

SPIKON

DOKUMENTACJA API

Wersja 1.0.3

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.....	6
4.1.1.	JavaScript + jQuery	6
4.1.2.	C# (.NET 2.0)	7
4.1.3.	Ruby.....	7
4.2.	Konto	8
4.2.1.	Stan środków	8
4.2.2.	Lista kont telefonicznych.....	9
4.2.3.	Inicjowanie połączenia	10
4.3.	Książki telefoniczne	11
4.3.1.	Lista typów książek telefonicznych.....	11
4.3.2.	Podgląd książki telefonicznej.....	12
4.3.3.	Dodawanie książki telefonicznej	13
4.3.4.	Kasowanie książki telefonicznej	14
4.3.5.	Dodawanie wpisu do książki telefonicznej	15
4.3.6.	Kasowanie wpisu z książki telefonicznej.....	16
4.4.	Historia połączeń (nagrania rozmów)	17
4.4.1.	Żądanie wygenerowania historii połączeń	17
4.4.2.	Lista żądań generacji historii połączeń.....	18
4.4.3.	Pobranie wygenerowanej historii połączeń	19
4.5.	Call Center	21
4.5.1.	Lista kampanii Call Center	21
4.5.2.	Podgląd monitoringu kampanii Call Center.....	22
5.	Lista błędów.....	24
6.	Rozwój	25

7. Opis programu „SPIKON – Tester API” 25

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.

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 konta
- 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 metodę POST protokołu HTTPS. Każde wywołanie metody API musi składać się z odpowiedniego adresu URL oraz parametrów. Każda metoda zwraca dane domyślnie w formacie JSON oraz opcjonalnie w formacie XML. Format XML może zostać wymuszony poprzez wysłanie odpowiedniego parametru.

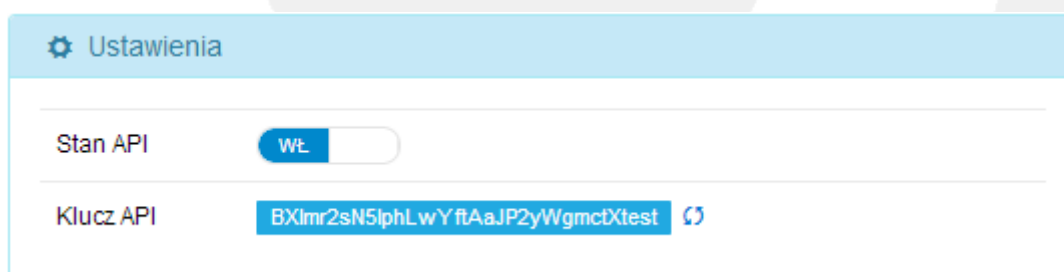
3. Bezpieczeństwo

3.1. Szyfrowanie

Użycie tylko i wyłącznie metody POST protokołu HTTPS niesie za sobą wiele zalet w stosunku do technologii REST, gdzie część zapytań musi być przesyłana metodą GET z parametrami w adresie URL. URL metody GET w HTTPS nie jest poddawany szyfrowaniu, więc osoby trzecie mogą stosunkowo łatwo zrealizować nasłuch komunikacji między klientem, a serwisem API platformy SPIKON i uzyskać tym sposobem klucz API. Metoda POST nie zawiera parametrów wywołania w adresie URL, są one przesyłane w części „Content” pakietu, dzięki czemu wszystkie poufne informacje ulegają szyfrowaniu.

3.2. Klucz API

Obowiązkowym parametrem 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.



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 umożliwi 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. Poniżej znajduje się przykład parametru zawierającego klucz:

`api_key=BXlmr2sN5lphLwYftAaJPYWgmctXtest`

4.1. Przykłady użycia

4.1.1. JavaScript + jQuery

```
$(function() {
  $.post('https://panel.spikon.pl/api/balance/show', {
    api_key: 'BXlmr2sN5lphLwYftAaJPYWgmctXtest',
    api_format: 0
  }, function(data) {
    console.log(data);
  });
});
```

4.1.2. C# (.NET 2.0)

```
using System;
using System.Text;
using System.Net;
using System.IO;

/ . . . /

WebRequest webRequest = WebRequest.Create("https://panel.spikon.pl/api/balance/show");
webRequest.Method = "POST";

string postData = "api_key=BX1mr2sN5lphLwYftAaJPyWgmctXtest&api_format=0";
byte[] byteArray = Encoding.UTF8.GetBytes(postData);

webRequest.ContentType = "application/x-www-form-urlencoded";
webRequest.ContentLength = byteArray.Length;

Stream dataStream = webRequest.GetRequestStream();
dataStream.Write(byteArray, 0, byteArray.Length);
dataStream.Close();

WebResponse webResponse = webRequest.GetResponse();
dataStream = webResponse.GetResponseStream();

StreamReader streamReader = new StreamReader(dataStream);
string responseFromServer = streamReader.ReadToEnd();

Console.WriteLine(responseFromServer);

streamReader.Close();
dataStream.Close();
webResponse.Close();
```

