Student Management System – User Guide

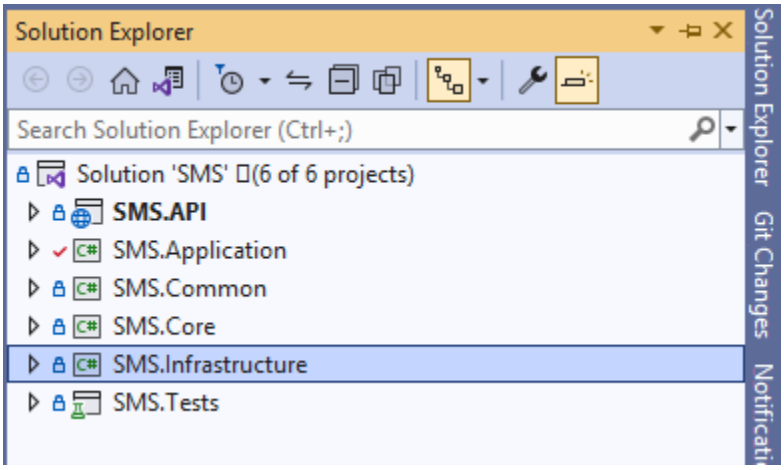
Backend: Dot Net Core – Web API – Entity Framework Code First Approach – Clear Architecture

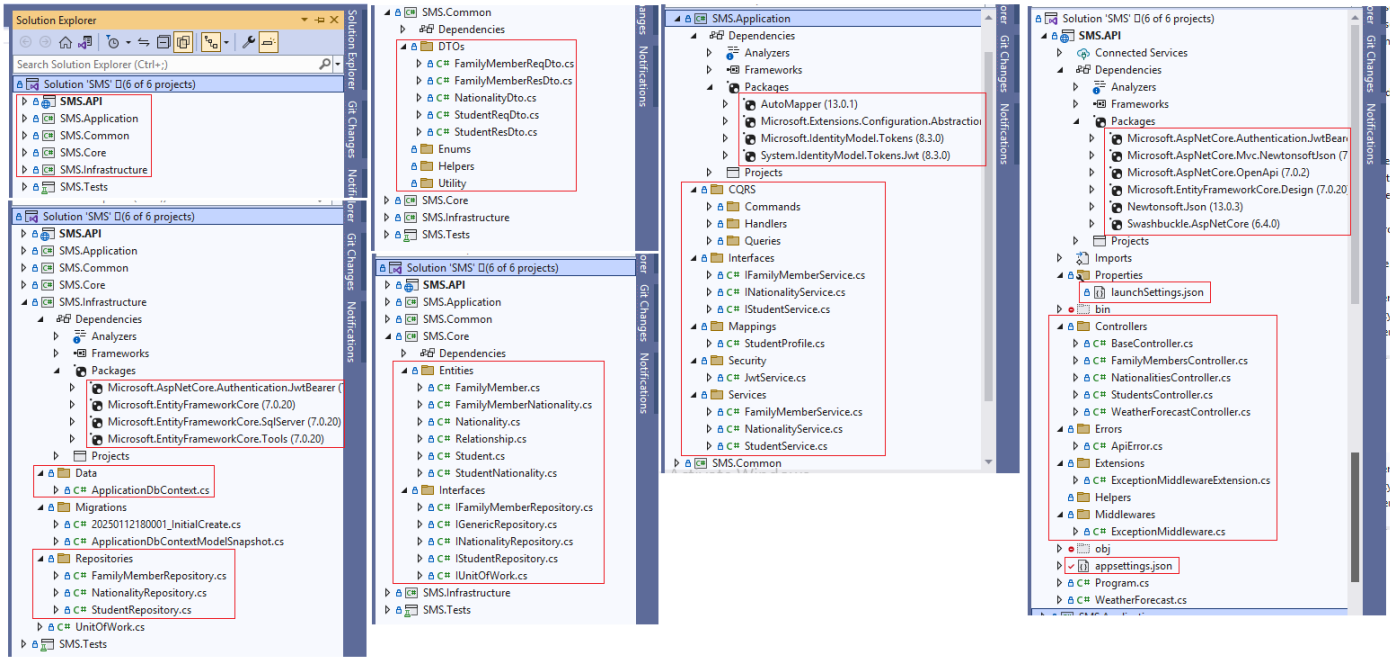
Projects – Core, Infrastructure, Application, Common, API, Tests

