

SOCAN API DOCUMENTATION

Getting Started with SOCAN's APIs	2
Introduction	2
Creating an Account.....	2
API Usage – Applications	2
Security	3
Encryption.....	3
OAuth.....	3
API Environments	4
Production Environment.....	4
Sandbox Environment.....	4
Operations	5
Overview.....	5
Live Performance Status.....	5
Live Performance Notification	8
Member Verification.....	16
Join SOCAN.....	17
Work Registration – Submission.....	17
Work Registration – Status	17
Work Registration – Acknowledgement.....	18
Appendix	19
Appendix 1: Country Table Codes.....	19
Appendix 2: Province/State Codes	24
Change Log.....	26

Getting Started with SOCAN's APIs

Introduction

SOCAN has a few APIs that will enable you to help connect creators and publishers you work with to SOCAN. With the APIs you will be able to:

- Notify SOCAN of live performances performed by their members
- View status of previously submitted live performances
- Register works on behalf of a SOCAN member
- Help creators or publishers join SOCAN
- Verify SOCAN members
- Notify SOCAN of live performances performed by members of other societies (coming soon)

Creating an Account

If you do not have a developer account, sign up to register for an account. You will need your username and password to properly get an access token. Your registration will need to be reviewed and approved to gain access to the developer portal.

If you have several developers working for the same company, you can invite new users to your account once your account is approved. Once approved, you will receive an email confirming the approval of your developer account, follow the instructions in the email to activate your account.

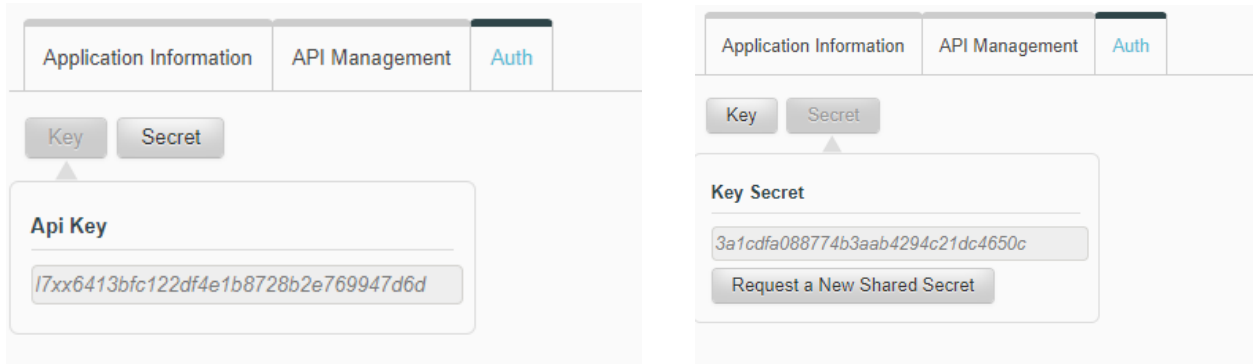
API Usage – Applications

To start using any API, an application must be created. If you already have an application created, you can add any API you have access to. For testing purposes only, the Sandbox APIs can be added. Once you are ready to use the production APIs, send an email with your application & organization name to socanapi@socan.com. Once access has been granted, you will be able to add the production APIs to your application.

Application Management – Add application

The image displays three screenshots of the SOCAN API Management interface. The top-left screenshot shows the 'Manage Applications' dashboard with a sidebar containing 'DASHBOARD', 'ORGANIZATION', and 'APPLICATIONS'. The main content area states 'There are no applications' and includes an 'Add Application' button. The top-right screenshot shows the 'Application Information' form with fields for 'Application Name *' (MyTestApp), 'Platform *' (Hybrid), and a 'Description' text area. The bottom-left screenshot shows the 'API Management' tab with 'Current APIs *' and 'Add APIs' sections, including a dropdown menu to 'Choose an API' and 'Next Step' and 'Save' buttons.

All applications created are reviewed and approved by SOCAN. Once approved by SOCAN, you will be able to view your application API key and Secret



Security

Encryption

All requests sent to SOCAN for processing must be sent over SSL or they will be rejected. All requests must use at least TLS 1.2 security protocol, or the request will be rejected/blocked.

