

SPIKON

DOKUMENTACJA API

Wersja 1.3.1

Spis treści

1.	Historia zmian.....	4
2.	Wstęp	5
3.	Bezpieczeństwo	5
3.1.	Szyfrowanie	5
3.2.	Klucz API	5
3.3.	Filtrowanie adresów IP	6
4.	Metody API.....	6
4.1.	Przykłady użycia i testowanie	6
4.1.1.	JavaScript + jQuery	6
4.1.2.	C# (.NET Core 3.1).....	7
4.1.3.	Ruby 2.3	7
4.1.4.	Python 3.....	8
4.1.5.	Testowanie	8
4.2.	Konto	9
4.2.1.	Stan środków	9
4.2.2.	Lista kont telefonicznych	10
4.2.3.	Inicjowanie połączenia	11
4.3.	Książki telefoniczne	12
4.3.1.	Lista typów książek telefonicznych.....	12
4.3.2.	Podgląd książki telefonicznej.....	13
4.3.3.	Dodawanie książki telefonicznej	14
4.3.4.	Kasowanie książki telefonicznej	15
4.3.5.	Dodawanie wpisu do książki telefonicznej	16
4.3.6.	Kasowanie wpisu z książki telefonicznej.....	17
4.4.	Historia połączeń (nagrania rozmów)	18
4.4.1.	Żądanie wygenerowania historii połączeń	18
4.4.2.	Lista żądań generacji historii połączeń	19
4.4.3.	Pobranie wygenerowanej historii połączeń	20
4.5.	Call Center	22
4.5.1.	Lista kampanii Call Center	22
4.5.2.	Lista konsultantów Call Center.....	23

4.5.3. Dodawanie konsultanta do kampanii Call Center (poł. przychodzące).....	25
4.5.4. Kasowanie konsultanta z kampanii Call Center (poł. przychodzące)	25
4.5.5. Dodawanie konsultanta do kampanii Call Center (poł. wychodzące).....	26
4.5.6. Kasowanie konsultanta z kampanii Call Center (poł. wychodzące).....	26
4.5.7. Ustawianie domyślnej kampanii Call Center (poł. wychodzące).....	27
4.5.8. Podgląd monitoringu kampanii Call Center.....	28
4.6. Wiadomości SMS.....	30
4.6.1. Wysłanie wiadomości	30
4.6.2. Sprawdzanie statusu wysyłki.....	31
4.7. Syntezator Mowy	32
4.7.1. Pobieranie listy języków	32
4.7.2. Pobieranie listy głosów.....	33
4.7.3. Generowanie zapowiedzi	34
4.7.4. Pobieranie listy zapowiedzi	35
4.7.5. Pobieranie pliku z zapowiedzią	36
4.7.6. Kasowanie zapowiedzi.....	36
4.7.7. Sprawdzanie ceny zapowiedzi.....	37
4.7.8. Sprawdzanie statusu generowania zapowiedzi.....	38
5. Lista błędów.....	39

1. Historia zmian

Data	Wersja dokumentu	Opis zmian
11-07-2014	1.0.0	Pierwsza wersja dokumentacji API platformy Spikon.
10-08-2014	1.0.1	Dokumentacja uzupełniona o pełną listę błędów zwracanych przez system API.
20-08-2014	1.0.2	Dokumentacja uzupełniona o opis programu „Spikon – Tester API”.
09-02-2016	1.0.3	Dokumentacja uzupełniona o opis nowych metod związanych z Call Center. Aktualizacja kontaktowego adresu e-mail.
28-09-2017	1.0.4	Dokumentacja uzupełniona o opis nowych metod związanych z Call Center.
28-01-2020	1.1	Dokumentacja uzupełniona o opis nowych metod związanych z wiadomościami SMS.
10-04-2020	1.2	Dokumentacja uzupełniona o opis nowych metod związanych z Syntezatorem Mowy.
24-04-2020	1.2.1	Dokumentacja uzupełniona nowe metody Syntezatora Mowy
05-05-2020	1.2.2	Uaktualniono przykłady użycia API w różnych językach programowania.
27-02-2023	1.3	Dokumentacja uzupełniona o metody związane z wysłaniem dwustronnych wiadomości SMS.
17-05-2023	1.3.1	Usunięto i poprawiono błędy merytoryczne w opisach.

2. Wstęp

API - „Application Programming Interface” czyli „Interfejs Programowania Aplikacji” to „okno” umożliwiające zewnętrznym systemom czy aplikacjom zdalne łączenie się z platformą Spikon i wykonywanie czynności takich jak:

- Pobieranie danych: listy połączeń, nagrania rozmów czy stan środków na koncie;
- Zarządzanie „treściami zmiennymi”, czyli np. książkami telefonicznymi (dodawanie, kasowanie);
- Wykonywanie akcji: zestawienie połączeń między kontem telefonicznym i numerem zewnętrznym;

API w platformie Spikon działa w oparciu o metody POST i GET protokołu HTTPS. Każde wywołanie metody API musi składać się z odpowiedniego adresu URL oraz parametrów lub nagłówków. Każda metoda zwraca dane domyślnie w formacie JSON oraz opcjonalnie w formacie XML. Format XML może zostać wymuszony poprzez wysłanie odpowiedniego parametru.