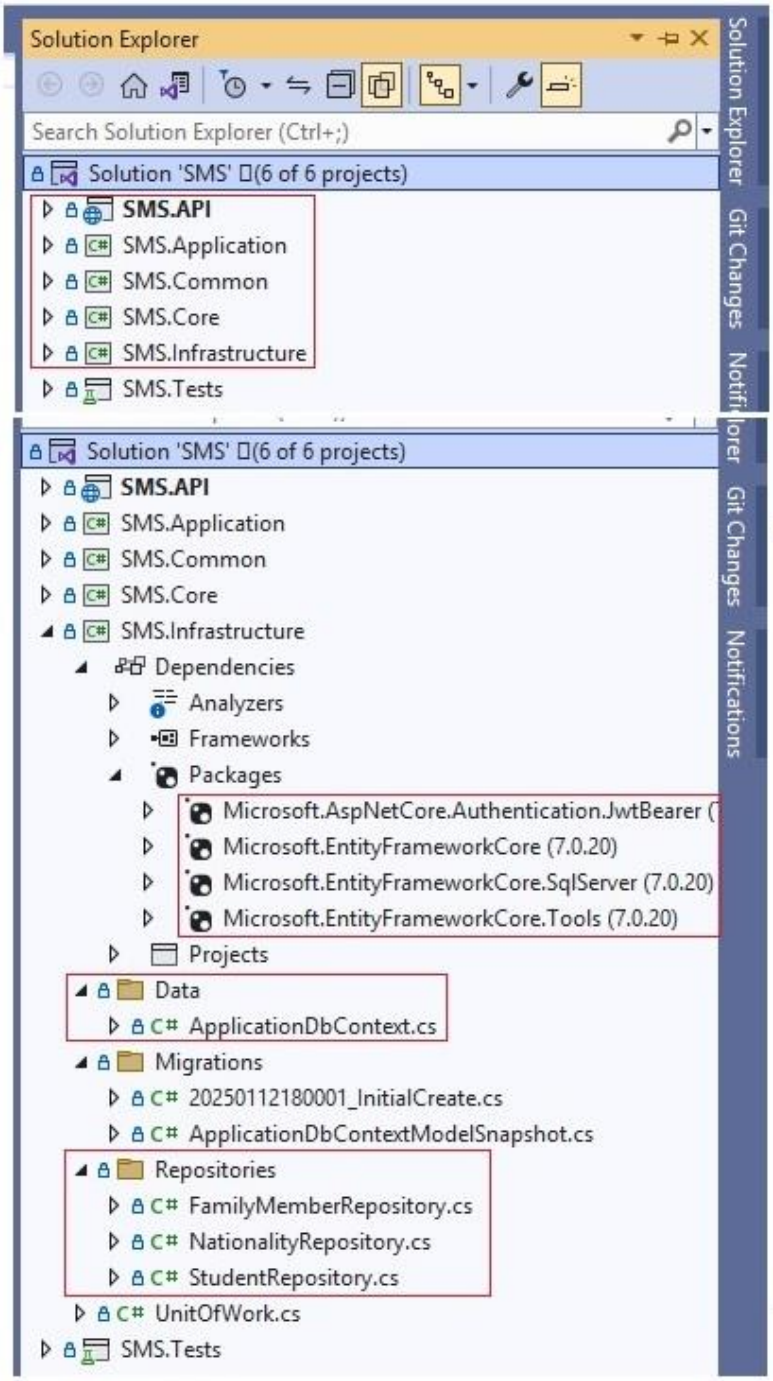
GitHub -.Net Core Backend Link: https://github.com/SheikGH/SMS_Asp.NetCore_WebAPI_Backend.git

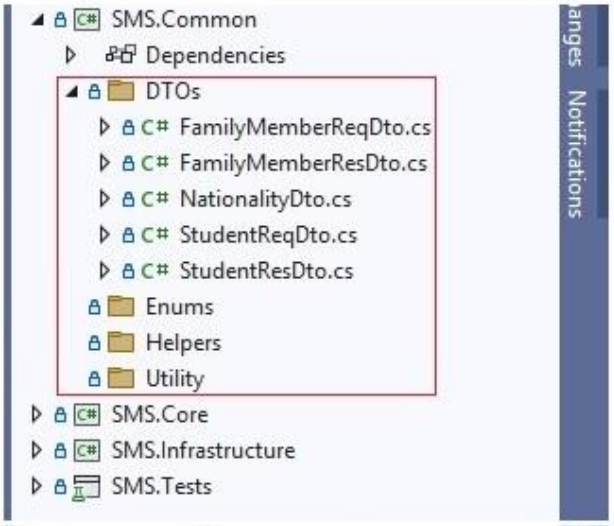
Site Url : **http://localhost:5071/swagger/index.html**

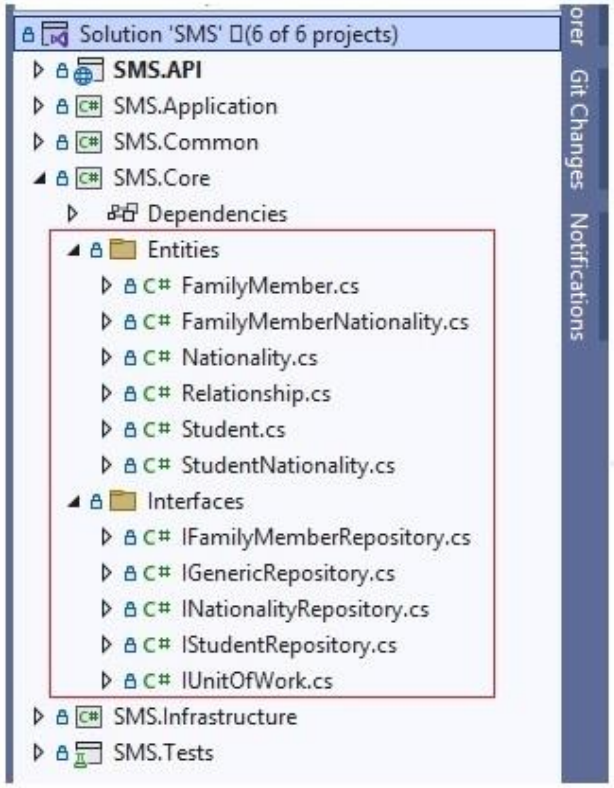
Folder Structure: Clear Architecture - Core, Infrastructure, Application, Common, API, Tests

