API Quickstart

iSpot.tv

Authentication

The iSpot API utilizes oAuth2.0 for authentication, which involves using your client_id and client_secret to make a POST request to the authentication endpoint to retrieve a bearer token. This token is used for each subsequent API call in order to return data. Further instructions as well as code samples can be found here.

Credentials

Your credentials will be provided to you by your Customer Success representative and should be obtained prior to moving past this step. These credentials are for demonstration purposes only, but yours will be the same number of characters as outlined below.

Client ID

Your client_id is a 20-character alphanumeric value used to authenticate

Ex: 89ab614c1732d98e123f

Client Secret

Your client_secret is a 40-character alphanumeric value

Ex: Tr1f3k6PzStGhQcWLuMAcKdXr4g4s6rUcHwyTSby

Grant Type

When authenticating the grant_type parameter will always be the string 'client_credentials'

Retrieving a Bearer Token

Using the given credentials, you will make a POST request to the endpoint https://api.ispot.tv/v4/oauth2/token and pass your client_id, client_secret and grant_type as a parameter in the request headers. Authentication will vary depending on which method you are using to interact with our API, but the mechanics of the request are largely the same.

Token Expiration

Please note that your token is valid for a period of 24 hours, at which point you will need to retrieve a new token. It is recommended that you limit the amount of token requests per day and only request a new one when absolutely necessary.

Example using curl from command line

Authenticating

```
curl --request POST \

--url 'https://api.ispot.tv/v4/oauth2/token' \

--data 'client_id=CLIENT_ID&client_secret=CLIENT_SECRET&grant_type=client_credentials'
```

After running the previous command, it should return your Bearer Token as shown below:

Sample Response with Bearer Token

```
{"token_type":"Bearer","expires_in":86400,"access_token":"eyJ0eTAiOiJKV1QiLCJhbGciOiJSUzI1
NiJ9.eyJerWQiOiIyWeLwMPBkMDdjZjRmZWRiYjZkNSIsImp0aSI6IjUyNDgxY2YzZTVmRQTkOGFiZG
I1YWQ3NWRhMjg4NDFIYjE3MTE2M2JmZGNiYjFkZWJhOTg1ZDRjZmUyZjY0ODExOTBIYjM5NWI1N
WViODM4IiwiaWF0IjoiMTYxSjg3MTIxTR45OTU0NzMiLCJuYmYiOiIxNjE2ODcxMjExLjk5NTQ3NyIsI
mV4cCI6IjE2MTY5NTc2MTEuOTg2NDE5Iiwic3ViIjoiIiwic2NvcGVzIjpbXX0.jh0UPXZR6CgPWMCTR
YUpQMk5AfBIIZJ3Vcw1Tlh1ZLeiAh3pZwAKsLVEt2mAvbC_XD77CWxx_1yub-
O2TX2Ty4GrjZms3XQBJw_xE6A6LQ77XNZ5E3iX0xEFOvIAIKx7MPtOGIHLsZg0W-2cDepLyE-
xUMIpIp6huI2ndId12rZOcEQgtT44W7XIKUIFtpGZCGdjQf13AfHYIfT2eInTRgZifnNLjpAkW9mx1Qas5
LXzLx6PTLUncIf3kCAwRSsTTxfmspKZDfI58I6LGxVW24p1Uw71RrdkJGzuuUdIpz25VTKxx_3ZSB8
6mEwzDABYaISGyqgXv0vi9eLoR7d_ug"}
```

Your bearer token is a 671-character alphanumeric value valid for 24 hours (86400 seconds)

Making an API call

The bearer token returned in the previous step will be used to make each subsequent API request. You'll want to copy and paste or store your bearer token in a variable, which will be passed through the request headers as a parameter. *Ex: --header 'Authorization: Bearer TOKEN_HERE'*

Depending on which method you are using to make requests and return data, the process will vary greatly, however the structure of the endpoint including filters and parameters will be the same. An exhaustive list of endpoints available via the API can be found HERE, you will need a username and password in order to access which should have been provided to you by your Customer Success representative. *Please note these credentials are different than your client_id and client_secret used for authentication!*

Structure of an API Request

{Base Endpoint} *{Content Type}* *{Query String & Filters}*

https://api.ispot.tv/v4/ metrics/audience/airings? filter[start_date]=2021-01-01&filter[end_date]=...

Filters

One of the most critical elements of any API request are the filters and parameters. These are necessary in order to return a manageable data set that includes only the necessary data that you would like to return. A list of the most common filters and includes can be found HERE.

Pagination

Using pagination ensures that returned API responses are easier to handle. For example, a given response may retrieve hundreds of thousands of results that are likely not very useful. Pagination also allows you to request multiple pages in parallel. The pagination parameters in this section let you control the volume of information that is returned in the response.

