

# Połączenie Partnera z serwisem JustPay poprzez - METODĘ 1 (paczki kodów)



Generowanie kodów: **po stronie JustPay**

Weryfikacja kodów: **po stronie Partnera**

## Spis treści

1. Kolejne kroki w stworzeniu własnego serwisu
2. Jak skonfigurować i wdrożyć przykładowe lub własne skrypty
3. Sposób generacji kodów
4. Definicja komend
5. Opis przykładowych skryptów
6. Przykład zabezpieczonej strony
7. Ochrona danych nietekstowych

## 1. Kolejne kroki w stworzeniu własnego serwisu

1. Rejestracja w serwisie JustPay i oczekiwanie na maila potwierdzającego rejestrację.
2. Po zalogowaniu - zdefiniowanie komend oraz numerów SMS Premium.
  - a. zakładka 'komendy' - przycisk 'dodaj nową komendę'
  - b. wybrać prefiks z listy i sufiks komendy
  - c. zatwierdzić przyciskiem 'OK' (system automatycznie przekieruje do punktu 3)
3. Dodanie numeru do komendy (link 'dodaj numer' w zakładce 'komendy')
  - a. wybór numeru SMS Premium z listy dostępnych
  - b. wybór metody generacji; w tym przypadku: 'Metoda 1'
  - c. wybór długości i liczby kodów w paczce (można zostawić wartości domyślne)
  - d. definicja SMS'a zwrotnego: w okienku należy wpisać treść SMS'a zwrotnego; ciąg znaków '\$1' zostanie zastąpiony jednym z kodów z paczki
  - e. zatwierdzenie numeru przyciskiem 'OK'
4. Po zdefiniowaniu wszystkich potrzebnych numerów (można ich dodać więcej niż jeden), należy zamówić aktywację komendy.
5. Po zaakceptowaniu komendy przez administratora systemu, użytkownik zostanie poinformowany e-mail'em. Podczas oczekiwania na akceptację można przystąpić do realizacji dalszych kroków.
6. Pobranie paczki z kodami. W zakładce 'Kody' znajduje się lista wygenerowanych paczek kodów (dla każdej kombinacji komenda/Numer SMS jedna paczka). Bieżącą (aktywną) paczkę można pobrać klikając 'Pobierz' w kolumnie 'Bieżąca paczka'. (można załączyć także własną paczkę kodów).
7. Po wykorzystaniu 90% kodów z bieżącej paczki, Partner otrzyma komunikat mailem o tym fakcie. Po tym należy pobrać nową paczkę oczekującą w serwisie JustPay i wstawić ją podobnie jak poprzednią paczkę.
8. W momencie dwóch paczek podłączonych u Partnera, system JustPay po wyczerpaniu się 1 paczki, automatycznie zacznie pobierać kody z kolejnej paczki.

Po otrzymaniu przez system JustPay SMSa Użytkownika:

1. Justpay odsyła Użytkownikowi kod z aktywnej paczki kodów znajdującej się w serwisie JustPay (identyczna paczka znajduje się u Partnera).
2. Użytkownik wpisuje kod na stronie www Partnera
3. Partner sprawdza czy kod zgadza się z paczką kodów, która wcześniej pobrał z serwisu JustPay.
4. Jeśli kod jest prawidłowy, Użytkownik otrzymuje dostęp do płatnej części serwisu.
5. Partner odznacza kod Użytkownika w swojej bazie jako użyty.

## **2. Jak skonfigurować i wdrożyć przykładowe lub własne skrypty**

1. Przygotowanie tabeli z kodami
  - a. utworzenie odpowiedniego schematu w bazie danych (realizacja tego kroku jest zależna od tego co jest udostępnione na serwerze hostingowym, na którym ma być uruchomiona strona: często udostępniany jest panel www do administracji bazą danych).