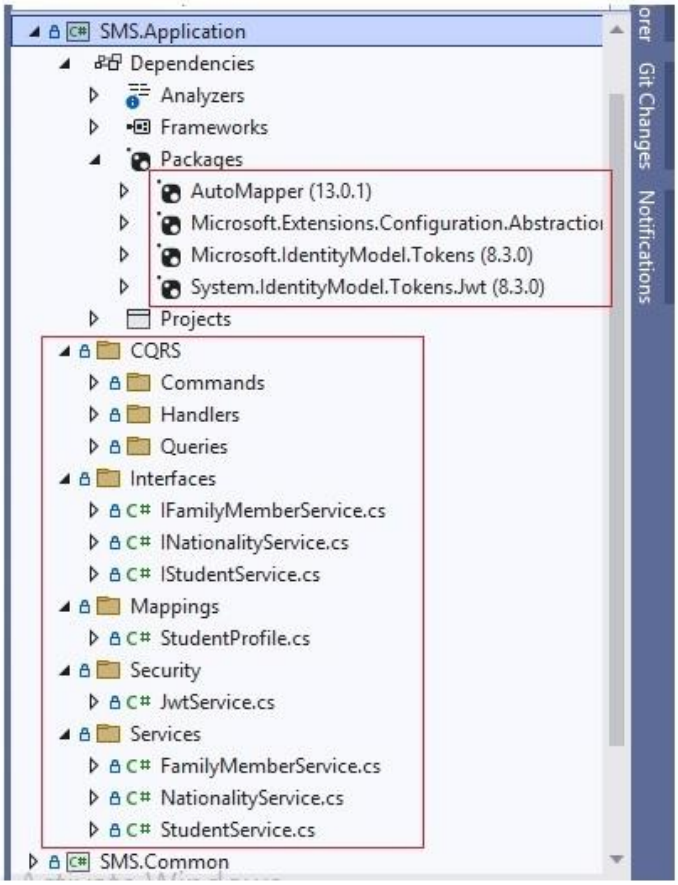


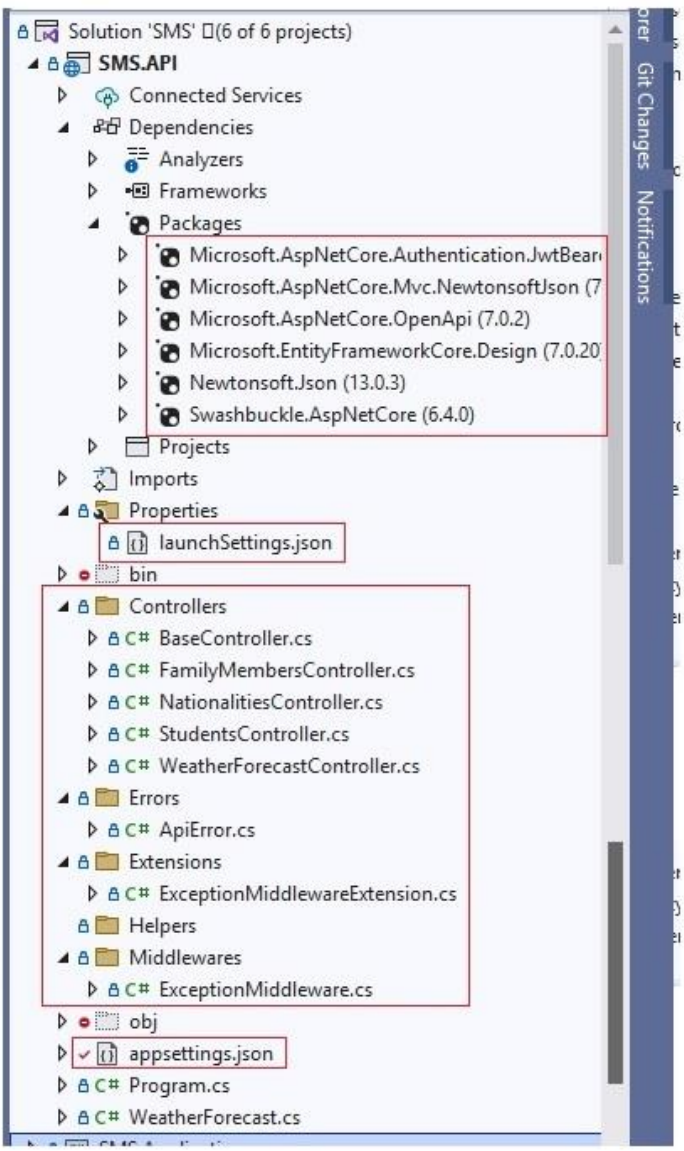

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SMS' (6 of 6 projects)
SMS.API
SMS.Application
SMS.Common
SMS.Core
SMS.Infrastructure
SMS.Tests
Dependencies
Analyzers
Frameworks
Packages
Microsoft.AspNetCore.Authentication.JwtBearer
Microsoft.EntityFrameworkCore (7.0.20)
Microsoft.EntityFrameworkCore.SqlServer (7.0.20)
Microsoft.EntityFrameworkCore.Tools (7.0.20)
Projects
Data
ApplicationDbContext.cs
Migrations
20250112180001_InitialCreate.cs
ApplicationDbContextModelSnapshot.cs
Repositories
FamilyMemberRepository.cs
NationalityRepository.cs
StudentRepository.cs
UnitOfWork.cs
DTOs
FamilyMemberReqDto.cs
FamilyMemberResDto.cs
NationalityDto.cs
StudentReqDto.cs
StudentResDto.cs
Enums
Helpers
Utility
Entities
FamilyMember.cs
FamilyMemberNationality.cs
Nationality.cs
Relationship.cs
Student.cs
StudentNationality.cs
Interfaces
IFamilyMemberRepository.cs
IGenericRepository.cs
INationalityRepository.cs
IStudentRepository.cs
IUnitOfWork.cs
AutoMapper (13.0.1)
Microsoft.Extensions.Configuration.Abstractio
Microsoft.IdentityModel.Tokens (8.3.0)
System.IdentityModel.Tokens.Jwt (8.3.0)
CQRS
Commands
Handlers
Queries
IFamilyMemberService.cs
INationalityService.cs
IStudentService.cs
Mappings
StudentProfile.cs
Security
JwtService.cs
Services
FamilyMemberService.cs
NationalityService.cs
StudentService.cs
Connected Services
Microsoft.AspNetCore.Authentication.JwtBear
Microsoft.AspNetCore.Mvc.NewtonsoftJson (7
Microsoft.AspNetCore.OpenApi (7.0.2)
Microsoft.EntityFrameworkCore.Design (7.0.20
Newtonsoft.Json (13.0.3)
Swashbuckle.AspNetCore (6.4.0)
Imports
Properties
launchSettings.json
bin
Controllers
BaseController.cs
FamilyMembersController.cs
NationalitiesController.cs
StudentsController.cs
WeatherForecastController.cs
Errors
ApiError.cs
Extensions