Values specified in the pagination parameters must be valid and within the specified range in order for the response to succeed. Note the default and maximum values for this in the following table.

iSpot REST API pagination parameters filters

Type	Parameter	Value
Use to specify page number.	*page[number]=*	Number
Use to specify response values per page.	*page[size]=*	Number Default: 100 Max: 10,000

The following sample request uses both parameters:

https://api.ispot.tv/v4/metrics/tv/airings?page[number]=3&page[size]=10000&filter[start_date]=2019-01-01&filter[end_date]=2019-01-07&sort=est_spend,-date

To get one day's worth of data (24 hours), make the start and end date filters the same in your API request.

Example: filter[start_date]=2019-06-05&filter[end_date]=2019-06-05

Pagination response format

Pagination data is located in the metadata section of the response. The metadata also includes next page and previous page links in order to traverse all available pages.

```
"pagination": {
        "total": #,                  →    Total response values
        "count": #,                  →    Values on given page
        "per_page": #,               →    Values per page
        "current_page": #,           →    Current page number
        "total_pages": #,            →    Total number of pages
        "links": {
                "previous": "link"
                "next": "link"
```

Rate limiting

Rate limiting controls the amount of incoming and outgoing network traffic. For example, if a particular API service is configured to allow 100 requests per minute, when a user exceeds that request limit, an error is triggered. Implementing rate limits allows for better data flow and data management.

The returned HTTP headers from an iSpot API request provide rate limit status on the following: the maximum number of requests you're permitted to make per minute, the number of requests remaining in the current rate limit window, and the amount of Coordinated Universal Time (UTC) in seconds before the current rate limit window resets. In this way, iSpot enforces rate limits to manage the network load. The rate limits in the following table apply to iSpot REST API endpoints.

iSpot REST API endpoint rate limits

Endpoint	Request limit	Reset window in seconds
/v4/access	100	900
/v4/account	100	900
/v4/airings	300	60
/v4/brands	50	60
/v4/celebrities	50	60
/v4/events	50	60
/v4/industries	50	300
/v4/metrics	75	60
/v4/movies	50	60
/v4/networks	50	60
/v4/notifications	100	60
/v4/product-categories	50	60
/v4/products	50	60
/v4/shows	50	60
/v4/spots	50	60
/v4/users	50	60

Note: iSpot API rate limits may change without notice.

More Documentation

For further documentation, please refer to the following API Integration Guide or the Developer Site.

Sample API Templates

Below are some examples demonstrating API calls using a few different methods/languages that should be used as a guide to get you started. The developer site also contains further examples for each endpoint written in 7 different programming languages. Please reach out to your Customer Success representative if you have any questions or need any assistance.

Curl

The commands below demonstrate authenticating and making a request to the API. These can be run in many different command line/shell environments and may vary slightly. Please ensure that you replace the CLIENT_ID and CLIENT_SECRET with your own credentials to authenticate, and replace the bearer token in your subsequent API call with the token returned by the first command.

Authentication (to retrieve Bearer Token)

```
curl --request POST \
  --url 'https://api.ispot.tv/v4/oauth2/token' \
  --data 'client_id=CLIENT_ID&client_secret=CLIENT_SECRET&grant_type=client_credentials'
```

Making Request (returning data)

```
curl -g --request GET \
    --url 'https://api.ispot.tv/v4/metrics/audience/airings?filter[start_date]=2021-01-
01&filter[end_date]=2021-01-01&page[number]=1&page[size]=10000' --header 'Authorization:
Bearer YOUR_TOKEN_HERE'
```

Example request with token:

```
curl -g --request GET \
    --url 'https://api.ispot.tv/v4/metrics/audience/airings?filter[start_date]=2021-01-
01&filter[end_date]=2021-01-01&page[number]=1&page[size]=10000' --header 'Authorization:
Bearer
eyJ0eTAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJerWQiOiIyWeLwMPBkMDdjZjRmZWRiYjZkNSIsImp0
aSI6IjUyNDgxY2YzZTVmRQTkOGFiZGI1YWQ3NWRhMjg4NDFlYjE3MTE2M2JmZGNiYjFkZWJhOTg1
ZDRjZmUyZjY0ODExOTBlYjM5NWI1NWViODM4IiwiaWF0IjoiMTYxSjg3MTIxTR45OTU0NzMiLCJuYm
YiOiIxNjE2ODcxMjExLjk5NTQ3NyIsImV4cCI6IjE2MTY5NTc2MTEuOTg2NDE5Iiwic3ViIjoiIiwic2NvcG
VzIjpbXX0.jh0UPXZR6CgPWMCTRYUpQMk5AfBIlZJ3Vcw1TIh1ZLeiAh3pZwAKsLVEt2mAvbC_XD77
CWxx_1yub-O2TX2Ty4GrjZms3XQBJw_xE6A6LQ77XNZ5E3iX0xEFOvIAIKx7MPtOGIHLsZg0W-
2cDepLyE-
xUMIpIp6hul2ndId12rZOcEQgtT44W7XIKUIFtpGZCGdjQf13AfHYIfT2eInTRgZifnNLjpAkW9mx1Qas5
LXzLx6PTLUncIf3kCAwRSsTTxfmspKZDfI58I6LGxVW24p1Uw71RrdkJGzuuUdIpz25VTKxx_3ZSB8
6mEwzDABYaISGyqgXv0vi9eLoR7d_ug'
```