OAuth

All requests submitted must include an OAuth Token. A token can be obtained by submitting a POST request to <https://api.socan.ca/auth/oauth/v2/token>. The request will need to contain the following parameters to obtain a token:

Query Parameters

Name	Required	Description
client_id: string	Required	This is the API key and can be found in the developer portal within your application
client_secret: string	Required	API Secret that that can found in the developer portal within your application
grant_type: string	Required	This value should be set as 'password'
username: string	Required	Submitter SOCAN username (your username used in the developer portal)
password: string	Required	Submitter SOCAN password (your password used in the developer portal)

If authorization is granted and request is successful a response will be sent with the following:

- access_token
- token_type
- expires_in
- refresh_token
- scope

Error Response (HTML Status code 4xx) – Object

Property Name	Description
error: string	The name of the error
error_description: string	The description of the error

Valid Errors

Error Name	Description
Invalid Client	The given client credentials are not valid <ul style="list-style-type: none">• This error occurs if the API key or Secret is not valid
Invalid Request	The resource owner could not be authenticated due to missing or invalid credentials <ul style="list-style-type: none">• This error would occur if the username or password is incorrect
Unsupported grant type	The given grant_type is not supported <ul style="list-style-type: none">• This error occurs if the grant_type is incorrect

API Environments

Production Environment

All production API services will be located at <https://api.socan.ca/api>. You will only be able to submit requests to production APIs your application has access to.

Sandbox Environment

We have a sandbox environment available which will allow you to test our APIs. The sandbox service is located at <https://api.socan.ca/sandbox>

Operations

Overview

Path	Operation	Description
/liveperformance	GET	Get the status of a previously submitted live performance
/liveperformance	POST	Submit a new live performance notification
/MemberVerification	POST	Find out if someone is a SOCAN member or not
/JoinSOCAN	POST	Initiate the application for membership for non-SOCAN members
/SubmitWork	POST	Submit a work on behalf of a SOCAN member
/getStatus	POST	Get the status of a submitted work registration
/getACK	POST	Get work registration acknowledgement file

Live Performance Status

Description

The live performance status operation will allow you to get the status of your previously submitted live performance. Data will be returned in JSON.

Request

Host

Sandbox: <https://api.socan.ca/sandbox>

Production: <https://api.socan.ca/api>

Path

Get /v3/liveperformance

Query Parameters

Name	Required	Description	Example
memberNo: string	Required	The member number used to submit the original live performance <ul style="list-style-type: none">Must be 7 or 8 digitsMust be numeric	9999999
confirmationNo: string	Required	The confirmation number received from SOCAN when the original performance was submitted <ul style="list-style-type: none">Must be numeric	156
apiKey: string Your SOCAN API Key	Required		Your API Key

Request Headers

- Authorization: must include Bearer Token
- Content-type: application/json

Example Request

URL:

https://api.socan.ca/sandbox/v3/liveperformance?memberNo=\${memberNumber}&confirmationNo=\${confirmationNumber}&apiKey=\${apiKey}`

```
const getPerformanceStatus = (accessToken) => {
  const options = {
    url: statusUrl,
    method: "GET",
    headers: {
      'Authorization': `Bearer ${accessToken}`,
      'Content-type': 'application/json'
    }
  };
  request(options, (err, res, body) => {
    // ACCEPT RESPONSE
  })
};
```

Successful Response Body (HTML Status code - 200) – Object

Property Name	Description	Example
memberNo: string	The member number used to submit the original live performance	9999999
confirmationNo: string	The confirmation number received from SOCAN when the original performance was submitted	156
performanceDate: string	Date of the performance (YYYY-DD-MM)	2017-21-08
performanceTime: string	Time of the performance <ul style="list-style-type: none">• Format: hh:mm PM	4 : 30 PM
status: string	The status of the live performance submission Available Statuses: <ul style="list-style-type: none">• Open: Submission hasn't been processed• On Hold – Invalid Evidence: proof submitted isn't valid• On Hold – No ticket price info: no information regarding the ticket price• On Hold – Incomplete Info: there is information that is missing• On Hold – Not Distributable: performance can't be paid• On Hold – Specify: See comments for additional details• Processed: performance submission has been processed• Duplicate: Performance already exists in SOCAN's system	PROCESSED

venue: string	Name of the venue the performance was in	"ACC"
artistName: string	Name of the artist that performed	"Justin Bieber"
adminComments: string	Any additional comments from Staff that processed the submission.	
contact: string	Contact information if more information is required	

Sample Response

```
{ memberNo: '9999999',
  confirmationNo: '172',
  performanceDate: '2018-10-10',
  performanceTime: '1 : 30 AM',
  status: 'PROCESSED',
  venue: 'TEST VENUE',
  artistName: 'MIKE JANET WEDS',
  adminComments: 'another test2',
  contact:
    'For more information on your performance status please contact members@socan.com' }
```