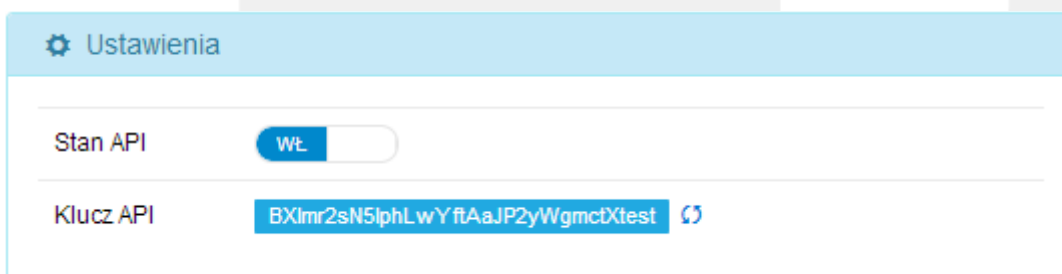
3. Bezpieczeństwo


3.1. Szyfrowanie

Użycie protokołu HTTPS oraz wymuszenie użycia nagłówków HTTP zamiast parametrów w wywołaniu GET powoduje zwiększenie bezpieczeństwa komunikacji. Parametry w metodzie POST oraz nagłówki HTTP w metodzie GET są szyfrowane, przez co praktycznie niemożliwe do „podstuchania” przez osoby trzecie.

3.2. Klucz API

Obowiązkowym parametrem (lub nagłówkiem) przekazywanym z każdą metodą jest klucz API. Każdy użytkownik platformy Spikon ma możliwość wygenerowania swojego klucza z poziomu panelu klienta. By to wykonać należy w menu po lewej stronie rozwinąć „Ustawienia i dodatki”, a następnie kliknąć w pozycję „API - ustawienia”. Domyślnie API jest wyłączone, po włączeniu nastąpi automatyczne generowanie nowego, unikalnego klucza.



Ustawienia	
Stan API	<input checked="" type="checkbox"/>
Klucz API	BXlmr2sN5phLwYftAaJP2yWgmcIXtest 

3.3. Filtrowanie adresów IP

Dodatkową metodą zabezpieczającą jest możliwość zdefiniowania adresu/adresów IP, z których platforma Spikon będzie przyjmować połączenia do systemu API. Każde zapytanie z adresu IP spoza listy otrzyma odpowiedź z błędem „IP address not whitelisted”.

Biała lista adresów IP + Dodaj

Biała lista umożliwia określenie z jakich adresów IP będzie można wykonywać żądania API. W przypadku gdy lista jest pusta każdy, znający klucz API ma możliwość wykonywania żądań API.

Adres IP	Akcje
93.184.216.119	

4. Metody API

Każda metoda API wymaga przesłania klucza API razem z pozostałymi parametrami.

4.1. Przykłady użycia i testowanie

4.1.1. JavaScript + jQuery

```
$(function() {
    $.post('https://panel.spikon.pl/api/balance/show', {
        api_key: 'BX1mr2sN5lphLwYftAaJPyWgmctXtest',
        api_format: 0
    }, function(data) {
        console.log(data);
    });
});
```

4.1.2. C# (.NET Core 3.1)

```
using System;
using System.Collections.Specialized;
using System.Net;
using System.Text;

/ ... /

string url = "https://panel.spikon.pl/api/balance/show";

var parameters = new NameValueCollection();
parameters["api_key"] = "BX1mr2sN51phLwYftAaJPyWgmctXtest";
parameters["api_format"] = "1";

var webResponse = webClient.UploadValues(url, "POST", parameters);
string webData = Encoding.UTF8.GetString(webResponse);

Console.WriteLine(webData);
```

4.1.3. Ruby 2.3

```
require 'net/https'
require 'uri'

uri = URI.parse('https://panel.spikon.pl/api/balance/show')

response = Net::HTTP.post_form(uri, {
  'api_key' => 'BX1mr2sN51phLwYftAaJPyWgmctXtest',
  'api_format' => 0
})

puts response.body
```

4.1.4. Python 3

```
from urllib.parse import urlencode
from urllib.request import Request, urlopen

url = 'https://panel.spikon.pl/api/balance/show'
data = {
    'api_key': 'BX1mr2sN5lphLwYftAaJPyWgmctXtest',
    'api_format': 0
}

request = Request(url, urlencode(data).encode())
json = urlopen(request).read().decode()
print(json)
```

4.1.5. Testowanie

Spikon udostępnia narzędzie do testowanie metod API. Jest ono dostępne pod adresem <https://panel.spikon.pl/addon/api>

4.2. Konto

4.2.1. Stan środków

URL	https://panel.spikon.pl/api/balance/show
Opis	Pobranie stanu środków.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>sign – znak dziesiętny odpowiedzi; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – przecinek (domyślnie) • 1 – kropka
Przykład odpowiedzi JSON	<pre>{"account": { "balance": "123,45" }}</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <account> <balance>123,45</balance> </account></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • balance – stan środków na koncie; może być ujemny
Uwagi	brak

