Use guide to the .NET Core 8 Web API application for CRUD operations with User and Account classes (Banking Management System)

This guide explains how to set up and operate the .NET Core 8 Web API application for managing User and Account entities.

How to Run the Software

1. Pre-requisites:

- Install .NET SDK 8.0 or higher.
- Install SQL Server Management Studio (SSMS) for managing the database.
- Install an IDE such as Visual Studio 2022.

2. Set Up SQL Server:

- Launch SQL Server Management Studio and create a new database named, for example, BankingManagementDB.
- Copy the connection string for your database from SSMS. It should look like this:

```
Server=YOUR_SERVER_NAME;Database=BankingManagementDB;Trusted_Connection=True;MultipleActiveResultSets=true
```

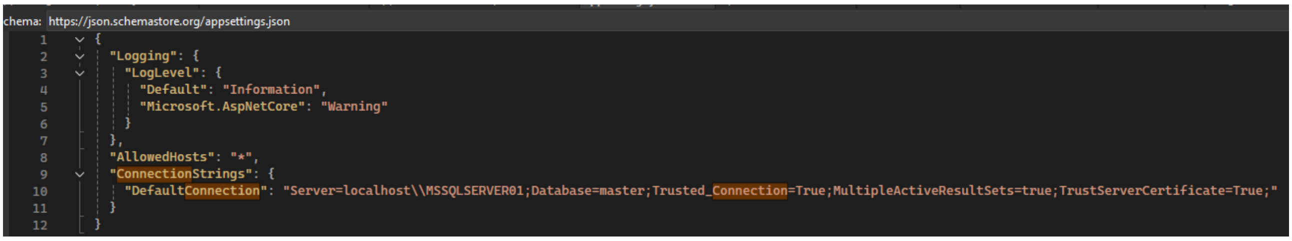
3. Configure the Application:

- Open the project in Visual Studio.
- Navigate to the appsettings.json file in the root directory.
- Update the ConnectionStrings section with your connection string:

```
"ConnectionStrings": {

  "DefaultConnection": "Server=YOUR_SERVER_NAME;Database=BankingManagementDB;Trusted_Connection=True;MultipleActiveResultSets=true"

}
```



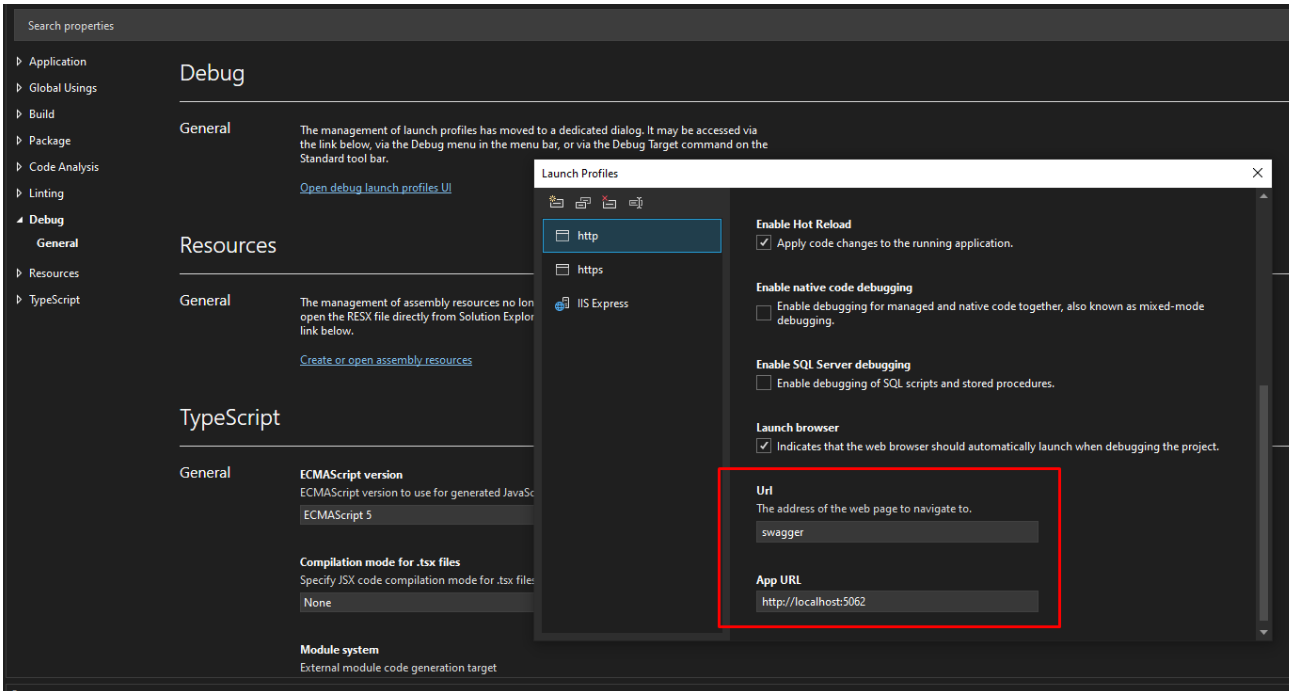
chema: https://json.schemastore.org/appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=localhost\\MSSQLSERVER01;Database=master;Trusted_Connection=True;MultipleActiveResultSets=true;TrustServerCertificate=True;"
  }
}
```

4. Set the Application Port:

In Visual Studio, go to Project Properties:

- Right-click the project name in the Solution Explorer and choose Properties.
- Under Debug, set the port (e.g., 5000 or 7000).



Debug

General

The management of launch profiles has moved to a dedicated dialog. It may be accessed via the link below, via the Debug menu in the menu bar, or via the Debug Target command on the Standard tool bar.

Open debug launch profiles UI

Resources

General

The management of assembly resources no lon open the RESX file directly from Solution Explor link below.

Create or open assembly resources

TypeScript

General

ECMAScript version
ECMAScript version to use for generated JavaSc
ECMAScript 5

Compilation mode for .tsx files
Specify JSX code compilation mode for .tsx files
None

Module system
External module code generation target

Launch Profiles

Enable Hot Reload
☑ Apply code changes to the running application.

Enable native code debugging
□ Enable debugging for managed and native code together, also known as mixed-mode debugging.

Enable SQL Server debugging
□ Enable debugging of SQL scripts and stored procedures.

Launch browser
☑ Indicates that the web browser should automatically launch when debugging the project.

Url
The address of the web page to navigate to.
swagger

App URL
http://localhost:5062

5. Build the Application:

In Visual Studio, open the Terminal or use the top menu bar:

- Run dotnet build to compile the application.
- Apply the database migrations using the following commands:

```
dotnet ef migrations add InitialCreate
```

```
dotnet ef database update
```

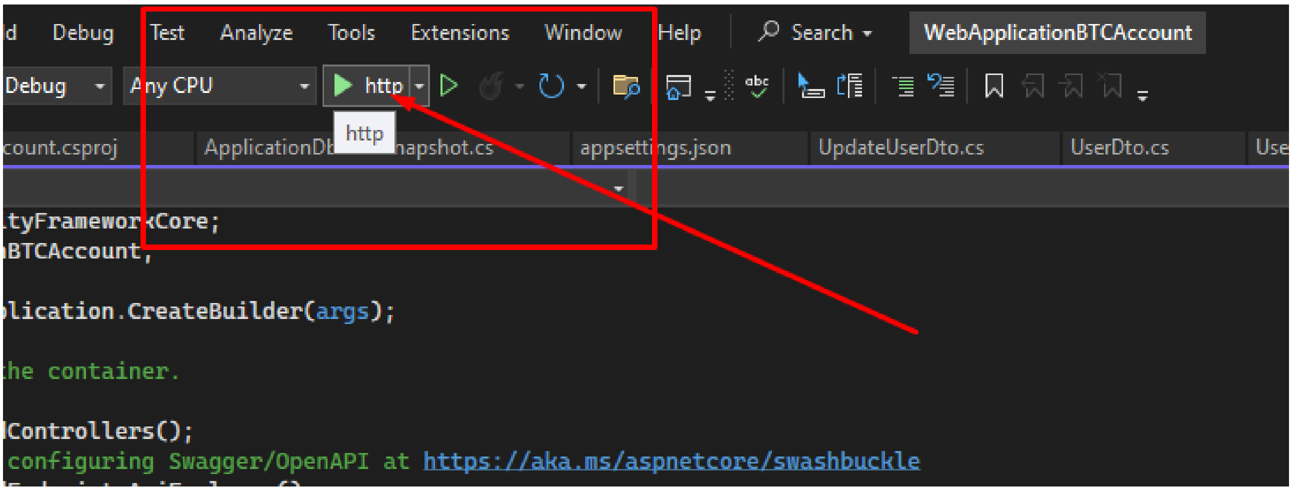
6. Run the Server:

- Press F5 or run the command:

```
dotnet run
```

The server will start, and the Swagger UI will be accessible at https://localhost:{port}/swagger

- OR press the green triangle button:



```
tyFrameworkCore;
BTCAccount;

lication.CreateBuilder(args);

he container.

Controllers();
configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
```

Operating Instructions

1. Using *Swagger Interface*:

If the Swagger didn't open by default, open a web browser and navigate to https://localhost:{port}/swagger.

Use the Swagger UI to interact with the API:

- ✓ Add a User: Use the POST /api/User/AddUser endpoint.
- ✓ Retrieve Users: Use GET /api/User/GetAllUsers or GET /api/User/GetUserById/{id}.
- ✓ Update a User: Use PUT /api/User/UpdateUser.
- ✓ Delete a User: Use DELETE /api/User/DeleteUser/{id}.

2. **Using Command Line (Console):**

Use tools like *curl* or *Postman* to make API requests. Example commands:

Add a User:

```
curl -X POST https://localhost:{port}/api/User/AddUser -H "Content-Type: application/json" -d '{"firstName":"John","lastName":"Doe","email":"john.doe@example.com"}'
```

Get All Users:

```
curl -X GET https://localhost:{port}/api/User/GetAllUsers
```

3. Database Interaction:

Open SSMS to directly view or modify the User and Account tables.