

The Vulture Fund API

Author:

Dominic Samo, BSc Computer Science and Physics, Certified Data Scientist and Software Engineer

Portfolio Website

[Dominic Samo](#)

Socials

[LinkedIn](#)

[Medium](#)

Environment Used To Code The Vulture Fund API

- Ubuntu 22.04 LTS
- Visual Studio Code
- .NET Version: 7.0.109
- MySQL Server Version: 8.0.33

API Testing

- Swagger (Autogenerated by .NET)
- Postman

Packages Used:

```
$ dotnet add package Microsoft.EntityFrameworkCore  
  
$ dotnet add package Pomelo.EntityFrameworkCore.MySql --version 7.0.0  
  
$ dotnet add package Microsoft.EntityFrameworkCore.Design  
  
$ dotnet add package Microsoft.EntityFrameworkCore.Tools  
  
$ dotnet add package Pomelo.EntityFrameworkCore.MySql  
  
$ dotnet add package Microsoft.AspNetCore.Mvc.Core
```

Useful Commands:

```
$ dotnet tool install --global dotnet-ef  
  
$ dotnet tool update --global dotnet-ef
```

```
$ dotnet tool restore

$ dotnet new tool-manifest

$ dotnet ef migrations add InitialCreate

$ dotnet ef database update
```

dotnet-tools.json

```
{
  "version": 1,
  "isRoot": true,
  "tools": {
    "dotnet-ef": {
      "version": "7.0.9",
      "commands": [
        "dotnet-ef"
      ]
    }
  }
}
```

NB:

All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server can be connected to the API, the only few lines that will have to change will be in the **Program.cs** file. A suitable Entity Framework Core (EF Core) provider would also have to be installed.

I used Pomelo.EntityFrameworkCore.MySql which is the Entity Framework Core (EF Core) provider for MySQL, MariaDB, Amazon Aurora, Azure Database for MySQL and other MySQL-compatible databases. It is built on top of MySqlConnectionConnector.

Got Pomelo.EntityFrameworkCore.MySql from [here](#).

Code

- C# was used.
- Allowed Cross-Origin Resource Sharing (CORS) for a test React Application running on Port: 3000:

```
var MyAllowedOrigins = "_myCORS";

builder.Services.AddCors(options =>{
    options.AddPolicy(name: MyAllowedOrigins,
        policy =>
        {
```

```
        policy.WithOrigins("http://localhost:3000")
        .AllowAnyHeader()
        .AllowAnyMethod()
        .AllowCredentials();
    });
});
```

Get the Contact details of the current executives

URL: `/api/Contacts`

Method: `GET`

Auth required: NO

Permissions required: None

Success Response

Code: `200 OK`

Content examples

Response body:

```
[
  {
    "id": "265c1835-1829-4aa3-a990-a89b591f7c8d",
    "fullName": "Maria Elda Samo dos Santos",
    "position": "Chief Legal Officer",
    "email": "maria.samo@tvf.co.sz",
    "phone": 8522143285,
    "address": "3 Tech Bros Bv, Capital Park Township"
  },
  {
    "id": "36c2b892-84cf-4a93-9030-dffc3e7e8909",
    "fullName": "Daniel K. Gadlela",
    "position": "Chief Information Officer",
    "email": "daniel.gadlela@tvf.co.sz",
    "phone": 4578227472,
    "address": "10 Jakalazi St, Westwing"
  },
  {
    "id": "46e55616-f104-454c-9630-5fa0e88d5ac9",
    "fullName": "Nomalanga F. Masilela",
    "position": "Chief Data Scientist",
    "email": "noma.masilela@tvf.co.sz",
    "phone": 4578222422,
    "address": "22 Jakalazi St, Westwing"
  }
]
```

Get the Contact details of a particular executive

URL : `/api/Contacts/id`

Method : `GET`

Auth required : NO

Permissions required : None

Success Response

Code : `200 OK`

Content examples

URL : `/api/Contacts/46e55616-f104-454c-9630-5fa0e88d5ac9`

Response body:

```
{
  "id": "46e55616-f104-454c-9630-5fa0e88d5ac9",
  "fullName": "Nomalanga F. Masilela",
  "position": "Chief Data Scientist",
  "email": "noma.masilela@tvf.co.sz",
  "phone": 4578222422,
  "address": "22 Jakalazi St, Westwing"
}
```

Add details of an executive

URL : `/api/Contacts`

Method : `POST`

Auth required : NO

Permissions required : None

Success Response

Code : `200 OK`

Content examples

Request body:

```
{
  "fullName": "Gugulethu M.S. Khumalo-Magongo",
  "position": "Chief Investment Officer",
}
```

```
{
  "email": "gugu.khumalo@tvf.co.sz",
  "phone": 8522143210,
  "address": "9 Indlovu Av, Capital Park Township"
}
```

Response body:

```
{
  "id": "c1b61d70-1a8a-4865-86b8-a0a742549f0f",
  "fullName": "Gugulethu M.S. Khumalo-Magongo",
  "position": "Chief Investment Officer",
  "email": "gugu.khumalo@tvf.co.sz",
  "phone": 8522143210,
  "address": "9 Indlovu Av, Capital Park Township"
}
```

Edit details of an executive

URL : `/api/Contacts`

Method : `PUT`

Auth required : NO

Permissions required : None

Success Response

Code : `200 OK`

Content examples

URL : `/api/Contacts/ecd81573-a00a-49c9-bac4-84ca63fdf0fc`

Edit from:

```
{
  "fullName": "Elton Pateguana Jr",
  "position": "Chief Financial Officer",
  "email": "elton.pateguana@tvf.co.sz",
  "phone": 4578227412,
  "address": "18 Freeway St, Westwing"
}
```

To:

Request body:

```
{
  "fullName": "Elton M. Pateguana Jr, CFA",
  "position": "Chief Financial Officer",
  "email": "elton.pateguana@tvf.co.sz",
  "phone": 4578227412,
  "address": "18 Freeway St, Westwing"
}
```

Response body:

```
{
  "id": "ecd81573-a00a-49c9-bac4-84ca63fdf0fc",
  "fullName": "Elton M. Pateguana Jr, CFA",
  "position": "Chief Financial Officer",
  "email": "elton.pateguana@tvf.co.sz",
  "phone": 4578227412,
  "address": "18 Freeway St, Westwing"
}
```

Delete details of an executive

URL : `/api/Contacts`

Method : `DELETE`

Auth required : NO

Permissions required : None

Success Response

Code : `200 OK`

Content examples

URL : `/api/Contacts/ecd81573-a00a-49c9-bac4-84ca63fdf0fc`

Response body:

```
{
  "id": "ecd81573-a00a-49c9-bac4-84ca63fdf0fc",
  "fullName": "Elton M. Pateguana Jr, CFA",
  "position": "Chief Financial Officer",
  "email": "elton.pateguana@tvf.co.sz",
  "phone": 4578227412,
  "address": "18 Freeway St, Westwing"
}
```

Notes

- Most modern web applications expose APIs that clients can use to interact with the application. A well-designed web API should aim to support:
 - Platform independence. Any client should be able to call the API, regardless of how the API is implemented internally. This requires using standard protocols, and having a mechanism whereby the client and the web service can agree on the format of the data to exchange.
 - Service evolution. The web API should be able to evolve and add functionality independently from client applications. As the API evolves, existing client applications should continue to function without modification. All functionality should be discoverable so that client applications can fully use it.
- [Read More about RESTful web API design](#)