DOCUMENTATION

IP2C-Web-API

Made with .NET 7 & EF Core

George Zalokostas

Overview

This is a Web-API project made in .NET Core (v. 7.0.102) with EF Core, using the Repository Pattern & Unit of Work. It has been built using the Database-First approach.

The program exposes and endpoint that gets some information for a given IP. The data are fetched from the https://ip2c.org website and are stored locally in a Database, and in an in-memory cache. The user has to be authorized to access the endpoints. This is done with a POST request in the /login and /register endpoints.

When a request is made for IP details, the system first checks if the requested IP is already cached. If the IP is found in the cache, its data is returned directly from the cache without querying the database or the ip2c website.

If the requested IP is not cached, the system checks the database to see if the IP data is already stored there. If the data is found in the database, it is retrieved and stored in the cache for future use, and then returned to the caller.

If the data is not found in the cache or the database, the system queries the ip2c website to obtain the data. Once the data is obtained, it is stored in the database and in the cache for future use, and then returned to the caller.

In summary, the system first checks the cache, then the database, and finally the ip2c website to obtain the requested IP data. The data is then stored in the cache and the database for future use.

The program exposes a second endpoint as well, which returns a report for the data in the database, grouped by the Country, and a count of how many data we have

Finally, there is a repeating job that runs every one hour, which compares each IP from the database, with the results from ip2c, and updates the database and cache in case of something has changed.

Endpoints

Authorization:

In order for the user to access any endpoints, he has to be authorized. This is done by sending a POST request to either /Auth/Register or /Auth/Login with his credentials. The program creates a HASH for the salt, and for the stored password using HMACSHA512.

When a new user is created, the hashed password is stored in the database, along with the salt.

When a user tries to login with a password, the program validates the password by using the stored salt, and the plain password provided. If a user is signed in , the program responds with a Bearer token, that he/she has to provide each time he/she tries to access an endpoint.

[POST] /Auth/Register with:

```
{
  "username": "test",
  "password": "test"
}
```

[POST] /Auth/Login with:

```
{
  "username": "test",
  "password": "test"
}
```

If a login succeeds, we can see a response like this:

200 OK:


```
{
   "data": "eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJuYW1laWQiOiIyIiwidW5pcXVlX25hbWUiOiJ0ZXN0IiwibmJmIjoxNjewN",
   "success": true,
```

```
  "message": ""}
```


If a login fails, the user can see a response like this:

400 Bad Request:


```
{
  "data": null,
  "success": false,
  "message": "Wrong password."
}
```


Inner endpoints:

[GET] /api/GetIPDetails/1.3.3.3

The user can find the details of an IP by sending a GET request to the GetIPDetails endpoint, providing an IP. The program checks if the current IP is valid or not, and returns either a 404 Not Found, or a 200 OK with the data below:


```
{
  "data": {
    "countryName": "China",
    "twoLetterCode": "CN",
    "threeLetterCode": "CHN"
  },
  "success": true,
  "message": ""
}
```


[GET] /api/GetReport/
[GET] /api/GetReport/GR
[GET] /api/GetReport/GR,CN,US

The user can get a report from the database's data, by sending a GET request to the GetReport endpoint. The report consists of how many IPs we have stored in the database, grouped by each Country.

Optionally, the user can provide some two letter codes in order to get the data for specific countries.