4.2.2. Lista kont telefonicznych

URL	https://panel.spikon.pl/api/phone_account/list
Opis	Pobranie listy kont telefonicznych.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML
Przykład odpowiedzi JSON	<pre>{ "phone-accounts": { "count": "1", "phone-account-1": { "id": "1", "number": "123456789", "internal": "-", "login": "12345678", "description": "Brak", "state": "1", "recording": "0" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <phone-accounts> <count>1</count> <phone-account-1> <id>1</id> <number>123456789</number> <internal>-</internal> <login>12345678</login> <description>Brak</description> <state>1</state> <recording>0</recording> </phone-account-1> </phone-accounts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba kont telefonicznych • id – identyfikator konta telefonicznego • numer – publiczny numer telefonu • internal – przypisany numer wewnętrzny; minus oznacza brak • login – login SIP konta telefonicznego • description – opis konta telefonicznego • state – stan rejestracji • recording – stan nagrywania; 0 – wyłączone, 1 - włączone
Uwagi	brak

4.2.3. Inicjowanie połączenia

URL	https://panel.spikon.pl/api/call/initiate
Opis	Inicjowanie połączenia telefonicznego pomiędzy kontem telefonicznym i zewnętrznym numerem telefonu. Po poprawnym wywołaniu metody w pierwszej kolejności zostanie wydzwonione konto telefoniczne, a następnie wybrany numer telefonu.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>phone_account_id – identyfikator konta telefonicznego; pobrany metodą „/api/phone_account/list”</p> <p>destination_number – docelowy numer telefonu; maksymalna długość: 25 znaków</p>
Przykład odpowiedzi JSON	<pre>{"call": { "state": "initiated" }}</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <call> <state>initiated</state> </call></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • state – stan inicjacji połączenia
Uwagi	brak

4.3. Książki telefoniczne

4.3.1. Lista typów książek telefonicznych

URL	https://panel.spikon.pl/api/phone_book/list
Opis	Pobranie listy typów książek telefonicznych.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML
Przykład odpowiedzi JSON	<pre>{ "phone-books": { "count": "2", "phone-book-1": { "id": "0", "name": "Książka główna" }, "phone-book-2": { "id": "1", "name": "Testowa" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <phone-books> <count>2</count> <phone-book-1> <id>0</id> <name>Książka główna</name> </phone-book-1> <phone-book-2> <id>1</id> <name>Testowa</name> </phone-book-2> </phone-books></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba książek telefonicznych • id – identyfikator książki telefonicznej • name – nazwa książki telefonicznej
Uwagi	brak

4.3.2. Podgląd książki telefonicznej

URL	https://panel.spikon.pl/ api/phone_book/show
Opis	Podgląd książki telefonicznej o danym identyfikatorze.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>phone_book_id – identyfikator książki telefonicznej; pobrany metodą „/api/phone_book/list”</p>
Przykład odpowiedzi JSON	<pre>{ "phone-book-elements": { "count": "1", "phone-book-element-1": { "name": "Jan Kowalski", "number": "123456789" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <phone-book-elements> <count>1</count> <phone-book-element-1> <name>Jan Kowalski</name> <number>123456789</number> </phone-book-element-1> </phone-book-elements></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba elementów w książce telefonicznej • name – nazwa wyświetlana • number – numer telefonu
Uwagi	brak

4.3.3. Dodawanie książki telefonicznej

URL	https://panel.spikon.pl/ api/phone_book/add
Opis	Dodawanie książki telefonicznej o danej nazwie.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>phone_book_name – nazwa dodawanej książki telefonicznej; maksymalna długość: 25 znaków</p>
Przykład odpowiedzi JSON	<pre>{ "phone-books": { "phone_book": { "id": "1", "name": "Testowa" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <phone-books> <phone-book> <id>1</id> <name>Testowa</name> </phone-book> </phone-books></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • id – identyfikator dodanej książki telefonicznej • name – nazwa dodanej książki telefonicznej
Uwagi	<ul style="list-style-type: none"> • Nazwa nowej książki musi być unikalna. Próba dodania książki o nieunikalnej nazwie skończy się wygenerowaniem błędu „Already exist”.

4.3.4. Kasowanie książki telefonicznej

URL	https://panel.spikon.pl/api/phone_book/delete
Opis	Kasowanie książki telefonicznej.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>phone_book_id – identyfikator książki telefonicznej; pobrany metodą „/api/phone_book/list”</p>
Przykład odpowiedzi JSON	<pre>{ "phone-books": { "phone_book": { "id": "1", "state": "deleted" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <phone-books> <phone-book> <id>1</id> <state>deleted</state> </phone-book> </phone-books></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • id – identyfikator skasowanej książki telefonicznej • state – stan operacji kasowania książki telefonicznej
Uwagi	<ul style="list-style-type: none"> • Próba skasowania nieistniejącej książki telefonicznej skończy się wygenerowaniem błędnie „No elements found (Could not find phonebook with id = '123')”. • Próba skasowania głównej książki telefonicznej (id = 0) skończy się wygenerowaniem błędnie „Forbidden (Cannot delete phonebook with id = 0)”.

