Backend Supporting documentation

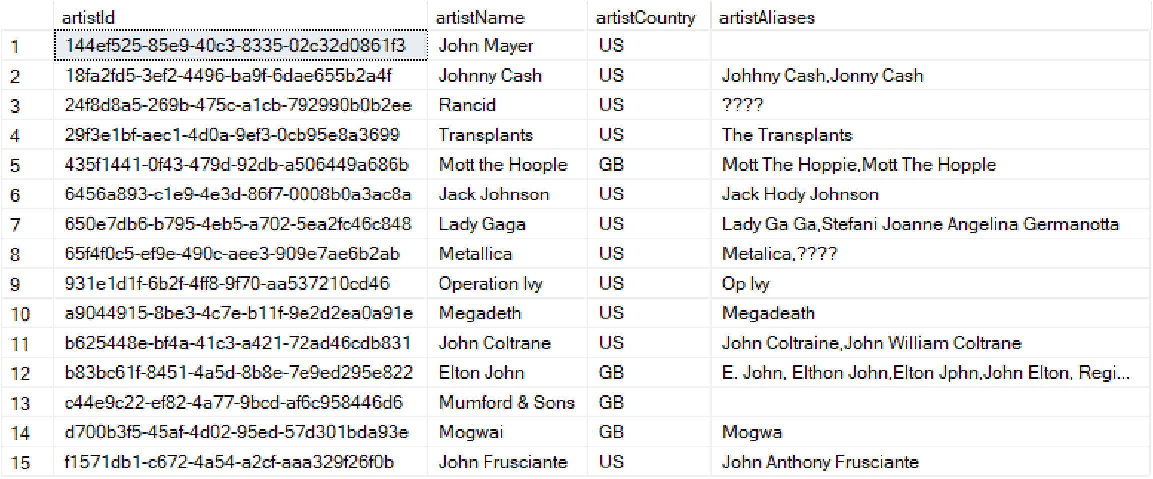
Required software*:*

- Sql server 2017 (v 15.0.2000.5)
- Sql Server Management Studio v.17 or latest
- Visual studio 2019 community
- .Net Framework 4.7.2

STEP 1: Database creation

1 Execute **ArtistAPI.sql** by copy and paste the contents in a new query on SSMS.

2 Once the script is executed, we should have on our table **dbo.artist** the following values:

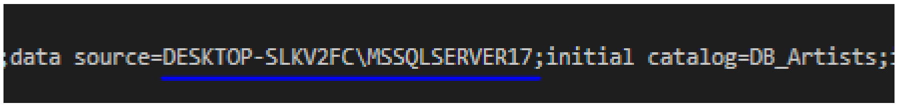


	artistId	artistName	artistCountry	artistAliases
1	144ef525-85e9-40c3-8335-02c32d0861f3	John Mayer	US	
2	18fa2fd5-3ef2-4496-ba9f-6dae655b2a4f	Johnny Cash	US	Johhny Cash,Jonny Cash
3	24f8d8a5-269b-475c-a1cb-792990b0b2ee	Rancid	US	????
4	29f3e1bf-aec1-4d0a-9ef3-0cb95e8a3699	Transplants	US	The Transplants
5	435f1441-0f43-479d-92db-a506449a686b	Mott the Hoople	GB	Mott The Hoppie,Mott The Hopple
6	6456a893-c1e9-4e3d-86f7-0008b0a3ac8a	Jack Johnson	US	Jack Hody Johnson
7	650e7db6-b795-4eb5-a702-5ea2fc46c848	Lady Gaga	US	Lady Ga Ga,Stefani Joanne Angelina Germanotta
8	65f4f0c5-ef9e-490c-aee3-909e7ae6b2ab	Metallica	US	Metalica,????
9	931e1d1f-6b2f-4ff8-9f70-aa537210cd46	Operation Ivy	US	Op Ivy
10	a9044915-8be3-4c7e-b11f-9e2d2ea0a91e	Megadeth	US	Megadeath
11	b625448e-bf4a-41c3-a421-72ad46cdb831	John Coltrane	US	John Coltraine,John William Coltrane
12	b83bc61f-8451-4a5d-8b8e-7e9ed295e822	Elton John	GB	E. John, Elthon John,Elton Jphn,John Elton, Regi...
13	c44e9c22-ef82-4a77-9bcd-af6c958446d6	Mumford & Sons	GB	
14	d700b3f5-45af-4d02-95ed-57d301bda93e	Mogwai	GB	Mogwa
15	f1571db1-c672-4a54-a2cf-aaa329f26f0b	John Frusciante	US	John Anthony Frusciante