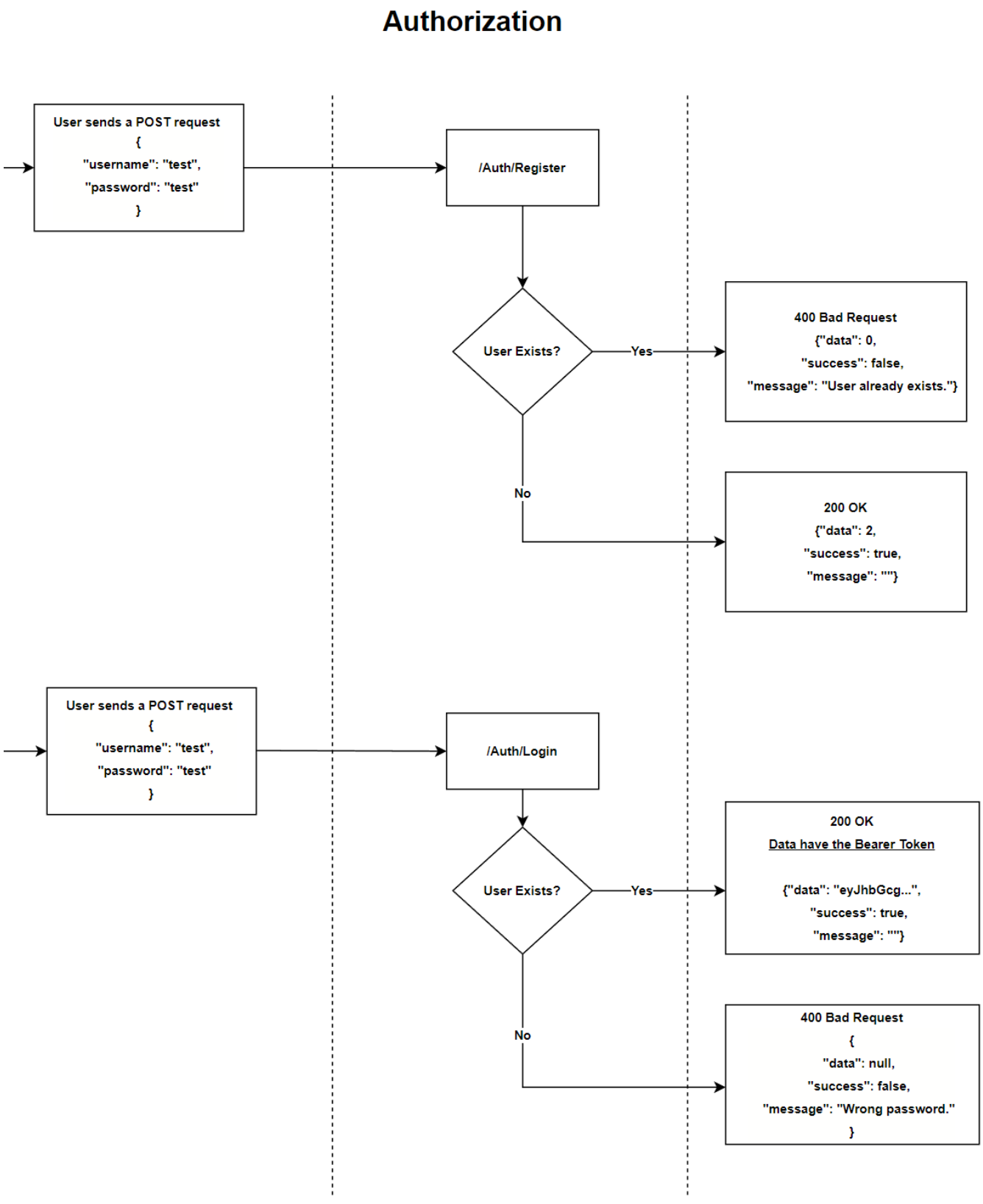
For example a /GetReport/GR can return either:


```
{
  "data": [],
  "success": false,
  "message": "No data found for the provided country codes."
}
```