Error Response (HTML Status code 4xx) – Object

Property Name	Description	Example
errorType: string	The type of error it is	Validation Error
errors: array	An array of objects containing the error code, the affected field and the description of the error	See code sample
errorCode: string	The code of the error	10000001
field: string	The impacted field of the error	memberNo
description: string	The description of the error	Member number must be provided

Sample Response

```
body
{ errorType: 'Request Error',
  errors:
    [ { errorCode: '9907001',
      description: 'Invalid Query Parameter - memberNo' } ] }
```

Error Codes

Error Code	Description
1003001	Live Performance Notification was not found
9907001	Invalid Query Parameter - memberNo Or Invalid Query Parameter - confirmationNo

Live Performance Notification

Description

This operation can be used to notify SOCAN of a live performance of a SOCAN member. Proof of submission must be submitted along with the request. To upload a file, set content-type to “multipart/form-data”. To The multi part request must contain the following fields

Field Name/Type	Required	Description
nlmp: string	Required	JSON string containing the performance details (see below)
file: string	Conditional	A file containing the proof the performance took place. This is only mandatory for Canadian performances.
filename: string	Conditional	The name and extension of the file sent in the request. This is only mandatory for Canadian performances.

Host

Sandbox: <https://api.socan.ca/sandbox>

Production: <https://api.socan.ca/api>

Path

POST /v3/liveperformance

(See this [documentation](#) if you are still on the previous version using this path – POST /SubmitNLMP)

Request Headers

- Authorization: must include Bearer Token
- Content-type: multipart/form-data

Query Parameters

Field Name/Type	Required	Description	Example
apiKey: string Your SOCAN API Key	Required		Your API Key

NLMP JSON string Parameters

Name	Required	Description	Example
memberNo: string	Required	Unique SOCAN ID of the member the performance is submitted on behalf of <ul style="list-style-type: none">• Must be 7 or 8 digits• Must be numeric	“99999999”
performanceType: string	Required	An indicator to determine if the performance took place in Canada or outside of Canada <ul style="list-style-type: none">• “C” - Canadian• “NC” - Non-Canadian	“C”

programType: string	Required	An indicator to determine the type of event. Available values <ul style="list-style-type: none"> • “MF” - Music Festival • “PC” - Popular Music Concert • “BC” - Bar/Club • “SC” - Serious/Classical 	“MF”
artistName: string	Required	The name of the artist that performed	“Drake”
otherArtists: string [] Max array length: 10	Optional	List of other artists or group that performed (Array of Strings)	[“Drake”, “Sia”]
actType: string	Conditional	An indicator to show the type of act that performed. This field is required if programType is PC or BC. <ul style="list-style-type: none"> • “HA” – Artist is the Headline Act • “HN” – Artist is the Headline No support • “SA” – Artist is the Supporting Act • “CH” – Artist is the Co-Headline • “NA” – Not Applicable – please specify 	“HA”
actOther: string	Conditional	A string to allow the user to describe the type of act that performed. This field is required if actType is = “NA”	“Whatever Act”
numberWorks: string	Optional	Total number of songs that was performed	“5”
ticketRange: string	Condition	An indicator to show if the ticket price at the bar/club was over \$6.00. If ticket price is under \$6.00 the user should not be able to submit a notification to SOCAN as these concerts are not eligible for royalty distribution. This field is required if programType is “BC” <ul style="list-style-type: none"> • “y” - yes • “n” - no 	“y”
proofType: string	Required	The type of documentation that is being submitted to prove that the performance took place.	“ticket stub”

		<p>Proof Types and allowed programTypes</p> <ul style="list-style-type: none"> • Contract <ul style="list-style-type: none"> ○ Allowed Program Type – MF, PC, BC, SC • Newspaper <ul style="list-style-type: none"> ○ Allowed Program Type –MF, PC, BC, SC • Ticket stub <ul style="list-style-type: none"> ○ Allowed Program Type – MF, PC, BC, SC • Program <ul style="list-style-type: none"> ○ Allowed Program Type – MF, SC 	
performanceDate: string	Required	<p>The date in which the concert was performed</p> <ul style="list-style-type: none"> • It can't be in the future • Must have taken place in the last 18 months • Format of the date must be as followed: YYYY/MM/DD 	"2019/02/12"
performanceTime: string	Required	<p>The specific time at which the performance started</p> <ul style="list-style-type: none"> • Format: hh:mm PM • Valid hh values: 1 to 12 • Valid mm values: 00, 15, 30, 45 • Valid PM values: AM, PM 	"4 : 45 PM"
venueInformation: object	Required	<p>An object with the following attributes:</p> <ul style="list-style-type: none"> • venueName • venueType • venueCapacity • venueAddress • venuePhone • venueWebsite 	
promoterInformation: object	Required	<p>An object with the following attributes:</p> <ul style="list-style-type: none"> • promoter • promoterAddress • promoterPhone • promoterEmail • promoterContact 	