STEP 2: Change ConnectionString

1 Open **ArtistAPI.sln** on Visual studio.

2 Go to the Web.config. Change the **data source** of the **connectionString**. Replace the underlined value in screenshot to your server instance.



```
;data source=DESKTOP-SLKV2FC\MSSQLSERVER17;initial catalog=DB_Artists;
```

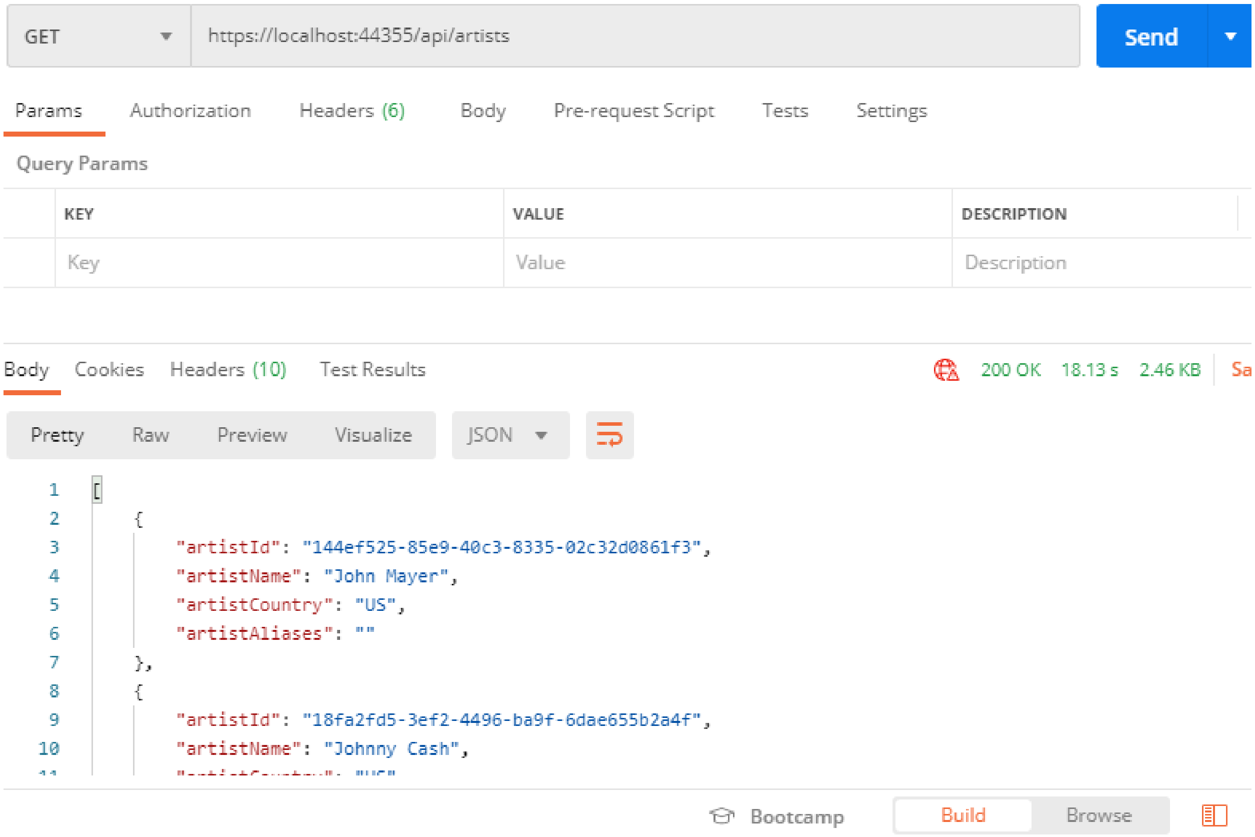
3 Right click the solution and restore nugget packages

4 Build solution and run the project

5 When the home page of *https://localhost:44355/* is correctly open, we can start testing on POSTMAN.

List of collections:

1- Query all artist in the database:
https://localhost:44355/api/aritsts

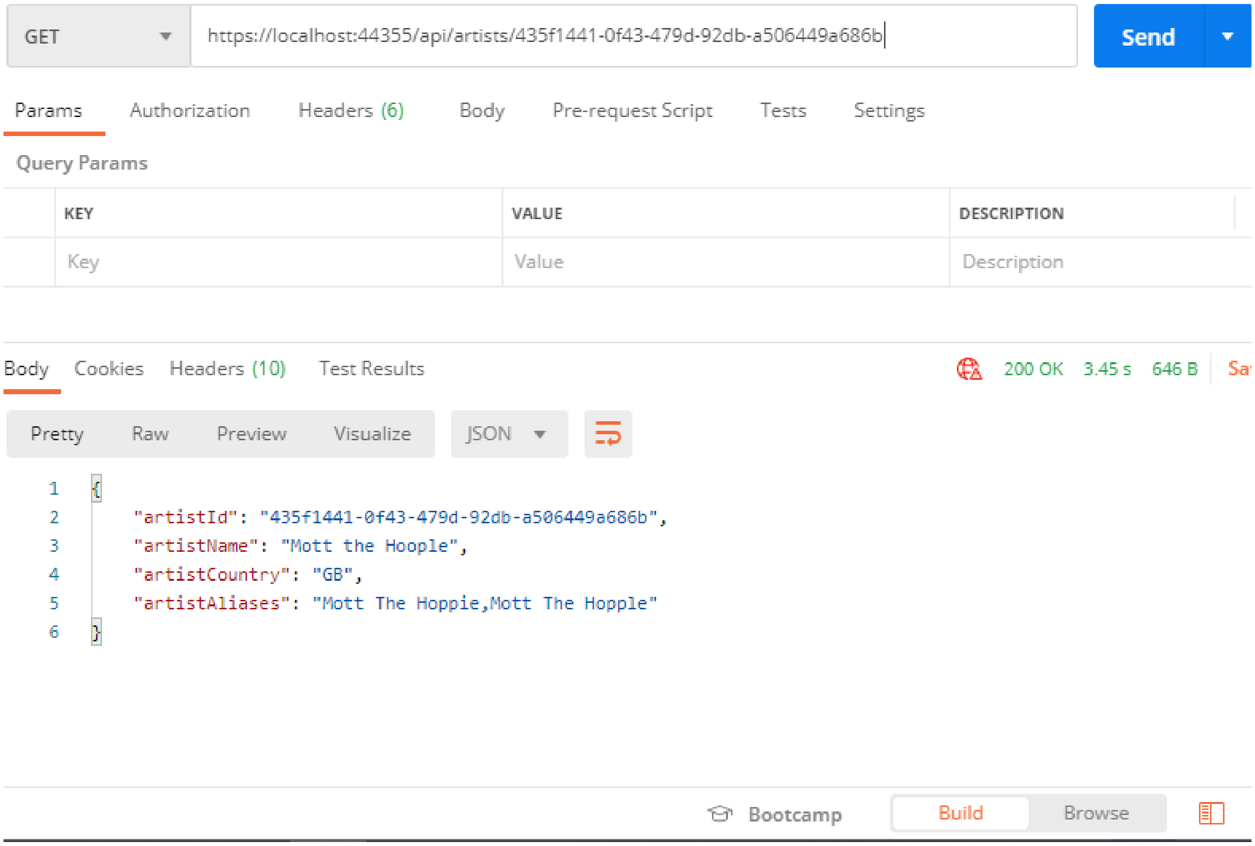


GET https://localhost:44355/api/artists

200 OK 18.13 s 2.46 KB

```
[
    {
        "artistId": "144ef525-85e9-40c3-8335-02c32d0861f3",
        "artistName": "John Mayer",
        "artistCountry": "US",
        "artistAliases": ""
    },
    {
        "artistId": "18fa2fd5-3ef2-4496-ba9f-6dae655b2a4f",
        "artistName": "Johnny Cash",
```

2- Query an artist by a specific Id:
https://localhost:44355/api/aritsts/<artistId>