ExceptionMiddlewareExtension.cs
Helpers
Middlewares
ExceptionMiddleware.cs
obj
appsettings.json
Program.cs
WeatherForecast.cs


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SMS' (6 of 6 projects)
SMS.API
SMS.Application
SMS.Common
SMS.Core
SMS.Infrastructure
SMS.Tests
Dependencies
Analyzers
Frameworks
Packages
Microsoft.AspNetCore.Authentication.JwtBearer (
Microsoft.EntityFrameworkCore (7.0.20)
Microsoft.EntityFrameworkCore.SqlServer (7.0.20)
Microsoft.EntityFrameworkCore.Tools (7.0.20)
Projects
Data
ApplicationDbContext.cs
Migrations
20250112180001_InitialCreate.cs
ApplicationDbContextModelSnapshot.cs
Repositories
FamilyMemberRepository.cs
NationalityRepository.cs
StudentRepository.cs
UnitOfWork.cs
Solution Explorer
Git Changes
Notifications


SMS.Common
Dependencies
DTOs
FamilyMemberReqDto.cs
FamilyMemberResDto.cs
NationalityDto.cs
StudentReqDto.cs
StudentResDto.cs
Enums
Helpers
Utility
SMS.Core
SMS.Infrastructure
SMS.Tests
Notifications


Solution 'SMS' (6 of 6 projects)
SMS.API
SMS.Application
SMS.Common
SMS.Core
Dependencies
Entities
FamilyMember.cs
FamilyMemberNationality.cs
Nationality.cs
Relationship.cs
Student.cs
StudentNationality.cs
Interfaces
IFamilyMemberRepository.cs
IGenericRepository.cs
INationalityRepository.cs
IStudentRepository.cs
IUnitOfWork.cs
SMS.Infrastructure
SMS.Tests
Git Changes
Notifications





✅ Prerequisites

Ensure the following are installed on your machine:

- .NET SDK 6.0 or later
- SQL Server
- Visual Studio 2022+ (or Visual Studio Code + C# Extensions)
- Git
- EF Core CLI Tools (included in .NET SDK)

🔁 1. Clone the Backend Repository

```
git clone https://github.com/SheikGH/SMS_Asp.NetCore_WebAPI_Backend.git
```

Navigate into the project folder:

```
cd SMS_Asp.NetCore_WebAPI_Backend
```

⚙️ 2. Open the Project

Open the solution in **Visual Studio** (`SMS.sln`) or open the folder in **VS Code**.