4.3.5. Dodawanie wpisu do książki telefonicznej

URL	https://panel.spikon.pl/api/phone_book_element/add
Opis	Dodawanie wpisu do książki telefonicznej.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>phone_book_id – identyfikator książki telefonicznej, do której dodać element; pobrany metodą „/api/phone_book/list”</p> <p>phone_book_element_name – nazwa wyświetlana; maksymalna długość: 120 znaków</p> <p>phone_book_element_number – numer telefonu; maksymalna długość: 25 znaków</p>
Przykład odpowiedzi JSON	<pre>{ "phone-book-elements": { "phone_book_element": { "id": "1", "name": "Jan Kowalski", "number": "123456789" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <phone-book-elements> <phone-book-element> <id>1</id> <name>Jan Kowalski</name> <number>123456789</number> </phone-book-element> </phone-book-elements></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • id – identyfikator dodanego wpisu • name – nazwa dodanego wpisu • numer – numer dodanego wpisu
Uwagi	brak

4.3.6. Kasowanie wpisu z książki telefonicznej

URL	https://panel.spikon.pl/api/phone_book_element/delete
Opis	Kasowanie wpisu w książce telefonicznej.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>phone_book_element_id – identyfikator wpisu w książce telefonicznej; pobrany metodą „/api/phone_book/show”</p>
Przykład odpowiedzi JSON	<pre>{ "phone-book-elements": { "phone_book_element": { "id": "1", "state": "deleted" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <phone-book-elements> <phone-book-element> <id>1</id> <state>deleted</state> </phone-book-element> </phone-book-elements></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • id – identyfikator skasowanego elementu • state – stan operacji kasowania wpisu w książce telefonicznej
Uwagi	<ul style="list-style-type: none"> • Próba kasowania nieistniejącego wpisu skończy się wygenerowaniem błędu „No elements found (Could not find phonebook element with id = '123')”.

4.4. Historia połączeń (nagrania rozmów)

4.4.1. Żądanie wygenerowania historii połączeń

URL	https://panel.spikon.pl/ api/call_list/order
Opis	Żądanie wygenerowania historii połączeń dla danego zakresu dat. Metoda nie zwraca żądanych danych natychmiast, żądanie jest kolejkowane i wykonywane w dogodnym dla systemu czasie. Metoda zwraca identyfikator żądania. Historię połączeń po wygenerowaniu można pobrać metodą „/api/call_list/show”.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>dir – kierunek połączenia, dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – przychodzące • 1 – wychodzące • 2 – oba <p>date_start – data początkowa w formacie YYYY-MM-DD</p> <p>date_end – data końcowa w formacie YYYY-MM-DD</p> <p>with_records_only – czy wyszukiwać tylko wpisy z nagraniami głosowymi, dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – nie • 1 – tak (domyślnie) <p>number – wyszukiwanie wpisów dla danego numeru telefonu (źródłowego lub docelowego); parametr opcjonalny; maksymalna długość: 25 znaków</p>
Przykład odpowiedzi JSON	<pre>{"call_list": { "call_list_id": "1" }}</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <call-list> <call-list-id>1</call-list-id> </call-list></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • call-list-id – identyfikator zamówionej historii połączeń
Uwagi	<ul style="list-style-type: none"> • Maksymalny zakres dat to 100 dni • Raporty są dostępne do pobrania przez maksymalnie miesiąc po wygenerowaniu

4.4.2. Lista żądań generacji historii połączeń

URL	https://panel.spikon.pl/api/call_list/list
Opis	Pobranie listy zamówionych historii połączeń.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML
Przykład odpowiedzi JSON	<pre>{ "call-lists": { "count": "2", "call-list-1": { "id": "1", "generated": "0" }, "call-list-2": { "id": "2", "generated": "1" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <call-lists> <count>2</count> <call-list-1> <id>2</id> <generated>0</generated> </call-list-1> <call-list-2> <id>2</id> <generated>1</generated> </call-list-2> </call-lists></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba zamówionych historii połączeń • id – identyfikator zamówionej historii połączeń • generated – status zamówienia, możliwe wartości: <ul style="list-style-type: none"> ○ 0 – czeka na generację ○ 1 – gotowy do pobrania
Uwagi	<ul style="list-style-type: none"> • Raporty są dostępne do pobrania przez maksymalnie miesiąc po wygenerowaniu

4.4.3. Pobranie wygenerowanej historii połączeń

URL	https://panel.spikon.pl/ api/call_list/show
Opis	Pobranie wygenerowanej historii połączeń.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>call_list_id – identyfikator zamówionej historii połączeń; pobrany metodą „/api/call_list/list”</p>
Przykład odpowiedzi JSON	<pre>{ "call-lists": { "count": "2", "call-list-1": { "date": "2014-08-20 10:46:54", "src": "123456789", "dst": "12345678", "duration": "123", "state": "Odebrane", "file": "https://panel.spikon.pl/channels/lists/call/get_playback_file?id=123456" }, "call-list-2": { "date": "2014-08-20 09:39:32", "src": "87654321", "dst": "987654321", "duration": "321", "state": "Odebrane", "file": "" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <call-lists> <count>2</count> <call-list-1> <date>2014-08-20 10:46:54</date> <src>123456789</src> <dst>12345678</dst> <duration>123</duration> <state>Odebrane</state> <file>https://panel.spikon.pl/channels/lists/call/get_playback_file?id=123456</file> </call-list-1> <call-list-2> <date>2014-08-20 09:39:32</date> <src>87654321</src> <dst>987654321</dst></pre>

	<pre><duration>321</duration> <state>Odebrane</state> <file></file> </call-list-2> </call-lists></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none">• date – data rozpoczęcia połączenia• src – numer źródłowy• dst – numer docelowy• duration – czas trwania rozmowy• state – status połączenia• file – link do pobrania nagrania rozmowy telefonicznej; jeśli w czasie rozpoczęcia nagrywania na koncie był aktywny pakiet „Nagrywanie – Wysyłka na konto FTP” wówczas w tym polu będzie zawarta tylko nazwa pliku; puste pole świadczy o wyłączonym nagrywaniu w momencie rozpoczęcia rozmowy telefonicznej
Uwagi	<ul style="list-style-type: none">• Raporty są dostępne do pobrania przez maksymalnie miesiąc po wygenerowaniu