venueName: string	Required	The name of the establishment where the performance took place <ul style="list-style-type: none"> • Must be between 0 - 50 characters 	“Adelaide Hall”
venueType: string	Conditional	An indicator to show the type of establishment. This field is required on all Non-Canadian performances (if performanceType = NC) Venue Types – Values <ul style="list-style-type: none"> • Club/Bar/Restaurant – CBR • Theatre – THT • Arena/Stadium – AST • Hotel/Casino – HCN • Festival Grounds – FSG • City Part – CTP • Church – CCH • Coffee Shop – CFS • Amusement Park – AMP • Aquarium/Zoo – AQZ • College/University/School – CUS • Convention Center/Exhibit Hall – CEH • Museum – MSM • Shopping Center – SHC • Retail Store – RTS • Other – OTR 	“CBR”
venueCapacity: string	Conditional	An indicator which reflects the capacity of the venue. This field is required on all Non-Canadian performances (if performanceType = NC) Capacity & Values <ul style="list-style-type: none"> • 0 – 250 - L1 • 251 – 500 - L2 • 501 - 1000 - L3 • 1001 - 5000 - L4 • 5001 – 10000 - L5 • 10001 and Above - L6 	“L1”
venueAddress: object	Required	An object with the following attributes: <ul style="list-style-type: none"> • street1 • street2 	

		<ul style="list-style-type: none"> • City • provinceOrState • postalOrZip • country 	
street1: string	Required	Address Line 1 of the venue where the performance was performed <ul style="list-style-type: none"> • Must be between 0 - 40 characters 	"25 Brock St"
street2: string	Optional	Address Line 2 of the venue where the performance was performed <ul style="list-style-type: none"> • Must be between 0 - 40 characters 	
city: string	Required	City of the venue where the performance was performed <ul style="list-style-type: none"> • Must be between 0 - 30 characters 	"Toronto"
provinceOrState: string	Required	Province or State Code of the venue where the performance took place <ul style="list-style-type: none"> • Value cannot be other if country is United States or Canada <p>See (Appendix 2 for code values)</p>	"ON"
postalOrZip: string	Conditional	Postal or Zip Code of the venue where the concert was performed. Must be provided when it is a Non-Canadian Performance (if performanceType = NC) <ul style="list-style-type: none"> • Must be between 0 - 10 characters 	"m5L4k0"
country: string	Required	Country Code of the venue where the concert was performed <p>(Appendix 2 for code values)</p>	"CAN"
venueWebsite: string	Optional	The URL of the venue's website <ul style="list-style-type: none"> • Must be between 0 - 200 characters 	"www.site.com"
promoter: string	Required	The name of the company promoted the concert	
promoterAddress: object	Required	An object with the following attributes: <ul style="list-style-type: none"> • street1 • street2 • City • provinceOrState 	

		<ul style="list-style-type: none"> postalOrZip country 	
promoterPhone: string	Required	The telephone number of the company who promoted the concert Canadian – 10 – 14 Digits (Final 4 can be characters) Non-Canadian – 0 – 20 Characters	"4169892330"
promoterEmail: string	Optional	The email address of the promoter	"promoter@gmail.com"
promoterContact: string	Optional	The contact name for the company who promoted the concert	"John Snow"
compositions: object []	Required	List of compositions (Array of objects) with the following attributes <ul style="list-style-type: none"> originalTitle composer workNo 	
originalTitle: string	Required	The title of the song performed <ul style="list-style-type: none"> Must be between 0 - 50 characters 	"Its cold outside"
composer: string	Optional	The composer/author of the song that was performed All composer names entered must be the Surname. Multiple entries must be separated by a "/". <ul style="list-style-type: none"> Must be between 0 - 60 characters 	
workNo: string	Optional	This is a SOCAN generated unique number assigned to each song.	
comments: string	Optional	Any additional comments need to be included in the submission <ul style="list-style-type: none"> Must be between 0 - 250 characters 	