Po utworzeniu schematu, należy odpowiednio zmodyfikować zmienne dostępne w klasie DBConfig w metodzie conn():

- \$user = nazwa użytkownika
- \$host = nazwa hosta na którym uruchomiona jest baza mysql
- \$pass = hasło użytkownika
- \$db = nazwa bazy

- b. utworzenie tabeli z kodami, przykładowo zapytaniem:

```
create table kody_codes (  
    code varchar(20) not null,  
    session_start timestamp,  
    valid_secs integer not null,  
    used boolean not null,  
    primary key (code)  
);
```

- c. załadowanie paczki kodów. Dla każdego z kodów z paczki, należy wykonać zapytanie SQL:

```
insert into kody_codes(code, valid_secs)  
values ('kod', 3600);
```

W miejsce 'kod' należy podstawić kod ze ściągniętego pliku, a w miejsce 3600 należy podstawić czas (w sekundach), jaki kod ma być ważny. Można tę wartość uzależnić od ceny sms'a: dla przykładu kody za 1zł mogą być ważne godzinę (jak w przykładzie), a kody za 2zł – 3 godziny.

2. Przygotowanie strony niezabezpieczonej z elementem umożliwiającym wejście do części zabezpieczonej. W odpowiednim miejscu na stronie, należy dodać następujący kod PHP:

```
<?php
include_once( 'CodeCheck.php' );
$c = new CodeCheck();
$c->render($label = "kod:", $actionsript = "index.php");
?>
```

Powyższy kod spowoduje wyświetlenie formularza do logowania. Zmienna \$label to tekst który pojawi się obok pola do wprowadzenia kodu otrzymanego przez użytkownika w smsie zwrotnym. Zmienna \$actionsript to skrypt do którego ma być przekierowany użytkownik po naciśnięciu przycisku Ok

3. Przygotowanie strony zabezpieczonej. W celu zabezpieczenia strony, należy dodać na samym początku w kodzie PHP dodać:

```
<?php
    session_start();
    include_once('CodeCheck.php');
    $c = new CodeCheck();
    if (!$c->isSessionCodeValid()) {
        include_once('login.php');
        exit;
    }
?>
```

Powyższy kod spowoduje sprawdzenie czy w sessji zapisane są już informacje o użytkowniku i kodzie który podał w trakcie logowania. Jeżeli tych informacji nie ma następuje załadowanie strony login.php

4. Skrypty przykładowe. Dla metody pierwszej przygotowano przykładowe skrypty (dokładny opis w rozdziale 5 poniżej):
  - a. CodeCheck.php – Klasa do sprawdzania poprawności wprowadzonego kodu przez użytkownika oraz generowania formularza.
  - b. DBConfig.php – Klasa do nawiązywania połączeń z bazą Mysql oraz realizacji zapytań na potrzeby klasy CodeCheck.php
  - c. index.php oraz other – zabezpieczone strony
  - d. login.php - strona która ładuje się w momencie gdy użytkownik jest nie zalogowany
  - e. logout.php – skrypt do wylogowania użytkownika

### **3. Sposób generacji kodów**

Sposób wykorzystania kodów, jak i walidacja poprawności oraz różnego rodzaju zabezpieczenia leżą po stronie Partnera. Niniejszy dokument ma na celu ułatwienie, poprzez przedstawienie kilku przykładów, uruchomienie własnego serwisu WWW, którego część

będzie chroniona kodami justpay. W dalszej części dokumentu przyjęte zostały następujące założenia:

1. Partner o nazwie „supergry”, zakończył rejestrację w systemie justpay i został zaakceptowany przez administratora,
2. „supergry” posiada serwis z grami, dostępnymi na stronach www,
3. Część z funkcjonalności gier (bardziej zaawansowane opcje) mają być płatne (udostępniane jedynie dla użytkowników, którzy zapłacili)
4. Dostęp do płatnych obszarów ma być wykupywany na określony, skończony czas.

#### **4. Definicja komend**

System justpay umożliwia zdefiniowanie jednocześnie wielu komend na wielu numerach. Umożliwia to Partnerom uruchomienie więcej niż jednego serwisu na jednym koncie w justpay. Taki rozdział jest przydatny przy późniejszym analizowaniu raportów — można rozdzielić przychody generowane przez poszczególne serwisy. Innym sposobem wykorzystania różnych komend jest rozdział na kanały publikacji: jeden serwis może być promowany poprzez reklamy w Internecie i równoległe przez ogłoszenia w prasie. Utworzenie dwóch komend w takim przypadku, umożliwi ocenę efektywności poszczególnych kanałów.