Python

```
#/venv/bin/python3
import http.client,json,csv,requests
import pandas as pd
from pandas import json_normalize

client_id = 'YOUR_CLIENT_ID_HERE'
client_secret = 'YOUR_CLIENT_SECRET_HERE'

def get_token():
   conn = http.client.HTTPSConnection("api.ispot.tv")
   payload =
"client_id={CLIENT_ID}&client_secret={CLIENT_SECRET}&grant_type=client_credentials".format(CLIENT_ID=client_id,
CLIENT_SECRET=client_secret)
   headers = {
      'content-type': "application/x-www-form-urlencoded"
   }
   conn.request("POST", "/v4/oauth2/token", payload, headers)

   res = conn.getresponse()
   token_raw = res.read()
   data_parsed = json.loads(token_raw.decode("utf-8"))

   access_token = data_parsed['access_token']
   conn.close()
   return (access_token)

conn = http.client.HTTPSConnection("api.ispot.tv")
payload = ""
headers = {'Authorization': "Bearer {TOKEN}".format(TOKEN=str(get_token())),
}

#######
## Set filters, parameters and define the api request.
endpoint = '/v4/metrics/audience/airings?'
page_size = '&page[size]=10000' # Do not alter
page_number = '&page[number]=1' # Do not alter
start_date = 'filter[start_date]=2021-01-01'
end_date = '&filter[end_date]=2021-01-14'
filters = '&filter[industry]=119&filter[airing_type]=N,R&filter[national_only]=1'
includes =
'&include=brand,parent_brand,spot,creative,day_of_week,day_part,episode,industry,network,genre,sub_genre,show,pro
duct'
params =  start_date + end_date + filters + includes
api_request = endpoint + params + page_size

conn.request("GET", api_request + page_number, payload, headers, )
res = conn.getresponse()
request_id = res.headers.get('X-Request-ID')
```