4.5. Call Center

4.5.1. Lista kampanii Call Center

URL	https://panel.spikon.pl/api/cc_campaigns/list
Opis	Pobranie listy kampanii Call Center.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML
Przykład odpowiedzi JSON	<pre>{ "cc-campaigns": { "count": "2", "cc-campaign-1": { "id": "1", "name": "Windykacja" }, "cc-campaign-2": { "id": "2", "name": "Infolinia" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <cc-campaigns> <count>2</count> <cc-campaign-1> <id>1</id> <name>Windykacja</name> </cc-campaign-1> <cc-campaign-2> <id>2</id> <name>Infolinia</name> </cc-campaign-2> </cc-campaigns></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba kampanii • id – identyfikator kampanii • name – nazwa kampanii
Uwagi	brak

4.5.2. Lista konsultantów Call Center

URL	https://panel.spikon.pl/api/cc_agent/list
Opis	Pobranie listy konsultantów Call Center.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML
Przykład odpowiedzi JSON	<pre>{ "cc-agents": { "count": "2", "cc-agent-1": { "id": "1", "name": "Jan Kowalski", "sip": "0000000", "ddi": "220000000", "in_campaigns": [1,2], "out_campaigns": [1], "default_out_campaign": 1 }, "cc-agent-2": { "id": "2", "name": "Janina Kowalska", "sip": "0000001", "ddi": "220000001", "in_campaigns": [3,4], "out_campaigns": [4], "default_out_campaign": 3 } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <cc-agents> <count>9</count> <cc-agent-1> <id>1</id> <name>Jan Kowalski</name> <sip>0000000</sip> <ddi>220000000</ddi> <in-campaigns type="array"> <in-campaign type="integer">1</in-campaign> <in-campaign type="integer">2</in-campaign> </in-campaigns> <out-campaigns type="array"> <out-campaign type="integer">1</out-campaign> </out-campaigns> <default-out-campaign type="integer">1</default-out-campaign> </cc-agent-1></pre>

	<pre> <cc-agent-2> <id>2</id> <name>Janina Kowalska</name> <sip>0000001</sip> <ddi>220000001</ddi> <in-campaigns type="array"> <in-campaign type="integer">3</in-campaign> <in-campaign type="integer">4</in-campaign> </in-campaigns> <out-campaigns type="array"> <out-campaign type="integer">4</out-campaign> </out-campaigns> <default-out-campaign type="integer">3</default-out-campaign> </cc-agent-2> </cc-agents> </pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba kampanii • id – identyfikator konsultanta • name – imię i nazwisko • sip – login SIP • ddi – publiczny numer telefonu • in_campaigns – identyfikatory kampanii Call Center (poł. przychodzące) • out_campaigns – identyfikatory kampanii Call Center (poł. wychodzące) • default_out_campaign – domyślna kampania Call Center (poł. wychodzące)
Uwagi	Brak

4.5.3. Dodawanie konsultanta do kampanii Call Center (poł. przychodzące)

URL	https://panel.spikon.pl/api/cc_in_campaign/add_agent
Opis	Dodawanie konsultanta do kampanii Call Center (poł. przychodzące).
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML agent_id – identyfikator konsultanta Call Center campaign_id – identyfikator kampanii Call Center
Przykład odpowiedzi JSON	<code>{"add_agent_status":"OK"}</code>
Przykład odpowiedzi XML	<code><?xml version="1.0" encoding="UTF-8"?></code> <code><add-agent></code> <code><status>OK</status></code> <code></add-agent></code>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • status – status wykonania akcji
Uwagi	Brak

4.5.4. Kasowanie konsultanta z kampanii Call Center (poł. przychodzące)

URL	https://panel.spikon.pl/api/cc_in_campaign/remove_agent
Opis	Kasowanie konsultanta z kampanii Call Center (poł. przychodzące).
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML agent_id – identyfikator konsultanta Call Center campaign_id – identyfikator kampanii Call Center
Przykład odpowiedzi JSON	<code>{"remove_agent":{"status":"OK"}}</code>
Przykład odpowiedzi XML	<code><?xml version="1.0" encoding="UTF-8"?></code> <code><remove-agent></code> <code><status>OK</status></code> <code></remove-agent></code>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • status – status wykonania akcji
Uwagi	Brak

4.5.5. Dodawanie konsultanta do kampanii Call Center (poł. wychodzące)

URL	https://panel.spikon.pl/api/cc_out_campaign/add_agent
Opis	Dodawanie konsultanta do kampanii Call Center (poł. wychodzące).
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML agent_id – identyfikator konsultanta Call Center campaign_id – identyfikator kampanii Call Center
Przykład odpowiedzi JSON	<code>{"add_agent_status":"OK"}</code>
Przykład odpowiedzi XML	<code><?xml version="1.0" encoding="UTF-8"?></code> <code><add-agent></code> <code><status>OK</status></code> <code></add-agent></code>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • status – status wykonania akcji
Uwagi	Brak