---

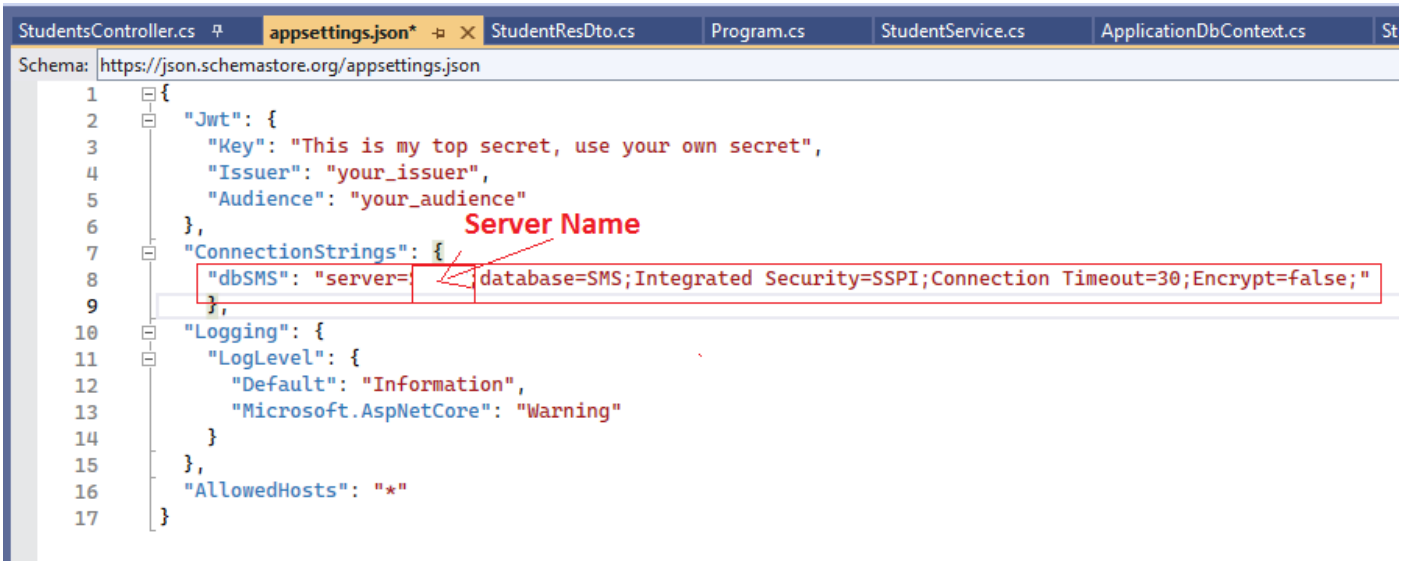
3. Configure the Database Connection

Open `appsettings.json` in the `SMS.API` project and **update your SQL Server connection string**:

```
"ConnectionStrings": {
  "DefaultConnection":
"Server=localhost;Database=SMS_DB;Trusted_Connection=True;MultipleActiveResultSets=true
"
}
```

Make sure the database name `SMS_DB` and SQL Server instance (e.g., `localhost`, `.\SQLEXPRESS`) match your setup.

Set up database details in the appsettings.json



---

4. Run EF Core Migration and Update Database

Make sure **SMS.API** is the **startup project**, then open the **Package Manager Console** (PMC):

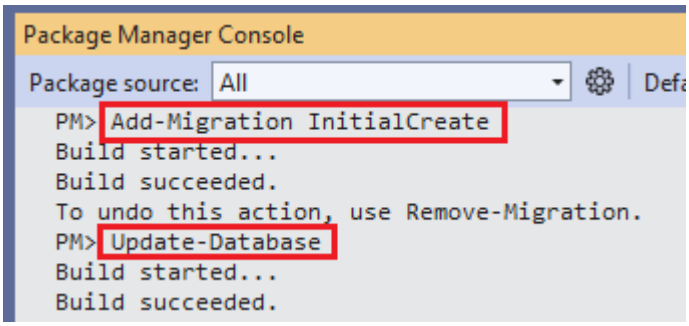
Option A: Using PMC in Visual Studio

```
PM> Add-Migration InitialCreate -Project SMS.Infrastructure -StartupProject SMS.API
PM> Update-Database -Project SMS.Infrastructure -StartupProject SMS.API
```

Run the following command in Package Manager Console so that it will automatically create database and its tables

```
PM> Add-Migration InitialCreate
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> Update-Database
```

```
Build started...
Build succeeded.
```



Option B: Using .NET CLI

```
dotnet ef migrations add InitialCreate -p SMS.Infrastructure -s SMS.API
dotnet ef database update -p SMS.Infrastructure -s SMS.API
```

```
echo Migration-DB Start...

dotnet ef migrations add InitialCreate -p SMS.Infrastructure -s SMS.API

dotnet ef database update -p SMS.Infrastructure -s SMS.API

echo Migration-DB End...
```

These commands will automatically create the `SMS_DB` database and all related tables.