Tworzenie wielu komend, nie ma żadnego wpływu na techniczne aspekty rozwiązania. Jeśli kilka komend promuje jeden serwis, to wszystkie otrzymane kody dostępowe powinny działać jednakowo. W przypadku istnienia kilku niezależnych serwisów, kod uzyskany przez wysłanie komendy związanej z serwisem A jest po prostu nieprawidłowy przy próbie dostępu do serwisu B. Z tego powodu w dalszej części przykładów przyjęte zostanie założenie, że Partner „supergry” zdefiniował tylko jedną komendę: KOD.GRA.

Zupełnie inaczej jest w przypadku numerów SMS Premium: jeśli KOD.GRA została opublikowana na wielu numerach dla przykładu: 7136 (numer za 1zł), 7555 (numer za 5zł), oraz 7936 (numer za 9zł), to uzyskany kod powinien dawać inne uprawnienia. Te uprawnienia zależą oczywiście od specyfiki serwisu i sprzedawanych danych, jednak w przypadku gier sensowne wydaje się być uzależnienie czasu dostępu do strony od ceny sms'a. Przykładowo: 24h dla 7136, tydzień dla 7555, miesiąc dla 7936.

Po zdefiniowaniu komendy, jak to zostało opisane do tej pory, Partner „supergry” powinien mieć 3 paczki kodów (ściągniętych z zakładki kody) o nazwach zbliżonych do:

- KOD.GRA\_7136\_20080416\_1116.csv (komenda KOD.GRA na 7136),
- KOD.GRA\_7555\_20080416\_1116.csv (komenda KOD.GRA na 7555),
- KOD.GRA\_7936\_20080416\_1116.csv (komenda KOD.GRA na 7936)

#### **5. Przykładowe skrypty**

Przedstawione skrypty są jedynie przykładami — często są one celowo uproszczone, czy też napisane wręcz schematycznie. W przypadku ich ewentualnej integracji z produkcyjnym systemem, należy pamiętać o ich dokładnym sprawdzeniu pod kontem wymogów konkretnej aplikacji, oraz zgodności ze standardami bezpieczeństwa przyjętymi w systemie docelowym.

## Wybór technologii

Najbardziej popularną kombinacją technologii na ogólnodostępnych serwerach hostingowych jest PHP + MySQL, dlatego też zdecydowaliśmy się na przedstawienie ich w tej technologii. Podczas tworzenia przykładów dołożono wszelkich starań, żeby kod był jak najbardziej zrozumiały, dlatego też przeniesienie przedstawionego rozwiązania do innej technologii nie powinno nastręczyć wielu problemów.

## Konfiguracja serwera WWW

Skrypty zostały przetestowane na PHP 5.2.10 z zainstalowaną obsługą mysql

Wersja serwera MySQL: 5.0.77 — wersja serwera bazodanowego podana jest czysto informacyjnie, ponieważ zastosowane struktury danych, oraz zapytania są na tyle proste, że działałyby z dowolną wersją.

## Schemat bazy

W przykładach użyto następującej konfiguracji: użytkownik kody, hasło: kody1234, nazwa schematu: test\_kody. Do działania skryptu „zabezpieczającego” potrzebna jest tylko jedna tabela: kody\_codes. Zawiera ona następujące pola:

- code typu varchar(20) — w tym polu znajduje się kod dostępowy wysłany sms'em,
- session\_start typu timestamp — w tym polu zapisywany jest czas pierwszego użycia kodu/czas załadowania kodu do bazy danych,
- valid\_secs typu integer — pole zawiera liczbę sekund ważności kodu (od momentu jego pierwszego użycia)
- used typu boolean — pole ustawiane na true po pierwszym użyciu kodu.

Ponieważ kody dostępowe są unikatowe, pole code może być (i jest) kluczem głównym tabeli kody\_codes.

Po utworzeniu opisanej tabeli, należy ją zasilić danymi z paczek kodów. Pole valid\_secs należy ustawić na wartość odpowiednią dla numeru związanego z daną paczką, przykładowo dla 7136, będzie to: 86400 (liczba sekund w 24h).

## Klasa CodeCheck

Klasa CodeCheck zdefiniowana w pliku CodeCheck.php, służy do „ochrony” skryptów PHP przy pomocy kodów serwisu. Połączenie z bazą danych jest inicjowane w konstruktorze tej klasy — tam też należy odpowiednio skonfigurować parametry tego połączenia.

Do poprawnego działania klasy wymagane są informacje przechowywane w sesji w przypadku braku możliwości zapisania identyfikatora SID w cookie, należy zapewnić jakąś inną metodę przekazywania tego parametru.