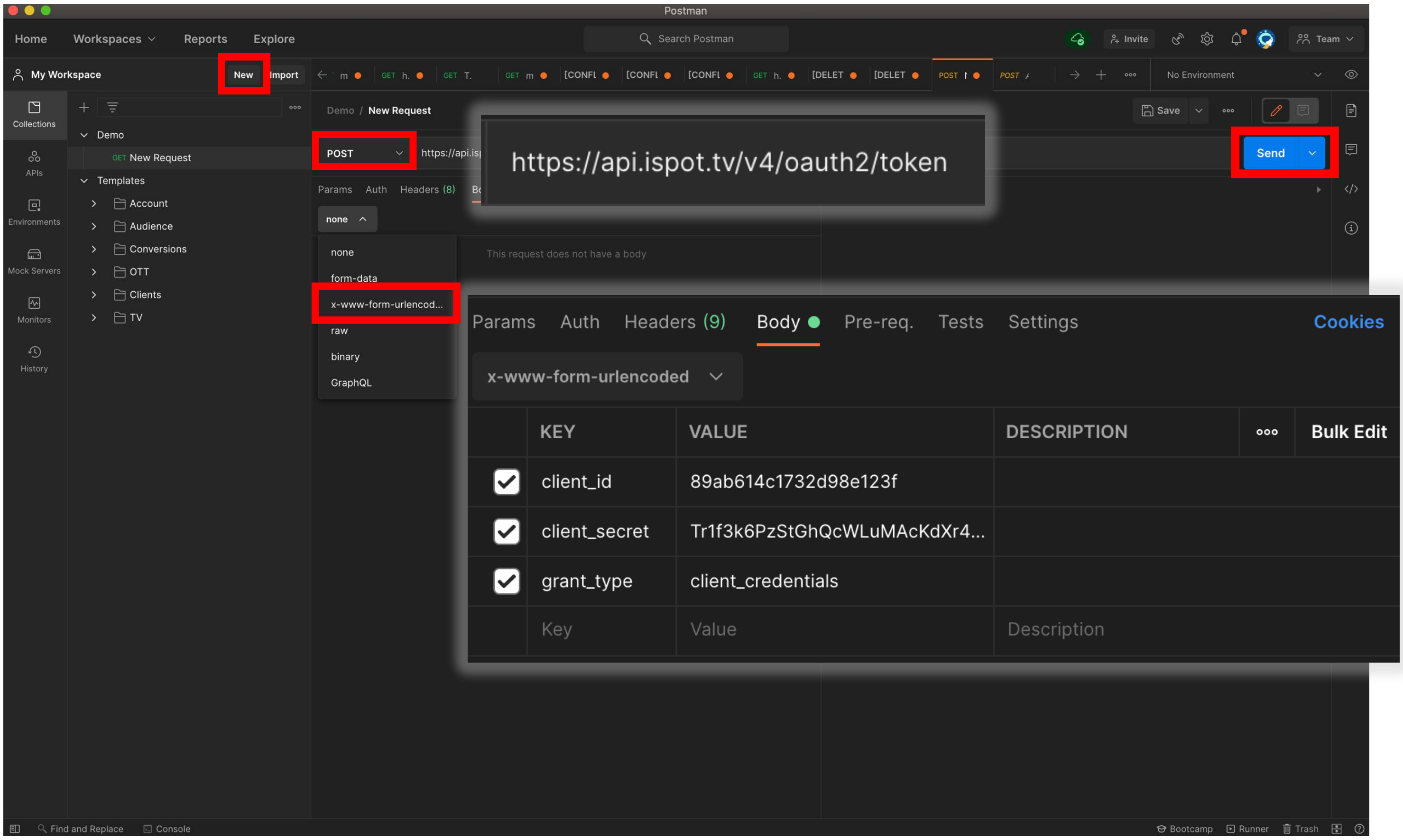
```
request_datetime = res.headers.get('Date')
print('\r\nRequest ID: {REQUEST_ID}\r\nDatetime: {REQUEST_DATETIME}\r\nStatus:
{STATUS}'.format(REQUEST_ID=request_id,REQUEST_DATETIME=request_datetime,STATUS=res.status))
final = res.read()
jsonResponse = json.loads(final.decode('utf-8'))

########
# Determine results and page data. JSON normalize converts all keys into headers for csv output
p = 0

p_total = jsonResponse.get('meta')['pagination']['total_pages']
total_values = jsonResponse.get('meta')['pagination']['total']
current_page = jsonResponse.get('meta')['pagination']['current_page']
print('\r\n{NRESULTS} results across {PAGE_NUM} pages\r\n'.format(NRESULTS=total_values, PAGE_NUM=p_total))
print('Completed: ' + api_request + page_number)
jsonResponse = jsonResponse['data']
json_norm = json_normalize(jsonResponse, sep='_')
if p == 0:
    json_norm.to_csv('output.csv', mode='w', sep=',', header=True, index=False)
else:
    json_norm.to_csv('output.csv', mode='a', sep=',', header=False, index=False)

while p < (p_total + 1):
    try:
        p += 1
        conn.request("GET", api_request + "&page[number]=" + str(int(current_page + 1)), payload, headers, )
        res = conn.getresponse()
        final = res.read()
        jsonResponse = json.loads(final.decode('utf-8'))
        jsonResponse = jsonResponse['data']
        json_norm = json_normalize(jsonResponse, sep='_')
        json_norm.to_csv('output.csv', mode='a', sep=',', header=False, index=False)
        print('Completed:
https://api.ispot.tv{ENDPOINT}&page[number]={CURRENT}'.format(ENDPOINT=api_request,CURRENT=str(int(current_pa
ge + 1))))
        current_page += 1
    except Exception:
        pass
else:
    pass
```

Postman

The following instructions describe how to retrieve your Bearer Token in Postman as well as how to get data via API requests.

To Retrieve your Bearer Token

1. In *My Workspace*, once you've added a new Collection, and expanded it, click *New* and then *Add a Request*. You can name it Authenticate.
2. Select POST from the dropdown.
3. Paste the endpoint https://api.ispot.tv/v4/oauth2/token
4. Click Body and select x-www-form-urlencoded
5. Enter the KEY and VALUE with your client_id, client_secret, and the grant_type set to client_credentials.
6. Click Send



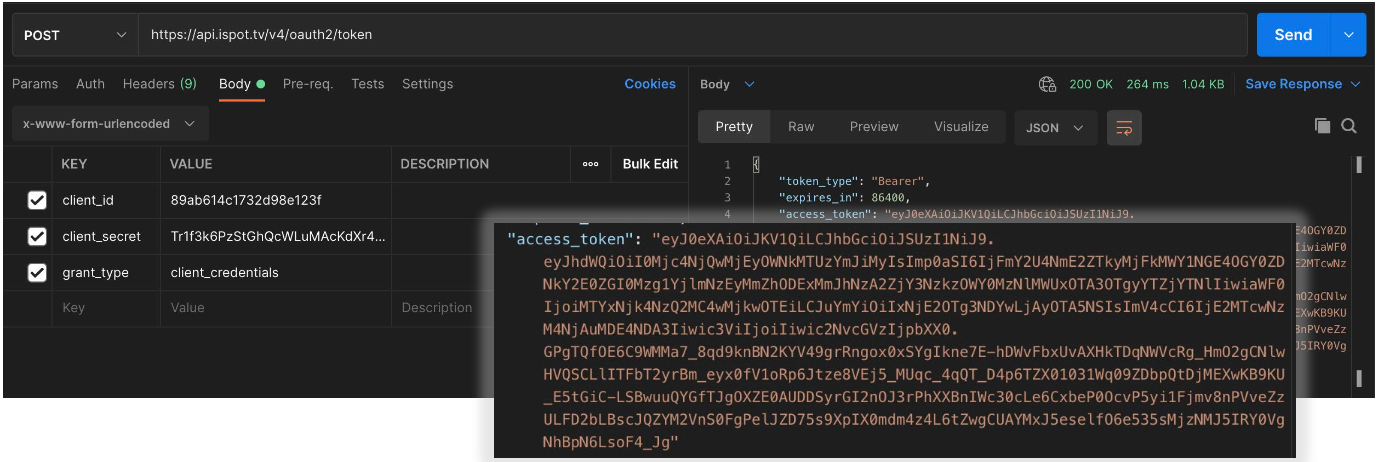
POST https://api.ispot.tv/v4/oauth2/token