▶☐ 5. Run the API

- In Visual Studio: Press `F5` or click on the **Run** button.
- In CLI: Navigate to `SMS.API` project and run:

```
cd SMS.API
dotnet run
```

Once the application runs, it should open Swagger UI at:

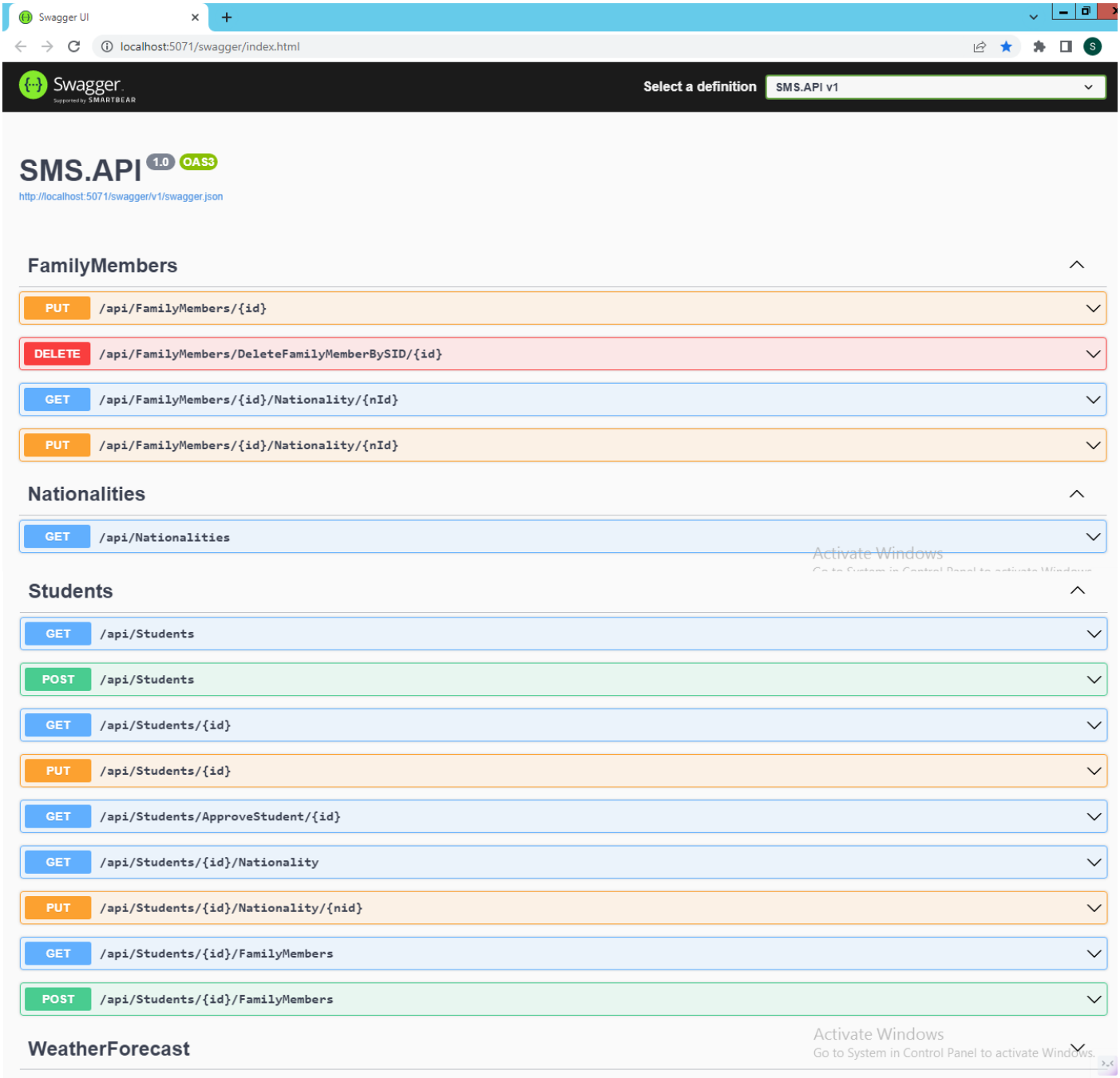
```
http://localhost:5071/swagger/index.html
```

📌 6. Verify Endpoints in Swagger

Test the following APIs in Swagger:

http://localhost:5071/swagger/index.html

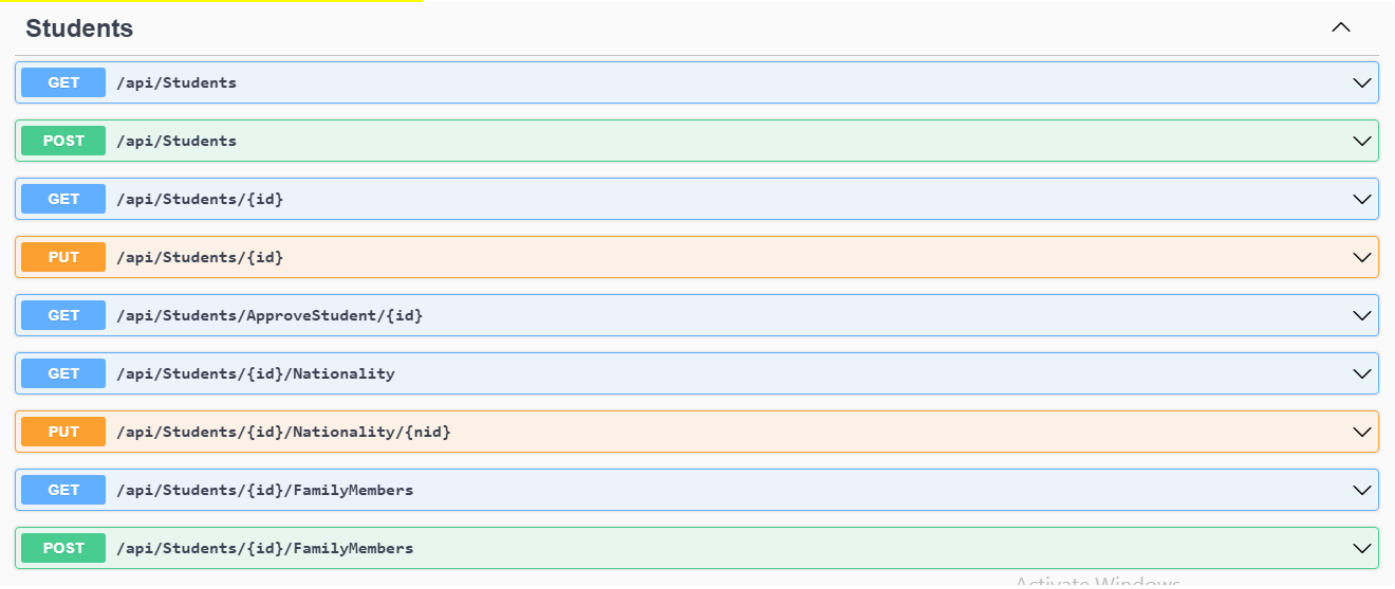
SMS.API with FamilyMembers, Nationalities and Students APIs



Students

- GET /api/Students
- POST /api/Students
- PUT /api/Students/{id}
- GET /api/Students/{id}/Nationality
- PUT /api/Students/{id}/Nationality/{id}

Students Endpoints:



Endpoints

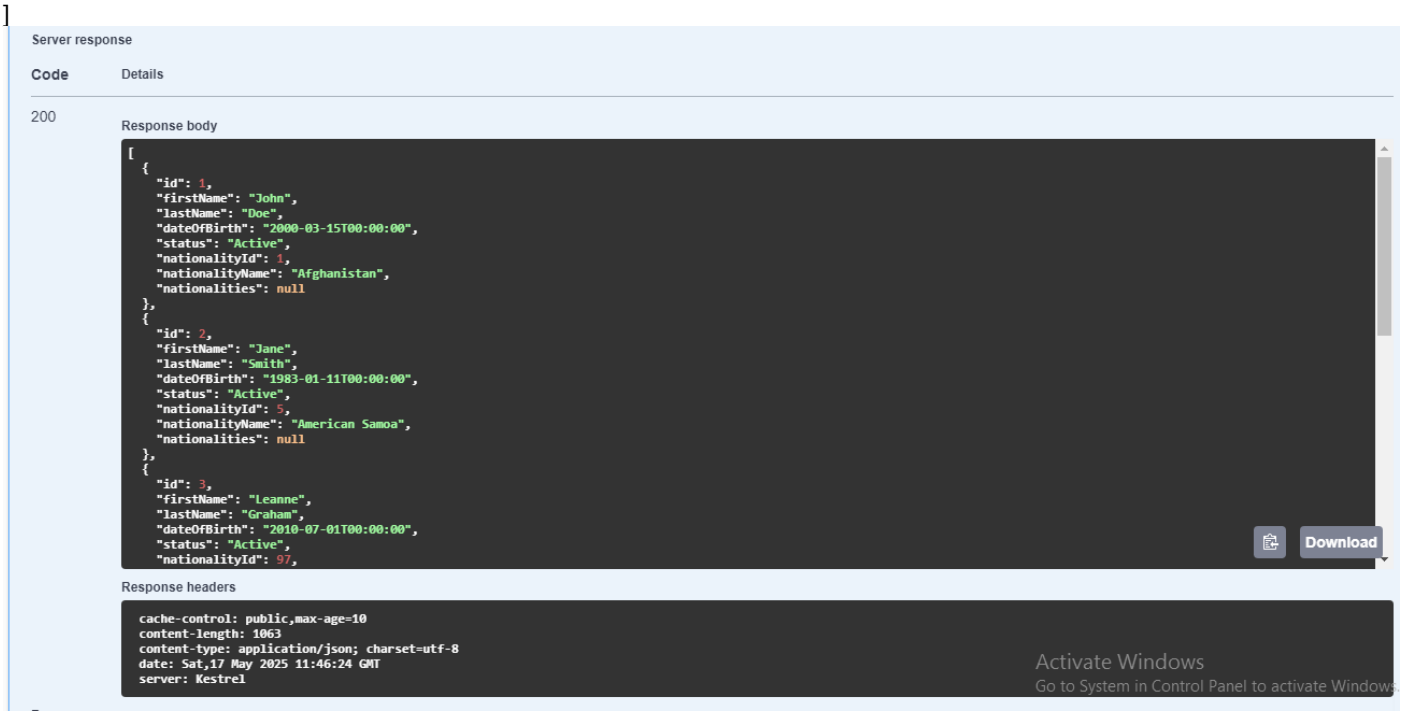
The following endpoints need to be created in order to fetch and store data.