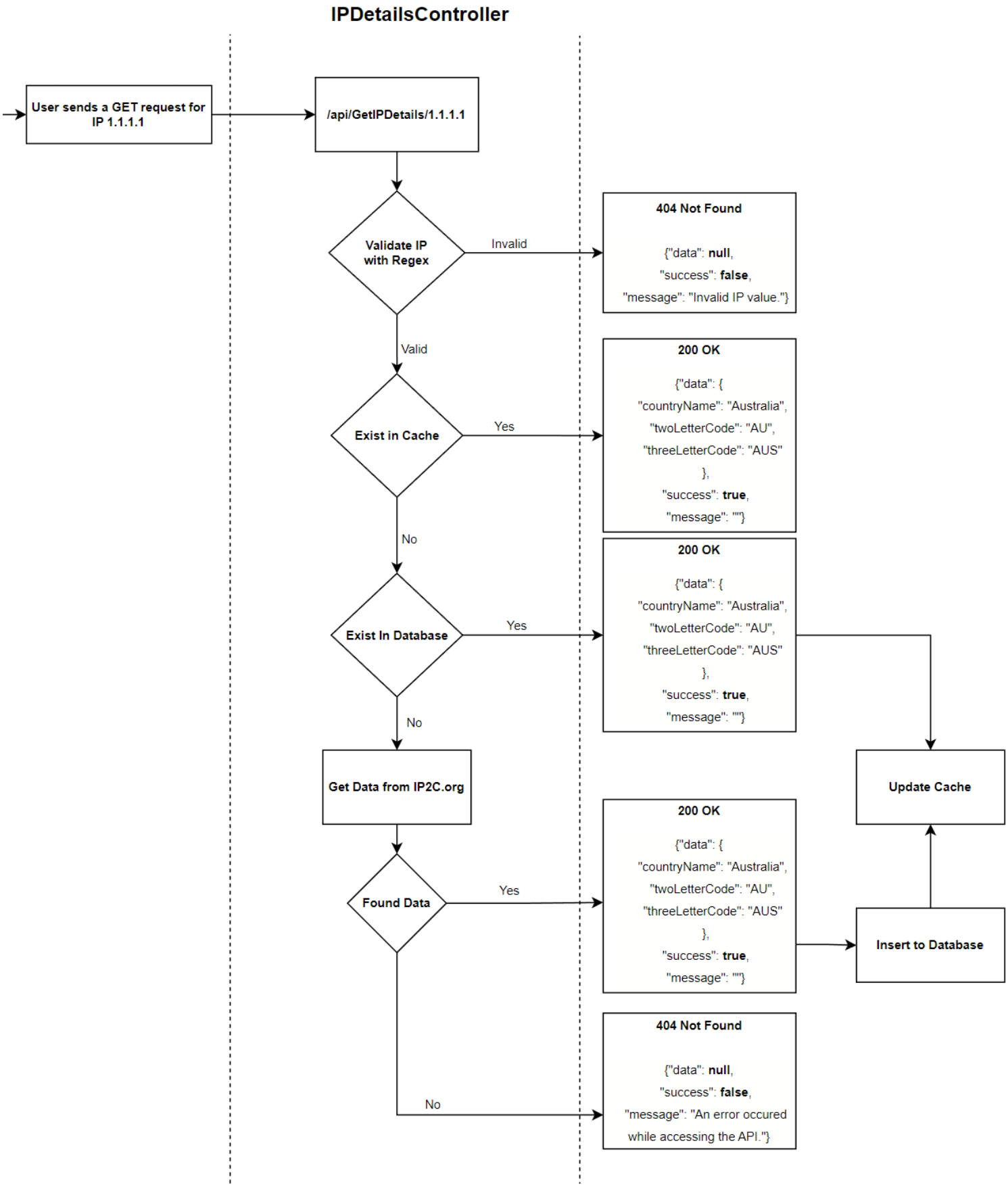

or


```
{
  "data": [
    {
      "countryName": "China",
      "addressesCount": 2,
      "lastAddressUpdated": "2023-04-02T14:53:37.8737424"
    }
  ],
  "success": true,
  "message": ""
}
```