Example Request

```
const apiMultiRequest = (accessToken, apiUrl, json) => {
  const form = new FormData();
  form.append('file', fs.createReadStream('/nlp.txt'), 'nlp.txt');
  form.append('fileName', 'test.txt');
  const formContent = {
    'file': fs.createReadStream(__dirname + '/nlp.txt'),
    'nlp': {
      value: JSON.stringify(json),
      options: {
        contentType: 'application/json'
      }
    },
    'fileName': 'test.txt'
  }
  const options = {
    url: `${apiUrl}${npApiKey}`,
    method: "POST",
    headers: {
      'Authorization': `Bearer ${accessToken}`,
      'Content-type': `multipart/form-data; boundary=${form._boundary}`
    },
    formData: formContent
  }
  request(options, (err, res, body) => {
  })
};
```

Successful Response (HTML Status code - 200) – Object

Property Name	Description	Example
memberNo: string	The member number the request was submitted for	“9999999”
confirmationNo: string	The confirmation number of the request. Keep this number as it will be required to pull the updated status of your submission	
message: string	A message informing you the request was accepted	

```
{ memberNo: '9999999',
  confirmationNo: '179',
  message: 'ACCEPTED' }
```

Error Response (HTML Status code 500) – Object

Example Response

```
{ errorType: 'Request Validation Error',
  errors: [
    { errorCode: '0100003',
      field: 'memberNo',
      description: 'Must be numeric' },
    { errorCode: '0300002',
      field: 'programType',
      description: 'Invalid code, see documentation for details' } ] }
```

Property Name	Description	Example
errorType: string	The type of error it is	Request Validation Error
errors: object []	An array of objects containing the error code, the affected field and the description of the error	See code sample
errorCode: string	The code of the error	1000001
field: string	The impacted field of the error	memberNo
description: string	The description of the error	Member number must be provided

Error Codes

Error Code	Parameter	Description
0100001	memberNo	Must be provided, it is a mandatory field
0100002	memberNo	Must be 7 or 8 Characters
0100003	memberNo	Must be numeric
0200001	performanceType	Must be provided, it is a mandatory field
0200002	performanceType	Invalid Code, see appendix for details

0300001	programType	Must be provided, it is a mandatory field
0300002	programType	Must be provided, it is a mandatory field
0400001	artistName	Must be provided, it is a mandatory field
0400002	artistName	Length must be between 0 - 50 characters
0500001 – [index]	otherArtists	Length must be between 0 - 50 characters
0600001	proofType	Must be provided, it is a mandatory field
0600002	proofType	Invalid Code, see appendix for details
0800001	actOther	Must be provided when actType is NA
0800001	actOther	Length must be between 0 - 50 characters
0900001	numberWorks	Entered Value must be Numeric
0900002	numberWorks	Entered Value must be between 0 - 10000
1000001	ticketRange	Must be provided when Program is BC
1000002	ticketRange	Must be a Boolean ('Y' or 'N')
1000003	ticketRange	Canadian Performances must have range above \$6
1200001	performanceDate	Must be provided, it is a mandatory field
1200002	performanceDate	Must be in Date Format – YYYY/MM/DD
1200003	performanceDate	Date can't be in the future
1200004	performanceDate	Date can't be older than 18 Months
1300001	venueName	Must be provided, it is a mandatory field
1300002	venueName	Must be between 0 - 50 characters
1302001	venueType	Must be provided when performance is Non-Canadian
1301002	venueType	Invalid Code, see appendix for details
1302001	venueCapacity	Must be provided when performance is Non-Canadian
1302002	venueCapacity	Invalid Code, see appendix for details
1400001	venueAddress.street1	Must be provided, it is a mandatory field
1400001	venueAddress.street1	Must be between 0 - 40 characters
1500001	venueAddress.street2	If entered must be between 0 - 40 characters
1600001	venueAddress.city	Must be provided, it is a mandatory field
1600002	venueAddress.city	Must be between 0 - 30 characters
1700001	venueAddress.provinceOrState	Must be provided, it is a mandatory field
1700002	venueAddress.provinceOrState	Invalid Code, see appendix for details
1800001	venueAddress.postalOrZip	Must be provided when performance is Non-Canadian
1800002	venueAddress.postalOrZip	Must be between 0 - 10 characters
1900001	venueAddress.country	Must be provided, it is a mandatory field
1900002	venueAddress.country	Invalid Code, see appendix for details
1901001	venueAddress.country	Venue should not be in Canada when International performance
1901002	venueAddress.country	Venue must be in Canada when Canadian performance
1920001	venueWebsite	If entered must be between 0 - 200 characters
2000001	performanceTime	Must be provided, it is a mandatory field