Params | Auth | Headers (9) | Body ● | Pre-req. | Tests | Settings | Cookies

x-www-form-urlencoded

	KEY	VALUE	DESCRIPTION
☑	client_id	89ab614c1732d98e123f	
☑	client_secret	Tr1f3k6PzStGhQcWLuMAcKdXr4...	
☑	grant_type	client_credentials	
	Key	Value	Description

The Bearer Token will then appear at the bottom



POST https://api.ispot.tv/v4/oauth2/token

x-www-form-urlencoded

	KEY	VALUE	DESCRIPTION
☑	client_id	89ab614c1732d98e123f	
☑	client_secret	Tr1f3k6PzStGhQcWLuMAcKdXr4...	
☑	grant_type	client_credentials	
	Key	Value	Description

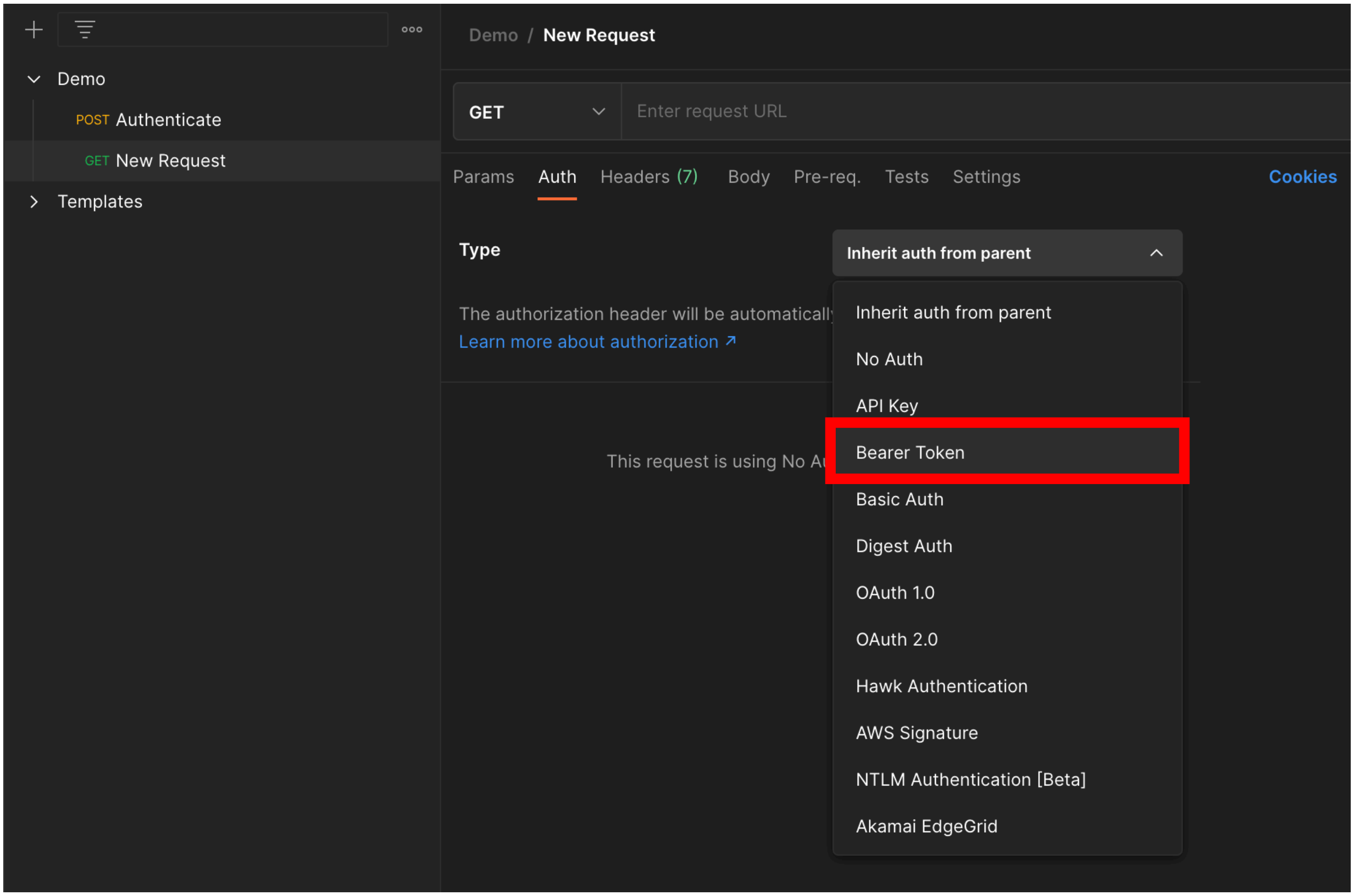
200 OK 264 ms 1.04 KB

```
{
    "token_type": "Bearer",
    "expires_in": 86400,
    "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.
```

```
"access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.
    eyJhdWQiOiI0Mjc4NjQwMjEyOWNkMTUzYmJiMyIsImp0aSI6IjFmY2U4NmE2ZTkyMjFkMWY1NGE4OGY0ZD
    NkY2E0ZGI0Mzg1YjlmNzEyMmZhODExMmJhNzA2ZjY3NzkzOWY0MzNlMWUxOTA3OTgyYTZjYTNlIiwiaWF0
    IjoiMTYxNjk4NzQ2MC4wMjkwOTEiLCJuYmYiOiIxNjE2OTg3NDYwLjAyOTA5NSIsImV4cCI6IjE2MTcwNz
    M4NjAuMDE4NDA3Iiwic3ViIjoiIiwic2NvcGVzIjpbXX0.
    GPgTQfOE6C9WMMa7_8qd9knBN2KYV49grRngox0xSYgIkne7E-hDWvFbxUvAXHkTDqNWVcRg_Hm02gCNlw
    HVQSCLlITFbT2yrBm_eyx0fV1oRp6Jtze8VEj5_MUqc_4qQT_D4p6TZX01031Wq09ZDbpQtDjMEXwKB9KU
    _E5tGiC-LSBwuuQYGfTJgOXZE0AUDDSyrGI2nOJ3rPhXXBnIWc30cLe6CxbeP0OcvP5yi1Fjmv8nPVveZz
    ULFD2bLBscJQZYM2VnS0FgPelJZD75s9XpIX0mdm4z4L6tZwgCUAYMxJ5eself06e535sMjzNMJ5IRY0Vg
    NhBpN6LsoF4_Jg"
```