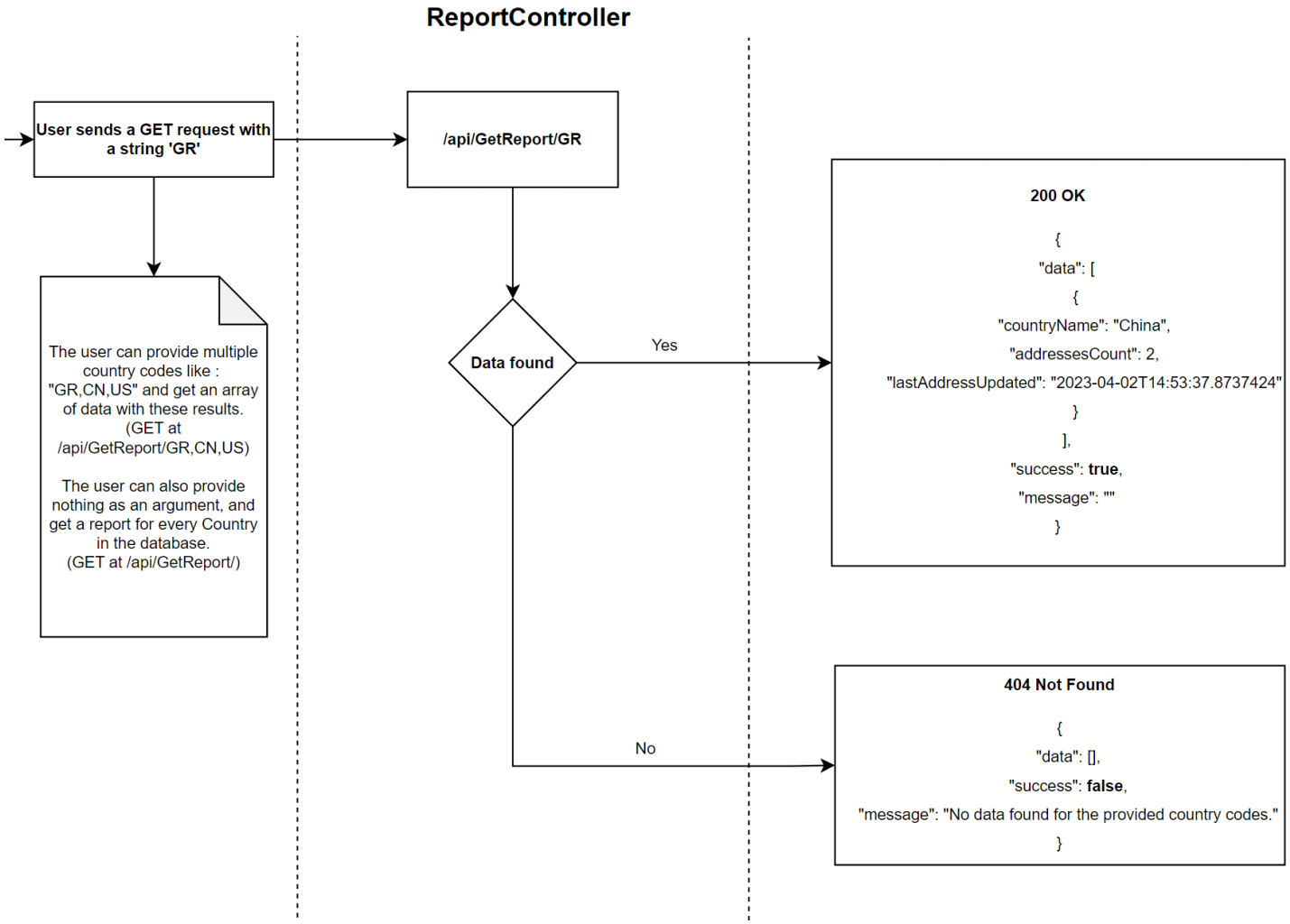
Flowcharts


Authorization
User sends a POST request
{
"username": "test",
"password": "test"
}
/Auth/Register
User Exists?
Yes
400 Bad Request
{"data": 0,
"success": false,
"message": "User already exists."}
No
200 OK
{"data": 2,
"success": true,
"message": ""}
User sends a POST request
{
"username": "test",
"password": "test"
}
/Auth/Login
User Exists?
Yes
200 OK
Data have the Bearer Token
{"data": "eyJhbGcg...",
"success": true,
"message": ""}
No
400 Bad Request
{
"data": null,
"success": false,
"message": "Wrong password."
}

/GetIPDetails/{id}



/GetReport/{strings?}



Background Sync

There is a job running every hour , that fetches all the IPs from the database in batches of 100 and updates the tables whether something has changed.

The batching process is done with pagination. There is an infinite loop that initially fetches the first 100 data, increments a counter , and then skips X times the data it has fetched, until no more entries are found.

Every batch is processed using a Parallel Foreach loop, of which each item is calling the IP2C API, and If a Country , the TwoLetterCode or the ThreeLetterCode has changed, the database is updated.

Finally, the Cache is also updated and the new requests will have the updated data.

Dependencies

The following dependencies have been installed using the .NET CLI

- dotnet tool install --global dotnet-ef
- dotnet add package Microsoft.EntityFrameworkCore.Design
- dotnet add package Microsoft.EntityFrameworkCore.SqlServer
- dotnet add package Microsoft.EntityFrameworkCore.Tools
- dotnet add package Dapper
- dotnet add package RestSharp
- dotnet add package Microsoft.Extensions.Configuration
- dotnet add package Microsoft.Toolkit
- dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer
- dotnet add package Swashbuckle.AspNetCore.Filters

App Settings:

```
"ConnectionStrings": {
    "DefaultConnection": "Data Source=DESKTOP-
    NPJCM17;Database=master;Trusted_Connection=true;TrustServerCertificate=tru
    e;"
  },
```

Scaffold Command:

dotnet ef dbcontext scaffold Name=DefaultConnection Microsoft.EntityFrameworkCore.SqlServer -o Models

Data Seed

```
/****** Object: Table [dbo].[Countries] Script Date: 12/10/2022 12:07:23 ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Countries](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [varchar](50) NOT NULL,

[TwoLetterCode] [char](2) NOT NULL,

[ThreeLetterCode] [char](3) NOT NULL,

[CreatedAt] [datetime2](7) NOT NULL,

CONSTRAINT [PK_Countries] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 95,

OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/****** Object: Table [dbo].[IPAddresses] Script Date: 12/10/2022 12:07:23 ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO
```

```
CREATE TABLE [dbo].[IPAddresses](
[Id] [int] IDENTITY(1,1) NOT NULL,
[CountryId] [int] NOT NULL,
[IP] [varchar](15) NOT NULL,
[CreatedAt] [datetime2](7) NOT NULL,
[UpdatedAt] [datetime2](7) NOT NULL,
CONSTRAINT [PK_IPAddresses] PRIMARY KEY CLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 95,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
CREATE TABLE [dbo].[Users] (
        [Id]         INT           IDENTITY (1, 1) NOT NULL,
        [Username]     NVARCHAR (MAX)  NOT NULL,
        [PasswordHash] VARBINARY (MAX) NOT NULL,
        [PasswordSalt] VARBINARY (MAX) NOT NULL,
        CONSTRAINT [PK_Users] PRIMARY KEY CLUSTERED ([Id] ASC)
)
GO
SET IDENTITY_INSERT [dbo].[Countries] ON
GO
INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (1,
N'Greece', N'GR', N'GRC', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))
GO
```