Students

Get all Students

Response Sample:

[
{
"ID": 1,
"firstName": "John",
"lastName": "Doe",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
}
]



Add a new Student with Basic Details Only

Request Sample:

{
"firstName": "John",

"lastName": "Doe",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
}

Response Sample:

{
"ID": 1,
"firstName": "John",
"lastName": "Doe",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
}

Updates a Student's Basic Details only

Request Sample:

{
"ID": 1,
"firstName": "John Update",
"lastName": "Doe",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
}

Response Sample:

{
"ID": 1,
"firstName": "John Update",
"lastName": "Doe",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
}

Gets the Nationality of a particular student

Response Sample:

{
"ID": 1,
"firstName": "John",
"lastName": "Doe",
"nationalityId": 1,
}

Updates a Student's Nationality

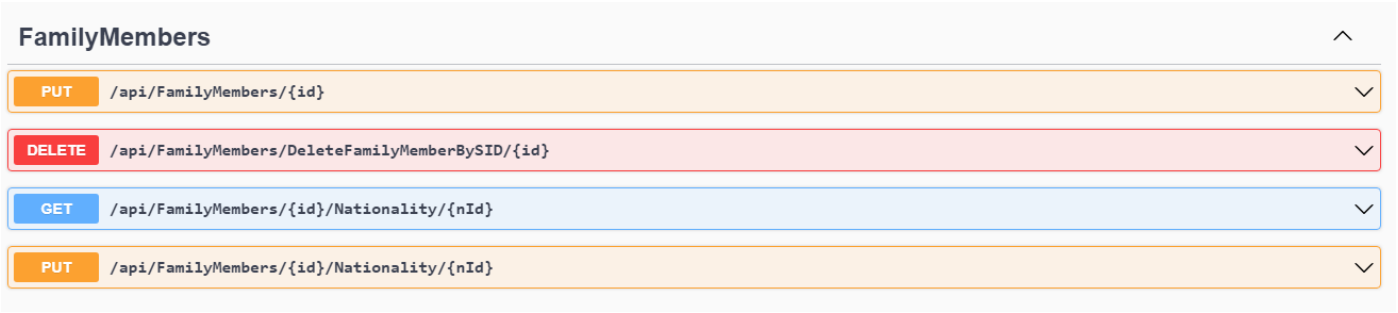
Response Sample:

{
"ID": 1,
"firstName": "John",
"lastName": "Doe",
"nationalityId":2
}

Family Members

- `GET /api/Students/{id}/FamilyMembers/`
- `POST /api/Students/{id}/FamilyMembers/`
- `PUT /api/FamilyMembers/{id}`
- `DELETE /api/FamilyMembers/{id}`
- `GET /api/FamilyMembers/{id}/Nationality/{id}`
- `PUT /api/FamilyMembers/{id}/Nationality/{id}`

Family Members Endpoints:



Gets Family Members for a particular Student

Response Sample:

[
{
"ID": 2,
"firstName": "John Son",
"lastName": "John Family",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
"relationshipId": 3
}
]

Creates a new Family Member for a particular Student (without the nationality)

Request Sample:

{
"firstName": "John Son",
"lastName": "John Family",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
"relationshipId": 3
}

Response Sample:

{
"ID": 2,
"firstName": "John Son",
"lastName": "John Family",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
"relationshipId": 3
}

Family Members

Updates a particular Family Member

Request Sample:

{
"firstName": "John Son",
"lastName": "John Family",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
"relationshipId": 3
}

Response Sample:

{
"ID": 2,
"firstName": "John Son",
"lastName": "John Family",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
"relationshipId": 3
}

Deletes a family member for a particular Student

Gets a nationality associated with a family member

Response Sample:

{
"ID": 2,
"firstName": "John Son",
"lastName": "John Family",
"dateOfBirth": "2023-07-31T12:44:55.403Z"

"relationshipId": 3,
"nationalityId":2
}

Updates a particular Family Member's Nationality

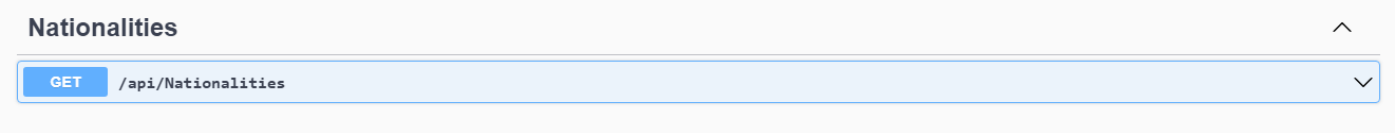
Response Sample:

{
"ID": 2,
"firstName": "John Son",
"lastName": "John Family",
"dateOfBirth": "2023-07-31T12:44:55.403Z"
"relationshipId": 3,
"nationalityId":1
}

Nationalities

- `GET /api/Nationalities`

Nationality Endpoints:



Nationality

Gets all nationalities in the system

□ 7. Run Unit Tests (Optional)

In Visual Studio:

- Open **Test Explorer** → Run All Tests.

In CLI:

```
cd SMS.Tests
dotnet test
```
