

## Dokumentacja techniczna API systemu SimPay.pl

Wersja 1.0 z dnia 24.03.2015 r.

### Wprowadzenie

API serwisu SimPay.pl opiera się o danych wysyłanych i zwracanych w formie JSON. W przypadku napotkania jakiegokolwiek błędu podczas obsługi żądania klienta, system zwraca informację sposób przedstawiony poniżej:

```
{
  "error": [
    {
      "error_code": "99",
      "error_name": "Empty API version; default 1",
      "error_value": null
    }
  ],
  "params": [
  ]
}
```

**error\_code** - kod zwracanego błędu

**error\_name** - krótka informacja na temat zwracanego błędu

**error\_value** - dodatkowa wartość dotycząca zwracanego błędu

### Tabela błędów zwracanych przez API SimPay

98	Empty method	Brak podanej metody, którą klient chce wykonać w ramach API
99	Empty API version; default 1	Brak podanej wersji API w adresie API - domyślna wersja API to 1 (v1)
100	Empty params	Brak parametrów do przetworzenia poprzez API
101	API version not exists	Podana wersja API przez klienta nie istnieje w systemie
102	API version status is offline	Podana przez klienta wersja API została wyłączona i nie jest dłużej wspierana

103	Client API login data was too short	Podany przez klienta parametr identyfikujący go posiada niepoprawną długość
104	Wrong client API login data (key, secret)	Podane przez klienta parametry identyfikujące są niepoprawne; Parametry <u>key</u> oraz <u>secret</u> podane podczas połączenia wymagają weryfikacji
105	Wrong client status	Konto klienta do którego podano dane identyfikujące jest obecnie zablokowane
106	Wrong API version method	Metoda podana przez klienta nie jest dostępna w danej wersji API lub nie istnieje w ogóle
107	No method require params	Dla metody podanej przez klienta podczas wywołania API potrzebne są parametry, które nie zostały podane przy wywołaniu tej metody
201	System error	Wystąpił nieoczekiwany błąd API - próba wywołania API powinna zostać powtórzona a przy powtarzaniu się zwracania kodu błędu prosimy o kontakt z działem Technicznym.
404	SMS code not found	Kod podany przez klienta do zweryfikowania w API nie istnieje
405	SMS code already used	Kod podany przez klienta do zweryfikowania w API został już wykorzystany; realizacja zamówienia w systemie klienta nie powinna być dokonana ponownie

## Lokalizacja dostępowa API

API systemu SimPay.pl dostępne jest pod adresem:

<https://simpay.pl/api/X>

Gdzie X to wersja API wykorzystywana przez danego użytkownika. Wersje API systemu różnią się funkcjonalnością oraz zakresem metod oraz zwracanych danych. W ramach rozwoju danej wersji API istnieje możliwość dodania nowych zwracanych danych, jednak nazwy metod pozostają w danej wersji zawsze bez zmian.

### Wersje API:

Wersja	Data	Nazwa	Odnosićnik URL
1	24/03/2015 r.	Wersja 1.0	<a href="http://simpay.pl/api/1">http://simpay.pl/api/1</a>

### Gotowe biblioteki API

System SimPay.pl zapewnia podstawowe biblioteki w różnych językach programowania, wspierające wszystkie podstawowe funkcje. Wraz z wydaniem nowych wersji, lub ich aktualizacji, publikowane są **na bieżąco** nowe.

W celu zapewnienia szybkiej integracji z systemem SimPay.pl zalecamy skorzystanie z gotowych **bibliotek**.

Język	Wersja API	Lokalizacja
PHP	v1	<a href="https://github.com/SimPay/API_PHP">https://github.com/SimPay/API_PHP</a>
Python	w trakcie	---
C++	w trakcie	---
JAVA	w trakcie	---
Ruby	w trakcie	---

### Dostęp do API - dane klienta

W celu wykonywania operacji na API SimPay niezbędne jest uwierzytelnienie klienta na podstawie parametrów **auth: [key, secret]**.

Parametry do autoryzacji API może pobrać w panelu Partnera SimPay:

Konto klienta / Edycja danych / API - <https://simpay.pl/panel/Client/API>



## Metody dostępne w API SimPay.pl

Metoda	Dostępność	Parametry			
		Nazwa	Opis	Wartość	Wymagane
status	v1	service_id	identyfikator usługi	Integer	tak
		number	numer na jaki wysłano sms	Integer	tak
		code	kod otrzymany w sms zwrotnym	String	tak
		show_used	czy kod jest ważny bezterminowo: domyślnie 0 - nie jest ; 1 - kod ważny bezterminowo	ENUM	---

### Przykład (PHP):

```
<?php
define('API_KEY',          'XXXXXXXX');
define('API_SECRET',      'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');
define('API_VERSION',    1);

require_once('SimPay.class.php');

try {
    $api = new SimPay(API_KEY, API_SECRET, API_VERSION);

    $api->getStatus(array(
        'service_id'    => '2',
        'number'        => '7355',
        'code'          => '5DB554',
    ));
}
```

```
catch(Exception $e) {  
    echo 'Error: ' . $e->getMessage();  
}
```

?>

W wyniku wywołania powyższej metody, wygenerowana zostaje informacja zwrotna w formacie JSON, zgodna z szablonem przedstawionym w wstępie do specyfikacji. Ze względu na pozytywne wywołanie jednej z metod, wiadomość zwrotna zostanie wzbogacona o nowy element **respond**:

JSON:

```
{  
    "respond": [],  
    "error": [],  
    "params": []  
}
```

W zależności od wywołanej metody, a także od statusu zwracanej wiadomości dane i zakres jest zmienny.

Dane zwracane dla wywołania metody **status**

Poprawna weryfikacja kodu:

<b>status</b>	string	Status weryfikacji kodu; Wartość: OK
<b>test</b>	enum	Czy weryfikowany kod należał do kodów testowych; 0 - nie, 1 - tak

Brak weryfikacji kodu (kod wykorzystany):

<b>code</b>	string	Wykorzystany kod
-------------	--------	------------------

<b>status</b>	string	Status kodu: USED
<b>time_used</b>	timestamp	Data i czas wykorzystania kodu

Biblioteka PHP posiada gotową funkcję do sprawdzenia wiadomości zwrotnej z API SimPay. Kompletny kod do weryfikacji kodu wygląda następująco:

```
<?php
define('API_KEY',          'XXXXXXXX');
define('API_SECRET',      'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');
define('API_VERSION',    1);

require_once('SimPay.class.php');

try {
    $api = new SimPay(API_KEY, API_SECRET, API_VERSION);

    $api->getStatus(array(
        'service_id'    => '2',
        'number'        => '7355',
        'code'          => '5DB554',
    ));

    if($api->check()) {
        // kod poprawny
    } else if($api->error()) {
        // niepoprawny kod
        // podczas rozwoju aplikacji system może wyświetlić błąd
        print_r($api->showError());
    }
}

catch(Exception $e) {
    echo 'Error: ' . $e->getMessage();
}

?>
```