2000002	performanceTime	Must be in the time format (hh : mm PM)
2100001	promoter	Must be provided, it is a mandatory field
2100002	promoter	Must be between 0 - 50 characters
2200001	promoterAddress.street1	Must be provided, it is a mandatory field
2200002	promoterAddress.street1	Must be between 0 - 40 characters
2300001	promoterAddress.street2	Must be between 0 - 40 characters
2400001	promoterAddress.city	Must be provided, it is a mandatory field
2400002	promoterAddress.city	Must be between 0 - 30 characters
2500001	promoterAddress.provinceOrState	Must be provided, it is a mandatory field
2500002	promoterAddress.provinceOrState	Invalid Code, see appendix for details
2700001	promoterAddress.country	Must be provided, it is a mandatory field
2800001	promoterPhone	Must be provided, it is a mandatory field
2800002	promoterPhone	Canadian Performance – must be between 10 - 14 digits
2800003	promoterPhone	Non-Canadian Performance – must be between 0 – 20 characters
3100001	compositions	You must enter between 1 - 50 compositions
9900002		Request could not be saved in DB
9900003		File could not be processed uploaded/saved
9900004		Invalid memberNo. Participant does not exist.
9900005		Duplicate submission
9906001		Invalid JSON {param}: is not defined in the schema and the schema does not allow additional properties
9907001	nlmp	Missing JSON
9920001	File	Error – missing file
9920001	fileName	Error – missing parameter fileName

Member Verification

Description

Member Verification API allows third party queries to be made to SOCAN with a combination of an artist name and IPI number. If the information submitted does, or does not correlate to a SOCAN member, SOCAN will verify and acknowledge that and return a response to the third part.

Request

Host

Sandbox: <https://api.socan.ca/sandbox>

Production: <https://api.socan.ca/api>

Path

POST /MemberVerification

See detail documentation [here](#)

Join SOCAN

Description

The Join SOCAN on API allows third party apps and services to facilitate a non-member of SOCAN to initiate the application for membership. The API will allow the third party to capture and transmit the required information to SOCAN and put the applicant in contact with SOCAN directly to complete the application process.

Request

Host

Sandbox: <https://api.socan.ca/sandbox>

Production: <https://api.socan.ca/api>

Path

POST /JoinSOCAN

See detail documentation [here](#)

Work Registration – Submission

Description

The Works Registration API allows submission of works registrations through SOCAN's secure gateway. These work registrations are tagged with all applicable metadata, sorted in SOCAN's database for proper identification wherever this music is played.

Request

Host

Sandbox: <https://api.socan.ca/sandbox>

Production: <https://api.socan.ca/api>

Path

POST /SubmitWork

See detail documentation [here](#)

Work Registration – Status

Description

When utilizing the Works Registration API, the Request Status API will allow the return of the current status of the work registration submission from SOCAN.

Request

Host

Sandbox: <https://api.socan.ca/sandbox>

Production: <https://api.socan.ca/api>

Path

POST /getStatus

See detail documentation [here](#)

Work Registration – Acknowledgement

Description

When utilizing the Works Registration API, the Acknowledgement API returns an acknowledgement file that details Works submitted that have been received and accepted by SOCAN.

Request

Host

Sandbox: <https://api.socan.ca/sandbox>

Production: <https://api.socan.ca/api>

Path

POST /getACK

See detail documentation [here](#)

Appendix

Appendix 1: Country Table Codes

Country	Value
ARUBA	ABW
AFGHANISTAN	AFG
ANGOLA	AGO
ANGUILLA	AIA
ALBANIA	ALB
ANDORRA	AND
NETHERLANDS ANTILLES	ANT
UNITED ARAB EMIRATES	ARE
ARGENTINA	ARG
ARMENIA	ARM
AMERICAN SAMOA	ASM
ANTARCTICA	ATA
FRENCH SOUTHERN TER.	ATF
ANTIGUA AND BARBUDA	ATG
AUSTRALIA	AUS
AUSTRIA	AUT
AZERBAIJAN	AZE
AZORES	AZO
BURUNDI	BDI
BELGIUM	BEL
BENIN	BEN
BURKINA FASO	BFA
BANGLADESH	BGD
BULGARIA	BGR
BAHRAIN	BHR
BAHAMAS	BHS
BOSNIA-HERZEGOVINA	BIH
BELARUS	BLR
BELIZE	BLZ
BERMUDA	BMU
BOLIVIA	BOL
BRAZIL	BRA
BARBADOS	BRB
BRUNEI DARUSSALAM	BRN
BHUTAN	BTN
BOUVET ISLAND	BVT
BOTSWANA	BWA
CENTRAL AFRICAN REP.	CAF
CANADA	CAN
COCOS (KEELING) ISL.	CCK
SWITZERLAND	CHE
CHILE	CHL

