayService (рисунок 9).

```
63 public function webhook(array $data, callable $callback_pay_success, callable $callback_pay_fail) : bool
64 {
65     $result = new Result($data, 'your private key (secret key)');
66     if($result->isValid()) {
67         if(in_array($data['order_status'], $this->getSuccessStatuses())) {
68             call_user_func($callback_pay_success, $data['order_id'], $data['order_status'], $data, null);
69         } else {
70             call_user_func($callback_pay_fail, $data['order_id'], $data['order_status'], $data, null);
71         }
72         return true;
73     } else {
74         return false;
75     }
76 }
```

Рис. 9. Реалізація методу *webhook* для класу *FondyService*

На цьому етапі завершена розробка інтерфейсу для сервісів оплат, залишилося тільки реалізувати бізнес-логіку, яка унікальна для кожного сайту зі своїми функціональними можливостями.

Як підсумок інтерфейс *PaymentMethodInterface* може формувати веб-форму для оплати, обробляти вебхук та має методи для отримання успішних статусів.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі детально розглянуто роботу платіжних сервісів *Fondy* та *LiqPay* – найпоширеніших платіжних сервісів в Україні. Описано алгоритм підключення таких систем до веб-сайту або мобільного додатку та запропоновано його програмну реалізацію.

Продемонстрований код – це лиш обгортка, яка полегшує взаємодію сервісу оплати із тим кодом, що надає можливість підключити декілька сервісів і об'єднати їх під один інтерфейс, що в свою чергу забирає пряму залежність від конкретної реалізації SDK. Також є сенс писати подібний інтерфейс і для сайтів із одним сервісом оплати. Оскільки нерідко бувають замовники, які з певних причин вирішили замінити сервіс оплати на своєму сайті, а якщо там буде реалізовано подібний інтерфейс, то заміна сервісу оплати практично зводиться до реалізації декількох методів із інтерфейсу з новим SDK. А це в свою чергу не буде передбачати внесення змін в уже робочу логіку із замовленнями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Безготівкові розрахунки. Національний банк України. <https://bank.gov.ua/ua/payments/nocash>
- 2 Економічна правда. З картки на картку: як працюють українські платіжні сервіси. *Економічна правда*. <https://www.epravda.com.ua/projects/fintech/2019/08/21/650653/>
- 3 Національний банк України. Розподіл безготівкових операцій з використанням платіжних карток, травень 2022 року. *Національний банк України*. <https://bank.gov.ua/ua/news/all/rozpodil-bezgotivkovih-operacij-z-vikoristannyam-platijnih-kartok-traven-2022-roku>
- 4 About webhooks - GitHub Docs. *GitHub Docs*. <https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks>
- 5 Cashless transactions - BLIK - Scan and pay - moBilet - P2P payments - Pre-paid top-up - Bank Millennium. *Klienci Indywidualni - Konta, pożyczki, kredyty mieszkaniowe, karty, lokaty - Bank Millennium*. <https://www.bankmillennium.pl/en/electronic-banking/mobile-application-individuals-business/cashless-payments>



- 6 Monobank запускає послугу інтернет-еквайрінгу. *Visit Ukraine - INFORMATION PAGE FOR FOREIGN TOURISTS*. <https://visitukraine.today/uk/blog/416/monobank-zapuskaje-poslugu-internet-ekvairingu>
- 7 Monobank запускає сервіс для оплати послуг онлайн. *The Village Україна*. <https://www.the-village.com.ua/village/business/news/316447-monobank-stvorit-servis-dlya-oplati-poslug-onlayn>
- 8 RSA Encryption Algorithm - Javatpoint. *www.javatpoint.com*. <https://www.javatpoint.com/rsa-encryption-algorithm>
- 9 Six macrotrends shaping the future of payments. *PwC*. <https://www.pwc.com.au/digitalpulse/report-future-payments.html>
- 10 Wisniewski, T. P., Polasik, M., Kotkowski, R., Moro, A. (2021). Switching from Cash to Cashless Payments during the COVID-19 Pandemic and Beyond. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3794790>
- 11 <https://www.liqpay.ua/about/info>
- 12 Webhook.site - Test, process and transform emails and HTTP requests. *Webhook.site - Test, process and transform emails and HTTP requests*. <https://webhook.site/#!/89a9a026-49ac-4452-b919-e4fbc23f29c5>
- 13 What does the acronym SDK mean? AT Internet definition. *AT Internet*. <https://www.atinternet.com/en/glossary/sdk/>
- 14 What's an acquiring bank and why you need one. *Adyen | The platform to help your business grow*. <https://www.adyen.com/blog/acquiring-bank>
- 15 What to expect at the ET future of money summit 2022 on july 26-27. *The Economic Times*. <https://economictimes.indiatimes.com/industry/banking/finance/what-to-expect-at-the-et-future-of-money-summit-2022-on-july-26-27/articleshow/92900593.cms>

**Andrii P. Stupin**

the 4th year student of the faculty of information technologies and mathematics,
Lesya Ukrainka Volyn National University, Lutsk, Ukraine
stupin.andrij2018@vnu.edu.ua

Liudmyla Y. Hlynchuk

Ph.D, Associate Professor at the Department of Computer Science and Cybersecurity
Lesya Ukrainka Volyn National University, Lutsk, Ukraine
ORCID ID:0000-0002-8943-9604
hlynchuk.ludmila@vnu.edu.ua