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (2, N'Germany', N'DE', N'DEU', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (3, N'Cyprus', N'CY', N'CYP', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (4, N'United States', N'US', N'USA', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (6, N'Spain', N'ES', N'ESP', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (7, N'France', N'FR', N'FRA', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (8, N'Italy', N'IT', N'IA ', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (9, N'Japan', N'JP', N'JPN', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

INSERT [dbo].[Countries] ([Id], [Name], [TwoLetterCode], [ThreeLetterCode], [CreatedAt]) VALUES (10, N'China', N'CN', N'CHN', CAST(N'2022-10-12T06:46:10.5000000' AS DateTime2))

GO

SET IDENTITY_INSERT [dbo].[Countries] OFF

GO

SET IDENTITY_INSERT [dbo].[IPAddresses] ON

```
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (6, 1,
N'44.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),
CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (7, 2,
N'45.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),
CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (8, 3,
N'46.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),
CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (9, 4,
N'47.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),
CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (10, 6,
N'49.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),
CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (11, 7,
N'41.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),
CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (12, 8,
N'42.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),
```

CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (13, 9,

N'43.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),

CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (14, 10,

N'50.255.255.254', CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2),

CAST(N'2022-10-12T07:04:06.8566667' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (15, 1,

N'44.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (16, 2,

N'45.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (17, 3,

N'46.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (18, 4,

N'47.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (19, 6,

N'49.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (20, 7,

N'41.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (21, 8,

N'42.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (22, 9,

N'43.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (23, 10,

N'50.25.55.254', CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2),

CAST(N'2022-10-12T07:04:33.3800000' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (24, 1,

N'44.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),

CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))

GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (25, 2,

N'45.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),

CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))

GO

```
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (26, 3,
N'46.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),
CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (27, 4,
N'47.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),
CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (28, 6,
N'49.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),
CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (29, 7,
N'41.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),
CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (30, 8,
N'42.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),
CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (31, 9,
N'43.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),
CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))
GO
INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (32, 10,
N'50.25.55.4', CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2),
CAST(N'2022-10-12T07:04:51.3233333' AS DateTime2))
```

```
GO

INSERT [dbo].[IPAddresses] ([Id], [CountryId], [IP], [CreatedAt], [UpdatedAt]) VALUES (33, 1,

N'10.20.30.40', CAST(N'2022-10-12T08:41:37.3100000' AS DateTime2),

CAST(N'2022-10-12T08:41:37.3100000' AS DateTime2))

GO

SET IDENTITY_INSERT [dbo].[IPAddresses] OFF

GO

SET ANSI_PADDING ON

GO

/****** Object: Index [IX_IPAddresses] Script Date: 12/10/2022 12:07:23 ******/

ALTER TABLE [dbo].[IPAddresses] ADD CONSTRAINT [IX_IPAddresses] UNIQUE NONCLUSTERED

(

[IP] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,

IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,

FILLFACTOR = 95, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Countries] ADD CONSTRAINT [DF_Countries_CreatedAt] DEFAULT (getutcdate())

FOR [CreatedAt]

GO

ALTER TABLE [dbo].[IPAddresses] ADD CONSTRAINT [DF_IPAddresses_CreatedAt] DEFAULT

(getutcdate()) FOR [CreatedAt]

GO

ALTER TABLE [dbo].[IPAddresses] ADD CONSTRAINT [DF_IPAddresses_UpdatedAt] DEFAULT

(getutcdate()) FOR [UpdatedAt]

GO

ALTER TABLE [dbo].[IPAddresses] WITH CHECK ADD CONSTRAINT [FK_IPAddresses_Countries]
```

```
FOREIGN KEY([CountryId])

REFERENCES [dbo].[Countries] ([Id])

GO

ALTER TABLE [dbo].[IPAddresses] CHECK CONSTRAINT [FK_IPAddresses_Countries]

GO
```