To Retrieve Actual Data

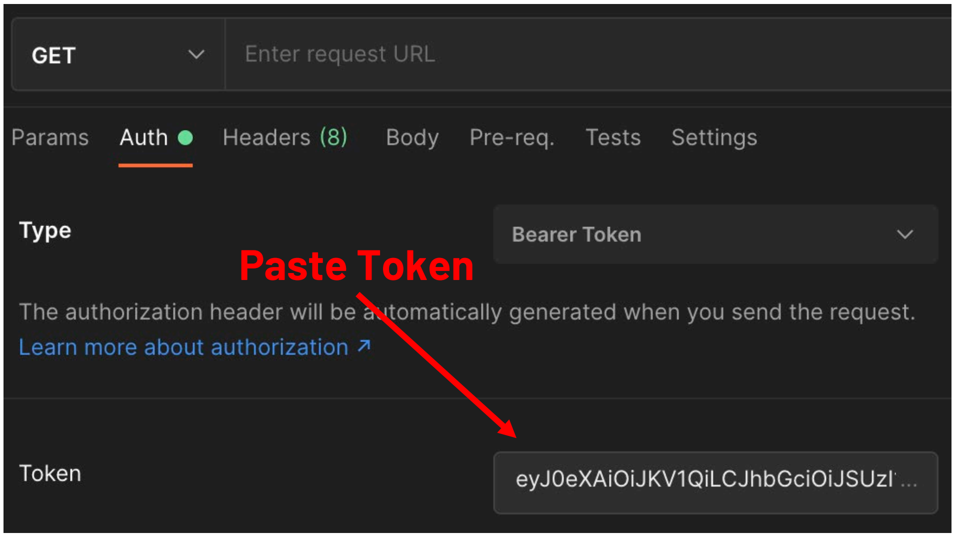
1. Create another New Request by clicking New, and creating a new request as you did before.
2. This remains as a GET. Enter the ispot request URL in the box next to GET:
   https://api.ispot.tv/v4/metrics/tv/airings
3. Click Auth or Authorization
4. From the Type dropdown box, select Bearer Token