CHINA	CHN
IVORY COAST	CIV
CAMEROON	CMR
CANARY ISLANDS	CNP
DEMOC. REP. OF CONGO	COD
REPUBLIC OF CONGO	COG
COOK ISLANDS	COK
COLOMBIA	COL
COMOROS	COM
CAPE VERDE	CPV
COSTA RICA	CRI
CUBA	CUB
CHRISTMAS ISLAND	CXR
CAYMAN ISLANDS	CYM
CYPRUS	CYP
CZECH REPUBLIC	CZE
GERMANY	DEU
DJIBOUTI	DJI
DOMINICA	DMA
DENMARK	DNK
DOMINICAN REPUBLIC	DOM
ALGERIA	DZA
ECUADOR	ECU
EGYPT	EGY
ERITREA	ERI
WESTERN SAHARA	ESH
SPAIN	ESP
ESTONIA	EST
ETHIOPIA	ETH
FINLAND	FIN
FIJI	FJI
FALKLAND ISLANDS	FLK
FRANCE	FRA
FAROE ISLANDS	FRO
MICRONESIA	FSM
GABON	GAB
UNITED KINGDOM	GBR
GEORGIA	GEO
GHANA	GHA
GIBRALTAR	GIB
GUINEA	GIN
GUADELOUPE	GLP
GAMBIA	GMB
GUINEA-BISSAU	GNB
EQUATORIAL GUINEA	GNQ
GUERNSEY	GNS
ISLE OF MAN	GRB

GREECE	GRC
GRENADA	GRD
GREENLAND	GRL
GUATEMALA	GTM
FRENCH GUIANA	GUF
GUAM	GUM
GUYANA	GUY
HONG KONG	HKG
HEARD/MCDONALD ISLD.	HMD
HONDURAS	HND
CROATIA	HRV
HAITI	HTI
HUNGARY	HUN
INDONESIA	IDN
INDIA	IND
BR.INDIAN OCEAN TER.	IOT
IRELAND	IRL
IRAN	IRN
IRAQ	IRQ
ICELAND	ISL
ISRAEL	ISR
ITALY	ITA
JAMAICA	JAM
JORDAN	JOR
JAPAN	JPN
JERSEY	JRS
KAZAKHSTAN	KAZ
KENYA	KEN
KYRGYZSTAN	KGZ
CAMBODIA	KHM
KIRIBATI	KIR
ST. KITTS AND NEVIS	KNA
REPUBLIC OF KOREA	KOR
KUWAIT	KWT
DEMOCR. REP. OF LAO	LAO
LEBANON	LBN
LIBERIA	LBR
LIBYAN AR.JAMAHIRIYA	LBY
SAINT LUCIA	LCA
LIECHTENSTEIN	LIE
SRI LANKA	LKA
LESOTHO	LSO
LITHUANIA	LTU
LUXEMBOURG	LUX
LATVIA	LVA
MACAO	MAC
MOROCCO	MAR

MONACO	MCO
REPUBLIC OF MOLDOVA	MDA
MADAGASCAR	MDG
MADEIRA ISLANDS	MDR
MALDIVES	MDV
MEXICO	MEX
MARSHALL ISLANDS	MHL
MACEDONIA	MKD
MALI	MLI
MALTA	MLT
MYANMAR	MMR
MONTENEGRO	MNE
MONGOLIA	MNG
NORTHERN MARIANA IL.	MNP
MOZAMBIQUE	MOZ
MAURITANIA	MRT
MONTSERRAT	MSR
MARTINIQUE	MTQ
MAURITIUS	MUS
MALAWI	MWI
MALAYSIA	MYS
MAYOTTE	MYT
NAMIBIA	NAM
NEW CALEDONIA	NCL
NIGER	NER
NORFOLK ISLANDS	NFK
NIGERIA	NGA
NICARAGUA	NIC
NIUE	NIU
NETHERLANDS	NLD
NORWAY	NOR
NEPAL	NPL
NAURU	NRU
NEW ZEALAND	NZL
OMAN	OMN
PAKISTAN	PAK
PANAMA	PAN
PITCAIRN	PCN
PERU	PER
PHILIPPINES	PHL
PALAU	PLW
PAPUA NEW GUINEA	PNG
POLAND	POL
PUERTO RICO	PRI
REP. OF NORTH KOREA	PRK
PORTUGAL	PRT
PARAGUAY	PRY