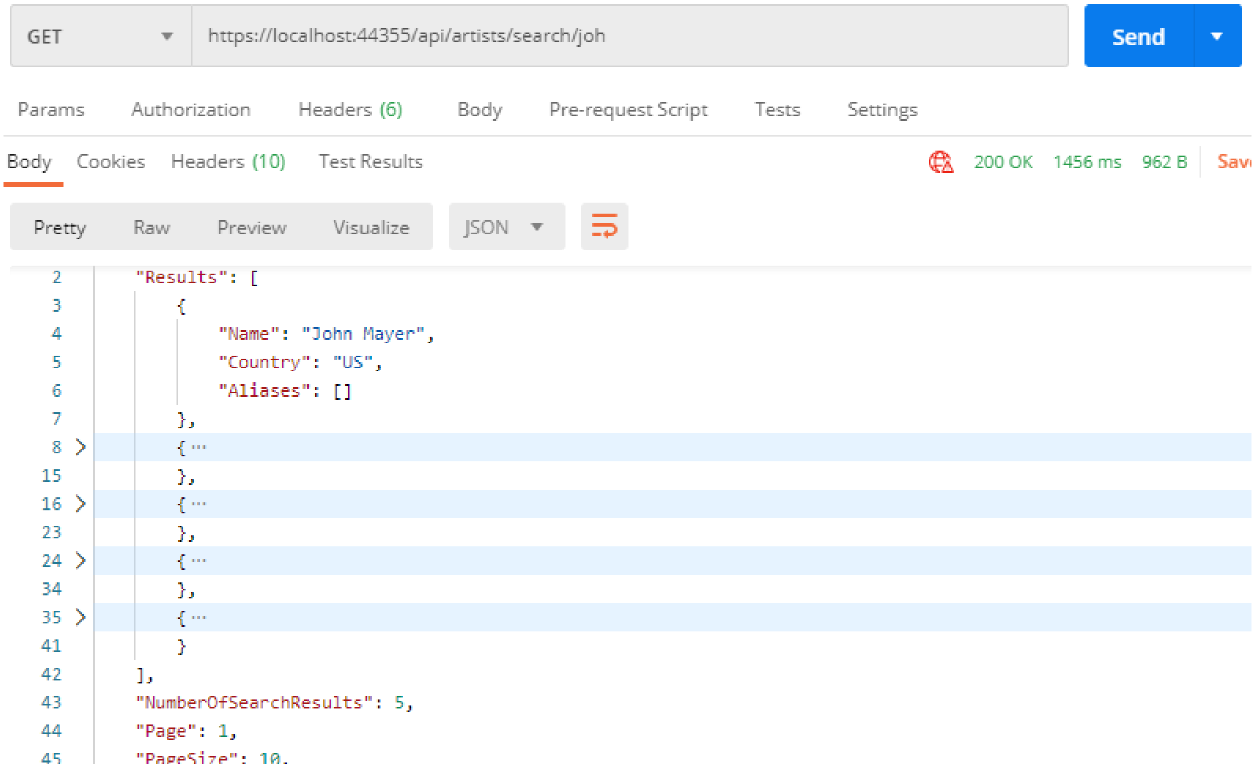
```
GET https://localhost:44355/api/artists/435f1441-0f43-479d-92db-a506449a686b
```

200 OK 3.45 s 646 B

```
{
    "artistId": "435f1441-0f43-479d-92db-a506449a686b",
    "artistName": "Mott the Hoople",
    "artistCountry": "GB",
    "artistAliases": "Mott The Hoppie,Mott The Hopple"
}
```

3- Query artists that have a name or one of their aliases starting with <criteria>
https://localhost:44366/api/artist/search/<criteria>/<pageNumber>/<pageSize>

PS: <pageNumber> and <pageSize> are optional and have a default value 1 and 10 respectively.