The authorization header will be automatically

This request is using No Au

5. Paste the Bearer Token in the Token box

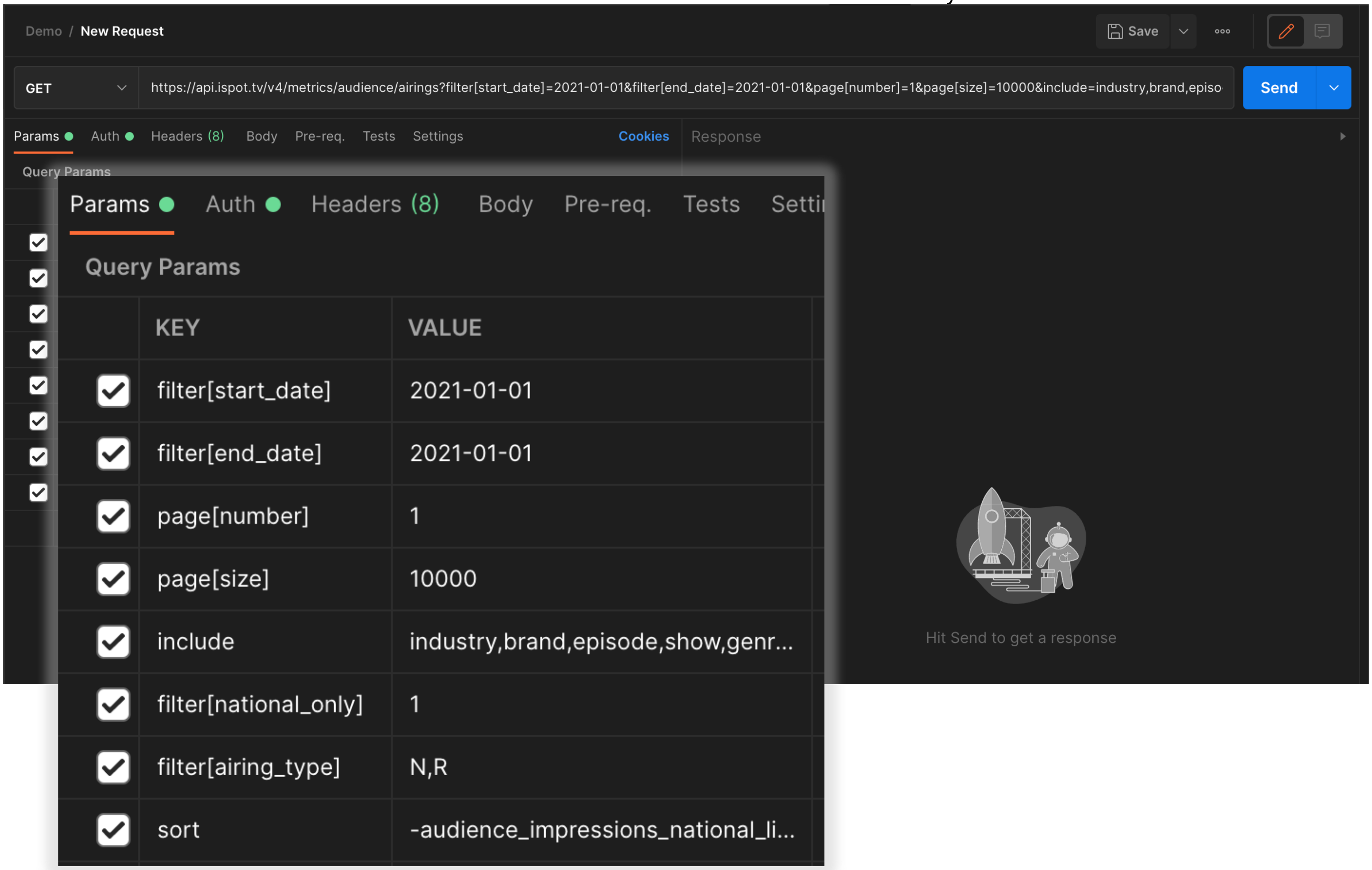


Type: Bearer Token

The authorization header will be automatically generated when you send the request.

Token: eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI' ...