Tetiana O. Hryshanovych

PhD, Associate Professor at the Department of Computer Science and Cybersecurity
Lesya Ukrainka Volyn National University, Lutsk, Ukraine
ORCID ID:0000-0002-3595-6964
hryshanovych.tatiana@vnu.edu.ua

ALGORITHM FOR CONNECTING ONLINE PAYMENT SERVICES FONDY, LIQPAY AND THEIR IMPLEMENTATION

АНОТАЦІЯ. The work describes the algorithms for connecting the two most common payment services in Ukraine - Fondy and LiqPay - to software tools (websites, mobile applications with Internet connection). Currently, such a topic is quite relevant, since not only the domestic economy, but also the world economy is actively transitioning to cashless payments. And this, in turn, poses challenges not only to economists, but also to information technology specialists. Now it is difficult to imagine a catalog site or a service site without online payment for purchased goods. Using non-cash payments, it is possible to pay almost everything: from goods to utilities and administrative fines. The purpose of our research is the development of an algorithm for connecting online payment services to websites and mobile applications connected to the Internet, and the software implementation of such an algorithm. Each service for making online payments has documentation and a set of development tools, utilities and documentation that allows you to create applications based on a certain technology or for a certain platform (SDK). SDKs typically have test credentials and test keys to enable test payments. Analysis of the scheme by which funds are debited from the client shows that two main methods must be implemented for the site server. These methods are: creation of a web form to proceed to the next stage of filling in payment details and receiving webhooks (a mechanism for sending requests when events occur in the system) from the service server. In our work, we describe a method for generating a web payment form and a method for processing webhooks. The proposed software solution is only a wrapper that facilitates the interaction of the payment service with the code by providing the opportunity to connect several services and combine them under one interface. This in turn removes the direct dependency on a specific SDK implementation. It also makes sense to write a similar interface for sites with one payment service, as there are often customers who, for certain reasons, decide to replace the payment service on their site. If a similar interface is implemented there, then replacing the payment service practically boils down to the implementation of several methods from the interface with the new SDK. And this will not bring changes to the already working logic with orders.

Keywords: online payments services, Fondy, LiqPay, webhook, cashless payments, online payments.

REFERENCES (TRANSLATED AND TRANSLITERATED)

- 1 Bezhotivkovi rozrakhunky. Natsionalnyi bank Ukrainy. <https://bank.gov.ua/ua/payments/nocash>
- 2 Ekonomichna pravda. Z kartky na kartku: yak pratsiuut ukraïnski platizhni servisy. Ekonomichna pravda. <https://www.epravda.com.ua/projects/fintech/2019/08/21/650653/>
- 3 Natsionalnyi bank Ukrainy. Rozpodil bezgotivkovykh operatsii z vykorystanniam platizhnykh kartok, traven 2022 roku. Natsionalnyi bank Ukrainy. <https://bank.gov.ua/ua/news/all/rozpodil-bezgotivkovih-operatsiy-z-vikoristanniam-platizhnykh-kartok-traven-2022-roku>



- 4 About webhooks - GitHub Docs. *GitHub Docs*. <https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks>
- 5 Cashless transactions - BLIK - Scan and pay - moBilet - P2P payments - Pre-paid top-up - Bank Millennium. *Klienci Indywidualni - Konta, pożyczki, kredyty mieszkaniowe, karty, lokaty - Bank Millennium*. <https://www.bankmillennium.pl/en/electronic-banking/mobile-application-individuals-business/cashless-payments>
- 6 Monobank zapuskaie posluhu internet-ekvaŭrinhu. Visit Ukraine - INFORMATION PAGE FOR FOREIGN TOURISTS. <https://visitukraine.today/uk/blog/416/monobank-zapuskaje-poslugu-internet-ekvairingu>
- 7 Monobank zapuskaie servis dlia oplaty posluh onlain. The Village Ukraina. <https://www.the-village.com.ua/village/business/news/316447-monobank-stvorit-servis-dlya-oplati-poslug-onlayn>
- 8 RSA Encryption Algorithm - Javatpoint. *www.javatpoint.com*. <https://www.javatpoint.com/rsa-encryption-algorithm>
- 9 Six macrotrends shaping the future of payments. *PwC*. <https://www.pwc.com.au/digitalpulse/report-future-payments.html>
- 10 Wisniewski, T. P., Polasik, M., Kotkowski, R., Moro, A. (2021). Switching from Cash to Cashless Payments during the COVID-19 Pandemic and Beyond. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3794790>
- 11 <https://www.liqpay.ua/about/info>
- 12 Webhook.site - Test, process and transform emails and HTTP requests. *Webhook.site - Test, process and transform emails and HTTP requests*. <https://webhook.site/#!/89a9a026-49ac-4452-b919-e4fbc23f29c5>
- 13 What does the acronym SDK mean? AT Internet definition. *AT Internet*. <https://www.atinternet.com/en/glossary/sdk/>
- 14 What's an acquiring bank and why you need one. *Adyen | The platform to help your business grow*. <https://www.adyen.com/blog/acquiring-bank>
- 15 What to expect at the ET future of money summit 2022 on july 26-27. *The Economic Times*. <https://economictimes.indiatimes.com/industry/banking/finance/what-to-expect-at-the-et-future-of-money-summit-2022-on-july-26-27/articleshow/92900593.cms>