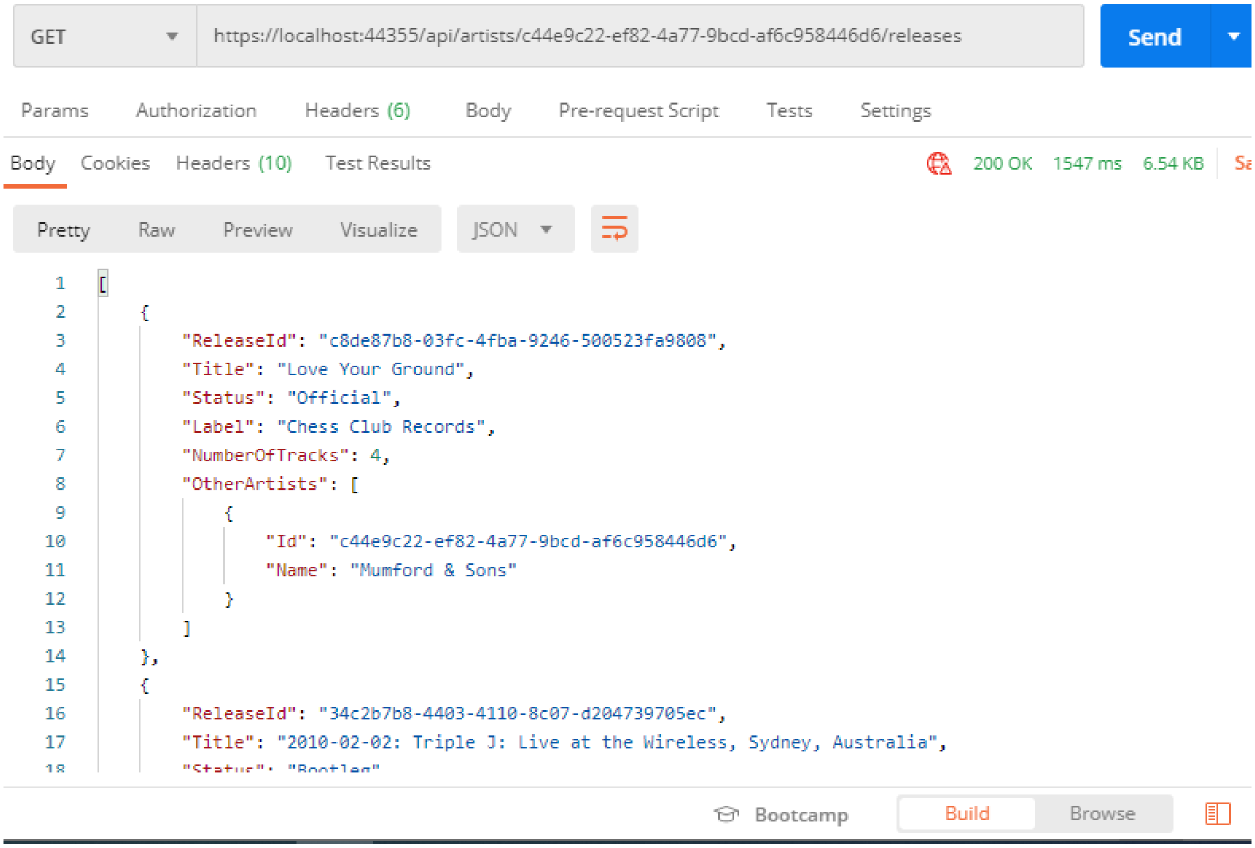
https://localhost:44355/api/artists/search/joh

200 OK 1456 ms 962 B

```
    "Results": [
        {
            "Name": "John Mayer",
            "Country": "US",
            "Aliases": []
        },
        {⋯
        },
        {⋯
        },
        {⋯
        },
        {⋯
        }
    ],
    "NumberOfSearchResults": 5,
    "Page": 1,
    "PageSize": 10,
```

4- Search releases by artistId
https://localhost:44366/api/artist/<artist_id>/releases

PS: Return **all the releases** for a selected artist where the **artist_id uniquely identifies an artist in the SQL Server database and MusicBrainz API.** So, it will return an empty array if one of the conditions is not fulfilled.