6. Click on Params to enter filters and limits shown below for the data you would like to retrieve



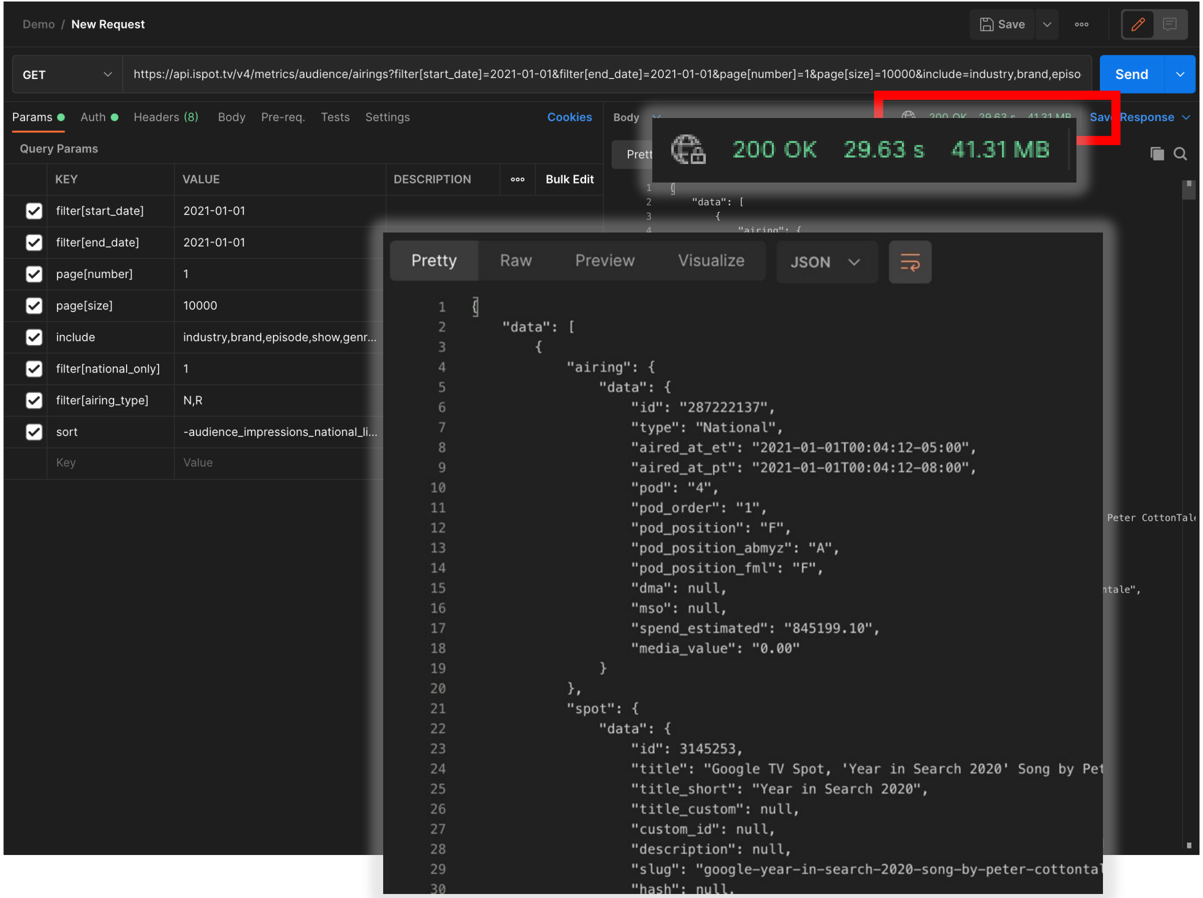
Demo / New Request
Save
Send
Params ● Auth ● Headers (8) Body Pre-req. Tests Settings
Cookies
Response
Hit Send to get a response

GET https://api.ispot.tv/v4/metrics/audience/airings?filter[start_date]=2021-01-01&filter[end_date]=2021-01-01&page[number]=1&page[size]=10000&include=industry,brand,episo

Query Params

	KEY	VALUE
☑	filter[start_date]	2021-01-01
☑	filter[end_date]	2021-01-01
☑	page[number]	1
☑	page[size]	10000
☑	include	industry,brand,episode,show,genr...
☑	filter[national_only]	1
☑	filter[airing_type]	N,R
☑	sort	-audience_impressions_national_li...

7. Click the Send button, and you should see your data appear at the bottom in JSON format



Demo / New Request
Save
Send
Params ● Auth ● Headers (8) Body Pre-req. Tests Settings
Cookies
Body
Save Response
Pretty Raw Preview Visualize JSON

GET https://api.ispot.tv/v4/metrics/audience/airings?filter[start_date]=2021-01-01&filter[end_date]=2021-01-01&page[number]=1&page[size]=10000&include=industry,brand,episo

Query Params

	KEY	VALUE	DESCRIPTION
☑	filter[start_date]	2021-01-01	
☑	filter[end_date]	2021-01-01	
☑	page[number]	1	
☑	page[size]	10000	
☑	include	industry,brand,episode,show,genr...	
☑	filter[national_only]	1	
☑	filter[airing_type]	N,R	
☑	sort	-audience_impressions_national_li...	
	Key	Value	

200 OK 29.63 s 41.31 MB

```
{
    "data": [
        {
            "airing": {
                "data": {
                    "id": "287222137",
                    "type": "National",
                    "aired_at_et": "2021-01-01T00:04:12-05:00",
                    "aired_at_pt": "2021-01-01T00:04:12-08:00",
                    "pod": "4",
                    "pod_order": "1",
                    "pod_position": "F",
                    "pod_position_abmyz": "A",
                    "pod_position_fml": "F",
                    "dma": null,
                    "mso": null,
                    "spend_estimated": "845199.10",
                    "media_value": "0.00"
                }
            },
            "spot": {
                "data": {
                    "id": 3145253,
                    "title": "Google TV Spot, 'Year in Search 2020' Song by Pet
                    "title_short": "Year in Search 2020",
                    "title_custom": null,
                    "custom_id": null,
                    "description": null,
                    "slug": "google-year-in-search-2020-song-by-peter-cottonta
                    "hash": null,
```