PALESTINIAN TERR.	PSE
FRENCH POLYNESIA	PYF
QATAR	QAT
REUNION	REU
ROMANIA	ROU
RUSSIAN FEDERATION	RUS
RWANDA	RWA
SAUDI ARABIA	SAU
SUDAN	SDN
SENEGAL	SEN
SINGAPORE	SGP
SOUTH GEORGIA	SGS
SAINT HELENA	SHN
SVALBARD ISLANDS	SJM
SOLOMON ISLANDS	SLB
SIERRA LEONE	SLE
EL SALVADOR	SLV
SAN MARINO	SMR
SOMALIA	SOM
ST. PIERRE & MIQUELON	SPM
SAO TOME & PRINCIPE	STP
SURINAME	SUR
SLOVAKIA	SVK
SLOVENIA	SVN
SWEDEN	SWE
SWAZILAND	SWZ
SEYCHELLES	SYC
SYRIAN ARAB REPUBLIC	SYR
TURKS & CAICOS ILDS.	TCA
CHAD	TCD
TOGO	TGO
THAILAND	THA
TAJIKISTAN	TJK
TOKELAU	TKL
TURKMENISTAN	TKM
TIMOR-LESTE	TMP
TONGA	TON
TRINIDAD AND TOBAGO	TTO
TUNISIA	TUN
TURKEY	TUR
TUVALU	TUV
TAIWAN	TWN
TANZANIA	TZA
UGANDA	UGA
UKRAINE	UKR
US OUTLYING ISLANDS	UMI
URUGUAY	URY

UNITED STATES	USA
UZBEKISTAN	UZB
VATICAN CITY STATE	VAT
ST.VINCENT GRENADINE	VCT
VENEZUELA	VEN
BR. VIRGIN ISLANDS	VGB
U.S. VIRGIN ISLANDS	VIR
VIETNAM	VNM
VANUATU	VUT
WALLIS & FUTUNA ILDS	WLF
SAMOA	WSM
YEMEN	YEM
SERBIA/MONTENEGRO	YUG
SOUTH AFRICA	ZAF
ZAMBIA	ZMB
ZIMBABWE	ZWE

Appendix 2: Province/State Codes

Province/State	Value
Other	O
Alberta	AB
British Columbia	BC
Manitoba	MB
New Brunswick	NB
Newfoundland and Labrador	NL
Northwest Territories	NT
Nova Scotia	NS
Nunavut	NU
Ontario	ON
Prince Edward Island	PE
Québec	QC
Saskatchewan	SK
Yukon Territory	YT
Alabama	AL
Alaska	AK
Arizona	AZ
Arkansas	AR
California	CA
Colorado	CO
Connecticut	CT
Delaware	DE
District of Columbia	DC
Florida	FL
Georgia	GA

Hawaii	HI
Idaho State	ID
Illinois	IL
Indiana	IN
Iowa	IA
Kansas	KS
Kentucky	KY
Louisiana	LA
Maine	ME
Maryland	MD
Massachusetts	MA
Michigan	MI
Minnesota	MN
Mississippi	MS
Missouri	MO
Montana	MT
Nebraska	NE
Nevada	NV
New Hampshire	NH
New Jersey	NJ
New Mexico	NM
New York	NY
North Carolina	NC
North Dakota	ND
Ohio	OH
Oklahoma	OK
Oregon	OR
Pennsylvania	PA
Rhode Island	RI
South Carolina	SC
South Dakota	SD
Tennessee	TN
Texas	TX
Utah	UT
Vermont	VT
Virginia	VA
Washington	WA
West Virginia	WV
Wisconsin	WI
Wyoming	WY

Change Log

Thursday May 14, 2020

- Added the following error codes to Live Performance API
 - Code - 1901001 - venueAddress.country
 - Venue should not be in Canada when International performance
 - Code 1901002 - venueAddress.country
 - venue must be in Canada when Canadian performance