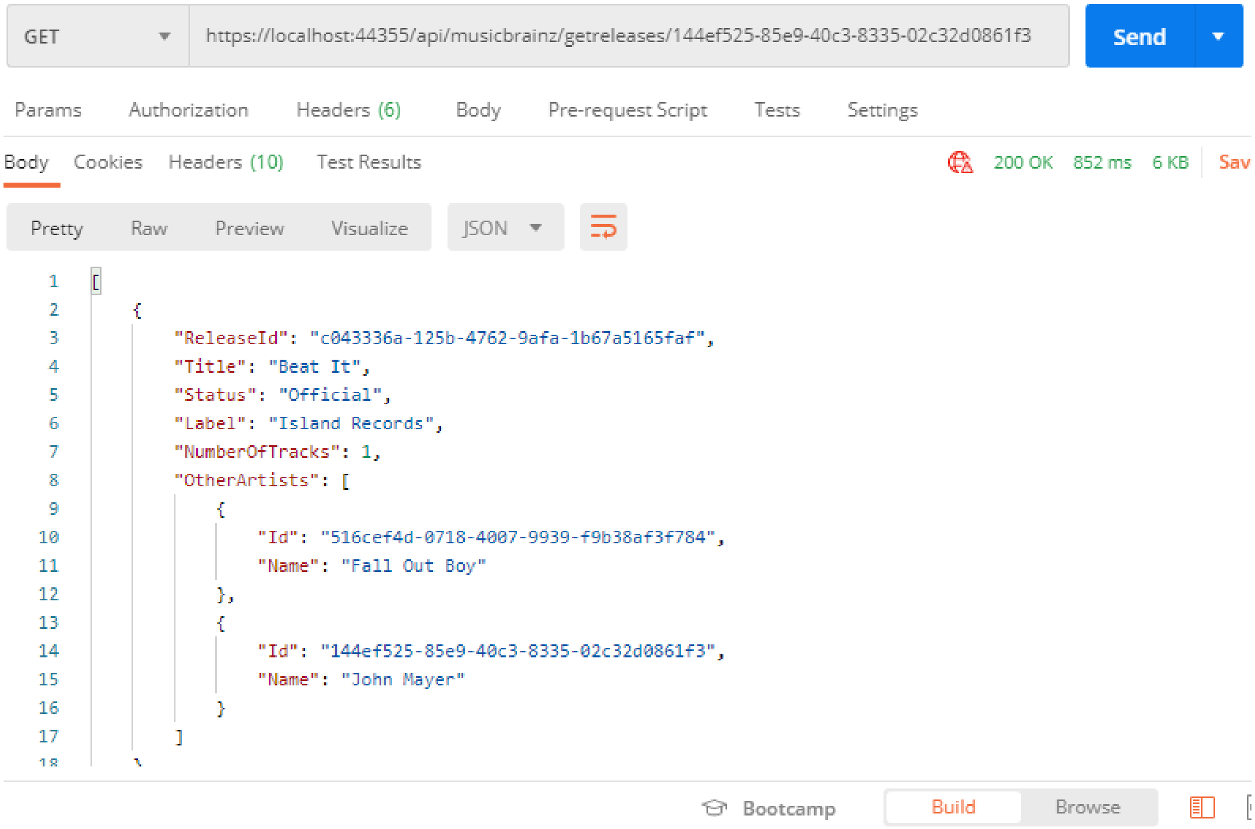
GET https://localhost:44355/api/artists/c44e9c22-ef82-4a77-9bcd-af6c958446d6/releases

200 OK 1547 ms 6.54 KB

```
[
    {
        "ReleaseId": "c8de87b8-03fc-4fba-9246-500523fa9808",
        "Title": "Love Your Ground",
        "Status": "Official",
        "Label": "Chess Club Records",
        "NumberOfTracks": 4,
        "OtherArtists": [
            {
                "Id": "c44e9c22-ef82-4a77-9bcd-af6c958446d6",
                "Name": "Mumford & Sons"
            }
        ]
    },
    {
        "ReleaseId": "34c2b7b8-4403-4110-8c07-d204739705ec",
        "Title": "2010-02-02: Triple J: Live at the Wireless, Sydney, Australia",
        "Status": "Bootleg"
```

5- The search releases above consume MusicBrainz API.
For testing the appropriate endpoint, we can query the following link:
https://localhost:44355/api/musicbrainz/getreleases/<artistId>

The artistId used in the below picture: 144ef525-85e9-40c3-8335-02c32d0861f3



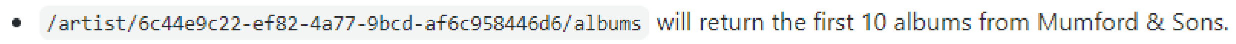
https://localhost:44355/api/musicbrainz/getreleases/144ef525-85e9-40c3-8335-02c32d0861f3

```
[
    {
        "ReleaseId": "c043336a-125b-4762-9afa-1b67a5165faf",
        "Title": "Beat It",
        "Status": "Official",
        "Label": "Island Records",
        "NumberOfTracks": 1,
        "OtherArtists": [
            {
                "Id": "516cef4d-0718-4007-9939-f9b38af3f784",
                "Name": "Fall Out Boy"
            },
            {
                "Id": "144ef525-85e9-40c3-8335-02c32d0861f3",
                "Name": "John Mayer"
            }
        ]
```

The data we got has been retrieved to the raw data from this link http://musicbrainz.org/ws/2/release/?query=arid:144ef525-85e9-40c3-8335-02c32d0861f3&fmt=json