4.5.6. Kasowanie konsultanta z kampanii Call Center (poł. wychodzące)

URL	https://panel.spikon.pl/api/cc_out_campaign/remove_agent
Opis	Kasowanie konsultanta z kampanii Call Center (poł. wychodzące).
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML agent_id – identyfikator konsultanta Call Center campaign_id – identyfikator kampanii Call Center
Przykład odpowiedzi JSON	<code>{"remove_agent":{"status":"OK"}}</code>
Przykład odpowiedzi XML	<code><?xml version="1.0" encoding="UTF-8"?></code> <code><remove-agent></code> <code><status>OK</status></code> <code></remove-agent></code>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • status – status wykonania akcji
Uwagi	Brak

4.5.7. Ustawianie domyślnej kampanii Call Center (poł. wychodzące)

URL	https://panel.spikon.pl/api/cc_out_campaign/set_default
Opis	Ustawianie domyślnej kampanii Call Center (poł. wychodzące).
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML agent_id – identyfikator konsultanta Call Center campaign_id – identyfikator kampanii Call Center
Przykład odpowiedzi JSON	<code>{"out_campaign_set_default":{"status":"OK"}}</code>
Przykład odpowiedzi XML	<code><?xml version="1.0" encoding="UTF-8"?></code> <code><out-campaign-set-default></code> <code><status>OK</status></code> <code></out-campaign-set-default></code>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • status – status wykonania akcji
Uwagi	Brak

4.5.8. Podgląd monitoringu kampanii Call Center

URL	https://panel.spikon.pl/api/cc_monitor/show
Opis	Podgląd monitoringu kampanii Call Center.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>cc_campaign_id – identyfikator kampanii Call Center; pobrany metodą „/api/cc_campaigns/list”</p>
Przykład odpowiedzi JSON	<pre>{ "cc-monitor": { "basic-data": { "free-members": 2, "busy-members": 0, "calls-waiting": 0, "answered": 0, "unanswered": 4, "in-queue": 0, "average-waiting-time": "00:00", "max-waiting": 20, "average-call-time": "00:00", "measured-sla": "01:00", "sla": 0 }, "campaign-members": { "cc-campaign-member-1": { "login": "1234567", "description": "Janina Kowalska", "state": "Wolny", "answered": "0", "last-call": "Brak danych", "pause": "Tak", "skill": "1" }, "cc-campaign-member-2": { "login": "2345678", "description": "Adam Nowacki", "state": "Wolny", "answered": "0", "last-call": "Brak danych", "pause": "Nie", "skill": "1" } }, "waiting-calls": {} } }</pre>

Przykład odpowiedzi XML

```

<?xml version="1.0" encoding="UTF-8"?>
<cc-monitor>
  <basic-data>
    <free-members type="integer">2</free-members>
    <busy-members type="integer">0</busy-members>
    <calls-waiting type="integer">0</calls-waiting>
    <answered type="integer">0</answered>
    <unanswered type="integer">4</unanswered>
    <in-queue type="integer">0</in-queue>
    <average-waiting-time>00:00</average-waiting-time>
    <max-waiting type="integer">20</max-waiting>
    <average-call-time>00:00</average-call-time>
    <measured-sla>01:00</measured-sla>
    <sla type="integer">0</sla>
  </basic-data>
  <campaign-members>
    <cc-campaign-member-1>
      <login>1234567</login>
      <description>Janina Kowalska</description>
      <state>Wolny</state>
      <answered>0</answered>
      <last-call>Brak danych</last-call>
      <pause>Tak</pause>
      <skill>1</skill>
    </cc-campaign-member-1>
    <cc-campaign-member-2>
      <login>2345678</login>
      <description>Adam Nowacki</description>
      <state>Wolny</state>
      <answered>0</answered>
      <last-call>Brak danych</last-call>
      <pause>Nie</pause>
      <skill>1</skill>
    </cc-campaign-member-2>
  </campaign-members>
  <waiting-calls>
</waiting-calls>
</cc-monitor>

```

Opis pól odpowiedzi

- **free-members** – liczba wolnych konsultantów
- **busy-members** – liczba zajętych konsultantów
- **calls-waiting** – liczba połączeń oczekujących
- **answered** – liczba połączeń odebranych
- **unanswered** – liczba połączeń nieodebranych
- **in-queue** – liczba połączeń w kolejce
- **average-waiting-time** – średni czas oczekiwania
- **max-waiting** – maks. ilość oczekujących
- **average-call-time** – średni czas rozmów
- **measured-sla** – mierzony czas SLA

	<ul style="list-style-type: none"> • sla – SLA, wyrażone w procentach • login – login SIP • description – opis • state – status • answered – połączenia odebrane • last-call – ostatnie połączenie • pause – przerwa • skill – poziom kompetencji
Uwagi	brak