Wykorzystanie klasy `CodeCheck` jest bardzo proste. Wystarczy na początku skryptu PHP, który ma być „chroniony”, dokleić na samym początku następujący fragment kodu:

```
<?php
    session_start();
    include_once('CodeCheck.php');
    $c = new CodeCheck();
    if (!$c->isSessionCodeValid()) {
        // akcja w przypadku niepoprawnego kodu
        exit;
    }
?>
```

W celu wyświetlenia formularza do wprowadzenia kodu przez użytkownika, można użyć następującego fragmentu kodu:

```
<?php
    include_once('CodeCheck.php');
    $c = new CodeCheck();
    $c->render("podpis:", "strona.php");
?>
```

W miejsce "podpis" należy wstawić etykietę, która ma być związana z polem tekstowym do wprowadzenia kodu, natomiast w miejsce "strona.php" — stronę, na którą użytkownik ma zostać przekierowany po wprowadzeniu poprawnego kodu

## **6. Przykład zabezpieczonej strony**

Kompletny przykład „zabezpieczonej” strony składa się z następujących plików:

- `CodeCheck.php` – Klasa do sprawdzania poprawności wprowadzonego kodu przez użytkownika oraz generowania formularza.
- `DBConfig.php` – Klasa do nawiązywania połączeń z bazą Mysql oraz realizacji zapytań na potrzeby klasy `CodeCheck.php`
- `index.php` oraz `other.php` – zabezpieczone strony
- `login.php` - strona która ładuje się w momencie gdy użytkownik jest nie zalogowany
- `logout.php` – skrypt do wylogowania użytkownika

Jeśli cała konfiguracja została przeprowadzona prawidłowo, to przykładowa strona powinna działać następująco:

1. bez podania jakiegokolwiek kodu powinna się wyświetlić strona logowania,

2. po podaniu poprawnego kodu na stronie `index.php`, bądź też na stronie `login.php`, użytkownik powinien mieć swobodną możliwość przechodzenia pomiędzy stronami zabezpieczonymi (bez konieczności ponownego podawania kodu)
3. ważność kodu upływa po upłynięciu `valid_secs` sekund od momentu pierwszego wprowadzenia kodu — po tym czasie kod po prostu jest uznawany za nieprawidłowy i nie daje możliwości wchodzenia na strony zabezpieczone

## **7. Ochrona danych nietekstowych**

Przedstawiona strona przykładowa ilustruje zabezpieczenie skryptów PHP. W przypadku konieczności ochrony danych nietekstowych (obrazków, plików muzycznych, itp) przedstawiony sposób nie będzie działał. Problem taki można rozwiązać na wiele sposobów. Najprostszym z nich jest wprowadzenie skryptu pośredniego, oraz przeniesienie danych do katalogu, do którego nie ma dostępu serwer WWW.

W Przykładzie **Ściąganie pliku po wpisaniu kodu** przedstawiony jest sposób w jaki można zabezpieczyć obrazki typu JPEG, PNG, oraz GIF.

### **i. Wariant 1 regułka modrewrite w pliku .htaccess:**

```
RewriteEngine on
RewriteCond %{REQUEST_URI} !.*mini.(png|jpg|gif)$
RewriteRule ^(.*)\.?(png|jpg|gif)$ image.php?f=$1 [L]
```

Powyższy wpis powoduje przekierowanie wszystkich requestów dotyczących plików z rozszerzeniem png, jpg lub gif (z wyjątkiem plików zawierających w nazwie frazę 'mini') na skrypt `image.php` w którym następuje weryfikacja czy użytkownik podał wcześniej kod

### **ii. Wariant 2 bez obsługi modrewrite :**

```

```

W momencie gdy na hostingu nie posiadamy obsługi modrewrite możemy zabezpieczyć pliki poprzez podanie ścieżki do pliku który chcemy objąć ochroną przy pomocy parametru `?f='ścieżka do pliku'` do skryptu `image.php`. Skrypt `image.php` należy umieścić w katalogu nadrzędnym dla katalogów w których znajdują się pliki z obrazkami.

W skrypcie `image.php` należy skonfigurować katalog (zmienna `$BASE`), w którym znajdują się pliki z obrazkami.

W razie pytań prosimy o kontakt: [justpay@avantis.pl](mailto:justpay@avantis.pl)