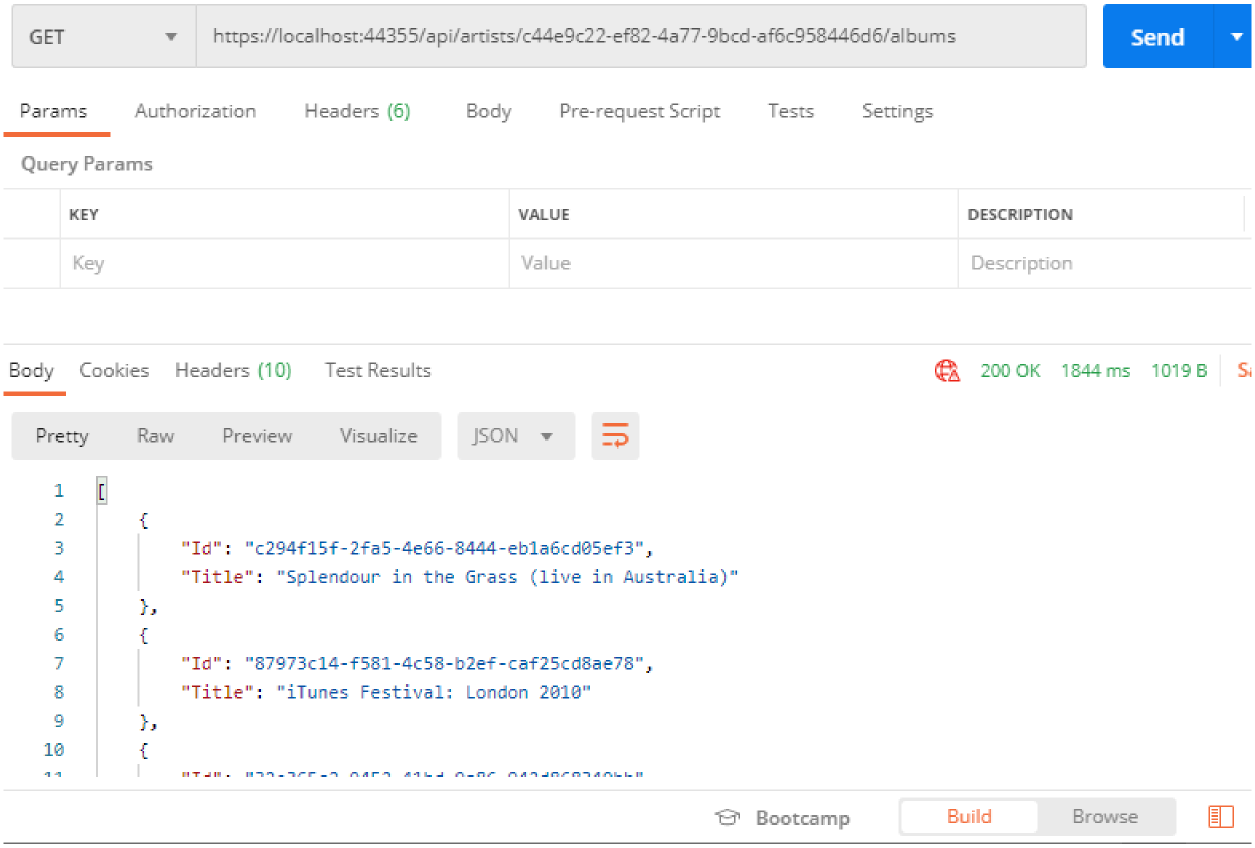
6- Query 10 first albums by artistId
https://localhost:44355/api/artists/<artistId> /albums
used artist id in the below picture: c44e9c22-ef82-4a77-9bcd-af6c958446d6

PS:
1 – The artistId provided in the assessment is wrong



- `/artist/6c44e9c22-ef82-4a77-9bcd-af6c958446d6/albums` will return the first 10 albums from Mumford & Sons.

It must begin with c but not with 6.



GET https://localhost:44355/api/artists/c44e9c22-ef82-4a77-9bcd-af6c958446d6/albums