4.6. Wiadomości SMS

4.6.1. Wysyłanie wiadomości

URL	https://panel.spikon.pl/api/sms/send
Opis	Wysyłanie wiadomości SMS na zadany numer/numery.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML <p>from – nazwa nadawcy; dopuszczalne wartości: SMS, Informacja, INFO, KONKURS, NOWOSC, OFERTA, PROMOCJA, WIZYTA; 2way</p> <p>to – numer docelowy; tylko numery polskie; 9 cyfr</p> <p>message – tekst wiadomości SMS; długość: maksymalnie 918 znaków (6 standardowych wiadomości SMS)</p>
Przykład odpowiedzi JSON	<pre>{ "sms": { "state": "OK", "ids": [12] } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <sms> <state> OK </state> <ids type="array"> <id type="integer">13</id> </ids> </sms></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • state – zawsze OK • ids – tablica identyfikatorów wiadomości; identyfikator na potrzeby metody /api/sms/state
Uwagi	<p>- Prosimy o kontakt z Biurem Obsługi Klienta w celu zgłoszenia nowej, własnej nazwy nadawcy.</p> <p>- Wartość „2way” parametru „from” oznacza umożliwienie wysłania odpowiedzi na wysłaną wiadomość SMS.</p>

4.6.2. Sprawdzanie statusu wysyłki

URL	https://panel.spikon.pl/api/sms/state
Opis	Sprawdzenie statusu wiadomości SMS.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML id – identyfikator wiadomości
Przykład odpowiedzi JSON	<pre>{ "sms": { "state": 1 } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <sms> <state type="integer">1</state> </sms></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • state – status zgodnie z listą poniżej • error – w przypadku statusu 2 w tym polu może znaleźć się opis błędu lub dodatkowe informacje
Lista statusów	-1 – Brak środków 0 – Nowa 1 – Obsługiwana 2 – Błąd 3 – Niedostarczona - błędny lub niedostępny numer 4 – Wysłana 5 – Dostarczona
Uwagi	brak

4.7. Syntezator Mowy

4.7.1. Pobieranie listy języków

URL	https://panel.spikon.pl/api/tts/language_list
Opis	Pobranie listy dostępnych języków.
Metoda	POST
Parametry	<p>api_key – klucz API</p> <p>api_format – format odpowiedzi API; dopuszczalne wartości:</p> <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML
Przykład odpowiedzi JSON	<pre>{ "tts": { "count": "28", [{ "language": "en-US", "description": "Angielski (Amerykański)" }, { "language": "pl-PL", "description": "Polski" }] } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <tts> <count>28</count> <languages type="array"> <language> <language>en-US</language> <description>Angielski (Amerykański)</description> </language> <language> <language>pl-PL</language> <description>Polski</description> </language> </languages> </tts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba dostępnych języków • languages – zbiór dostępnych języków <ul style="list-style-type: none"> ○ language – kod języka, do użycia w metodzie /api/tts/voice_list ○ description – nazwa języka
Uwagi	Powyższe przykłady prezentują małą część dostępnych języków.

4.7.2. Pobieranie listy głosów

URL	https://panel.spikon.pl/api/tts/voice_list
Opis	Pobieranie listy głosów dla wybranego języka.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML language – kod języka (np. pl-PL)
Przykład odpowiedzi JSON	<pre>{ "tts": { "count": "4", "voices": [{ "name": "Ewa", "gender": "F" }, { "name": "Jacek", "gender": "M" }] } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <tts> <count>4</count> <voices type="array"> <voice> <name>Ewa</name> <gender>F</gender> </voice> <voice> <name>Jacek</name> <gender>M</gender> </voice> </voices> </tts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba głosów • voices – lista głosów <ul style="list-style-type: none"> ○ name – nazwa głosu, do użycia w metodzie /api/tts/generate ○ gender – płeć głosu: F – kobieta, M - mężczyzna
Uwagi	Powyższe przykłady prezentują część dostępnych głosów.

4.7.3. Generowanie zapowiedzi

URL	https://panel.spikon.pl/api/tts/generate
Opis	Generowanie zapowiedzi z wybranym głosem.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML voice – nazwa głosu (np. Ewa) name – opis generowanej zapowiedzi text – treść zapowiedzi (max. 3000 znaków)
Przykład odpowiedzi JSON	<pre>{ "tts": { "tts_id": "1" } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <tts> <tts-id>1</tts-id> </tts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • tts-id – identyfikator wygenerowanej zapowiedzi, do wykorzystania w metodzie /api/tts/download oraz /api/tts/delete
Uwagi	brak

4.7.4. Pobieranie listy zapowiedzi

URL	https://panel.spikon.pl/api/tts/list
Opis	Pobieranie listy wszystkich zapowiedzi konta Spikon.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML
Przykład odpowiedzi JSON	<pre>{ "tts ": { "count": "2", "tts- 1": { "id": "1", "name": "Zapowiedź 1", "generated": "1" }, "tts- 2": { "id": "2", "name": "Zapowiedź 2", "generated": "0" } } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <tts> <count>2</count> <tts-1> <id>1</id> <name>Zapowiedź 1</name> <generated>1</generated> </tts-1> <tts-2> <id>2</id> <name>Zapowiedź 2</name> <generated>0</generated> </tts-2> </tts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • count – liczba zapowiedzi • id – identyfikator zapowiedzi, do wykorzystania w metodzie /api/tts/download oraz /api/tts/delete • name – nazwa zapowiedzi • generated – 1: gotowe do pobrania, 0: oczekuje na wygenerowanie
Uwagi	brak