4.1.3. Ruby

```
require "net/https"
require "uri"

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

response = Net::HTTP.post_form(uri, {
  "api_key" => "BX1mr2sN5lphLwYftAaJPyWgmctXtest",
  "api_format" => "0"
})

puts response.body
```

4.2. Konto

4.2.1. Stan środków

URL	https://panel.spikon.pl/api/balance/show
Opis	Pobranie stanu środków.
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.
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>{ "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.
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.
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>{ "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.
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.
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.
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 błędu „No elements found (Could not find phonebook with id = '123')”. • Próba skasowania głównej książki telefonicznej (id = 0) skończy się wygenerowaniem błędu „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.
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.
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 skasowania 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	<p>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”.</p>
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.
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>{ "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.
Parametry	api_key – klucz API call_list_id – identyfikator zamówionej historii połączeń; pobrany metodą „/api/call_list/list”
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> <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.
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. Podgląd monitoringu kampanii Call Center

URL	https://panel.spikon.pl/ api/cc_monitor/show
Opis	Podgląd monitoringu kampanii Call Center.
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

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.
-1000	General failure, please contact service!	<p>Błąd ogólny, proszę o kontakt z serwisem. Błąd występujący ekstremalnie rzadko, może wskazywać na poważny 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: 58 727 07 77

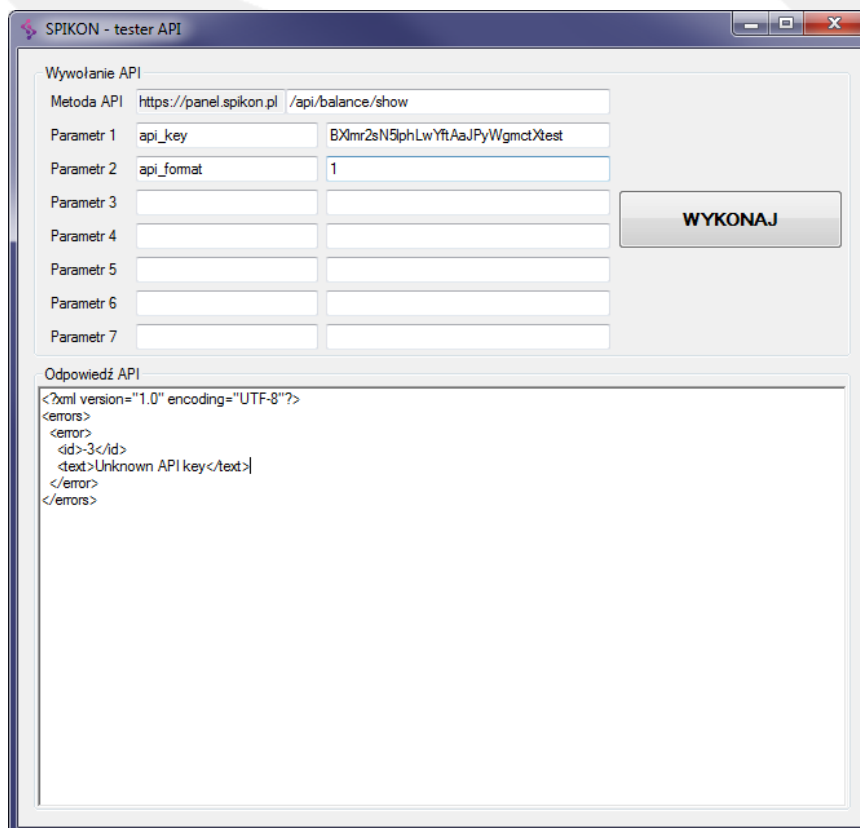
6. Rozwój

Dział rozwoju platformy SPIKON czeka na opinie i propozycje dotyczące systemu API. Zapraszamy do kontaktu:

- Adres e-mail: kontakt@spikon.pl
- Numer telefonu: 58 727 07 77

7. Opis programu „SPIKON – Tester API”

Program „SPIKON – Tester API” to intuicyjna aplikacja pozwalająca na łatwe testowanie API platformy SPIKON. Interfejs aplikacji składa się z jednego okna zawierającego formularz oraz pole odpowiedzi systemu API.



Wywołanie API

Metoda API

Parametr 1

Parametr 2

Parametr 3

Parametr 4

Parametr 5

Parametr 6

Parametr 7

WYKONAJ

Odpowiedź API

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
<error>
<id>-3</id>
<text>Unknown API key</text>
</error>
</errors>
```

Testowe żądanie API wykonać można uzupełniając pole „Metoda API” oraz jeden lub wiele parametrów. Po wciśnięciu przycisku „WYKONAJ” dana metoda zostanie przesłana do systemu API platformy SPIKON. Odpowiedź z systemu zostanie wyświetlona w polu „Odpowiedź API”.

Program jest darmowy, dostępny dla każdego zalogowanego klienta platformy SPIKON pod adresem: http://s.spikon.pl/api/SPIKON_Tester_API_v_1_0.zip

„SPIKON – Tester API” wymaga darmowego pakietu „Microsoft .NET Framework 4”, który jest dostępny pod adresem: <http://www.microsoft.com/pl-pl/download/details.aspx?id=17851>