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

200 OK 1844 ms 1019 B

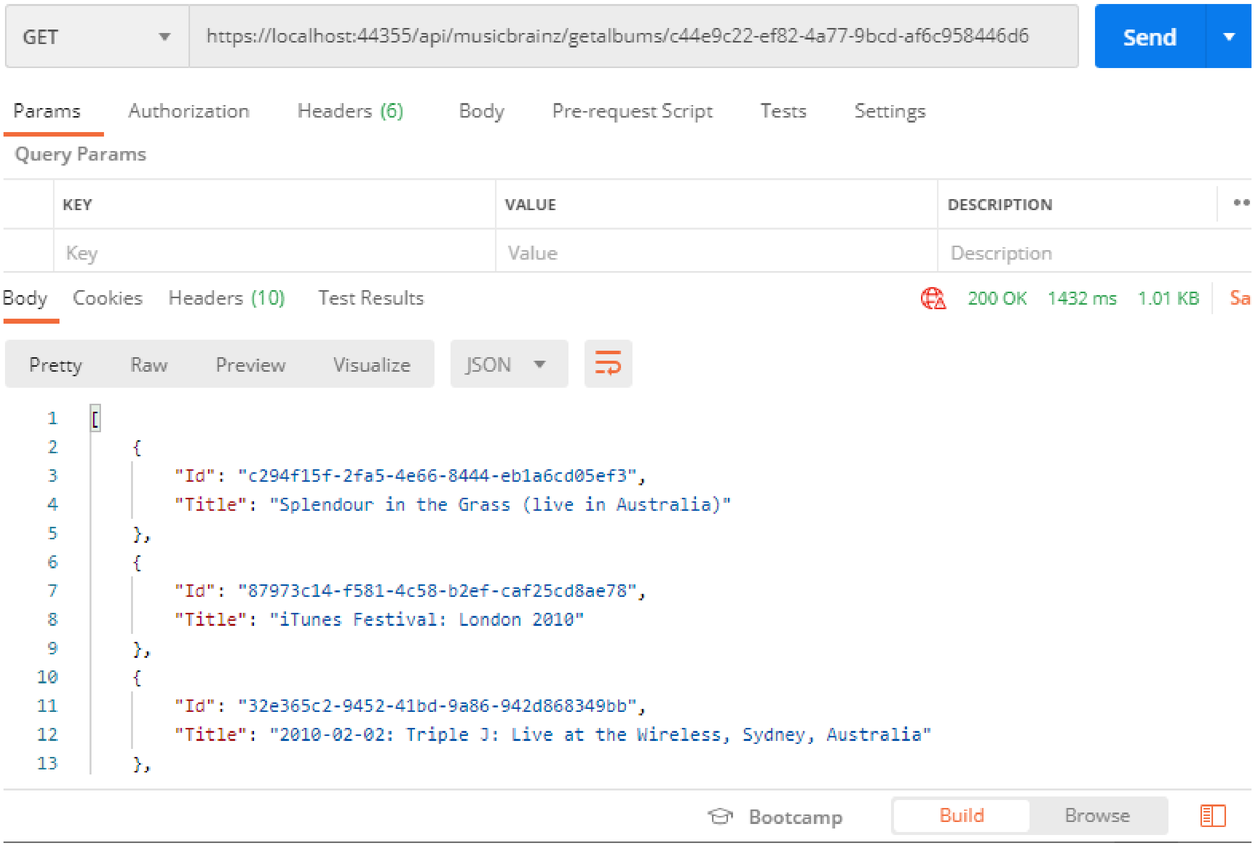
```
[
    {
        "Id": "c294f15f-2fa5-4e66-8444-eb1a6cd05ef3",
        "Title": "Splendour in the Grass (live in Australia)"
    },
    {
        "Id": "87973c14-f581-4c58-b2ef-caf25cd8ae78",
        "Title": "iTunes Festival: London 2010"
    },
    {
```

2 – The album is selected in function of the "primary-type" attribute of each item in release-group list.
By the way – We only have 7 albums when using the key "c44e9c22-ef82-4a77-9bcd-af6c958446d6"

The raw value of the above query can be checked from:
https://musicbrainz.org/ws/2/release-group/?query=arid:83b9cbe7-9857-49e2-ab8e-b57b01038103&fmt=json

7- The sixth endpoint is consuming MusicBrainzApi that can be checked from this link: https://localhost:44355/api/musicbrainz/getalbums/<artistId>

The used artistId in the below and above picture: c44e9c22-ef82-4a77-9bcd-af6c958446d6



GET https://localhost:44355/api/musicbrainz/getalbums/c44e9c22-ef82-4a77-9bcd-af6c958446d6

200 OK 1432 ms 1.01 KB

```
[
    {
        "Id": "c294f15f-2fa5-4e66-8444-eb1a6cd05ef3",
        "Title": "Splendour in the Grass (live in Australia)"
    },
    {
        "Id": "87973c14-f581-4c58-b2ef-caf25cd8ae78",
        "Title": "iTunes Festival: London 2010"
    },
    {
        "Id": "32e365c2-9452-41bd-9a86-942d868349bb",
        "Title": "2010-02-02: Triple J: Live at the Wireless, Sydney, Australia"
    },
```