4.7.5. Pobieranie pliku z zapowiedzią

URL	https://panel.spikon.pl/api/tts/download
Opis	Pobieranie pliku z zapowiedzią. Wysyłany jest plik w formacie MP3 (domyślnie) lub WAVE PCM.
Metoda	GET
Nagłówki	Spikon-API-Key – klucz API
Parametry	tts_id – identyfikator zapowiedzi pcm_8k16bm – zmiana formatu wysyłanego pliku na WAVE PCM, 8000 herców, 16 bitów, mono <ul style="list-style-type: none"> • 0 – wyłączone, zostanie wysłany plik MP3 (domyślnie) • 1 – włączone
Przykład odpowiedzi JSON	Nd.
Przykład odpowiedzi XML	Nd.
Opis pól odpowiedzi	Nd.
Uwagi	Ta metoda zwraca stream z plikiem MP3 lub WAVE. <u>UWAGA.</u> <u>Wymagane wysłanie klucza API w nagłówku HTTP zamiast w parametrze.</u>

4.7.6. Kasowanie zapowiedzi

URL	https://panel.spikon.pl/api/tts/delete
Opis	Kasowanie zapowiedzi.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML tts_id – identyfikator zapowiedzi
Przykład odpowiedzi JSON	<pre>{ "tts": { "id": "1", "state": "deleted" } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <tts> <id>1</id> <state>deleted</state> </tts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • id – identyfikator zapowiedzi • state – zawsze deleted
Uwagi	brak

4.7.7. Sprawdzanie ceny zapowiedzi

URL	https://panel.spikon.pl/api/tts/price
Opis	Sprawdzanie ceny zapowiedzi.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML voice – nazwa głosu (np. Ewa) text – treść zapowiedzi (max. 3000 znaków)
Przykład odpowiedzi JSON	<pre>{ "tts": { "cost_net": "9.0", "cost_gross": "11.07" } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <tts> <cost-net>9.0</cost-net> <cost-gross>11.07</cost-gross> </tts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • cost_net – koszt zapowiedzi (PLN netto) • cost_gross – koszt zapowiedzi (PLN brutto)
Uwagi	brak

4.7.8. Sprawdzanie statusu generowania zapowiedzi

URL	https://panel.spikon.pl/api/tts/status
Opis	Sprawdzanie statusu generowania zapowiedzi.
Metoda	POST
Parametry	api_key – klucz API api_format – format odpowiedzi API; dopuszczalne wartości: <ul style="list-style-type: none"> • 0 – JSON (domyślnie) • 1 – XML tts_id – identyfikator zapowiedzi
Przykład odpowiedzi JSON	<pre>{ "tts": { "id": "70", "generated": "1" } }</pre>
Przykład odpowiedzi XML	<pre><?xml version="1.0" encoding="UTF-8"?> <tts> <id>70</id> <generated>1</generated> </tts></pre>
Opis pól odpowiedzi	<ul style="list-style-type: none"> • id – identyfikator zapowiedzi • generated – aktualny stan generowania zapowiedzi <ul style="list-style-type: none"> ○ 0 – oczekuje na generację ○ 1 – wygenerowana
Uwagi	brak

5. Lista błędów

Każda z metod opisanych powyżej może w określonych przypadkach zwrócić komunikat błędu. Dołożyliśmy wszelkich starań by każdy z komunikatów wystarczająco dobrze opisywał zaistniały problem. Poniżej znajduje się pełna lista błędów, które mogą wystąpić podczas korzystania z API platformy Spikon.

ID błędu	Nazwa błędu	Opis błędu
-1	Unknown method	Nieznana metoda API. Możliwe wystąpienie tzw. „literówki” w adresie URL metody.
-2	API key missing	Klucz API nie został przesłany z parametrami żądania.
-3	Unknown API key	Nieznany klucz API. Najczęściej występuje po wygenerowaniu nowego klucza. Klucz API można podejrzeć w panelu klienta platformy Spikon, w menu „Ustawienia i dodatki” -> „API - ustawienia”.
-4	API turned off	API jest wyłączone. API można włączyć w panelu klienta platformy Spikon, w menu „Ustawienia i dodatki” -> „API - ustawienia”.
-5	IP address not whitelisted	Adres IP żądającego nie znajduje się na białej liście. Zarządzanie białą listą adresów IP jest możliwe z poziomu panelu klienta platformy Spikon, w menu „Ustawienia i dodatki” -> „API - ustawienia”.
-6	Wrong value for parameter	Błędna wartość parametru. Dodatkowe pole „desc” komunikatu zawiera dokładny opis, który parametr ma błędną wartość.
-7	Parameter missing	Brakujący parametr. Dodatkowe pole „desc” komunikatu zawiera nazwę brakującego parametru.
-8	No elements found	Nie znaleziono elementów. Błąd informujący o braku wyników żądania.
-9	Not ready	Wynik żądania nie jest gotowy do pobrania, proszę spróbować później.
-10	Already exist	Dodawany element już istnieje i nie może zostać zduplikowany.
-11	Forbidden	Wykonywane żądanie jest niemożliwe do wykonania; brak dostępu.
-12	Data in wrong format	Błędny format danych wejściowych.
-1000	General failure, please contact service!	<p>Błąd ogólny, proszę o kontakt z serwisem. Błąd występujący ekstremalnie rzadko, może wskazywać na błąd w systemie. Po wystąpieniu tego błędu serdecznie prosimy o poinformowanie o nim serwis Spikon za pośrednictwem jednego z dostępnych kanałów kontaktu:</p> <ul style="list-style-type: none"> • Adres e-mail: serwis@spikon.pl • Numer telefonu: 588 808 888