

# Assessment Document

## Assessment requirements

1. Use an external API.
2. Retrieve data from the API and save it to a SQL database.
3. Create two endpoints:
  - **Get all student details**
  - **Get student details by ID**
4. If data by ID is not found in the database/removed from the database, retrieve it from the API and save it to the database.

## Method I used

1. **Fetch and Save from API:**
  - First, I get student data from the external API.
  - For each student, I check if their ID already exists in the database.
  - If the ID already exists, I skip saving that student to avoid duplicates.
  - If the ID is new, I save the student data into the database.
2. **Get All Students:**
  - I fetch and return all the student records from the database.
3. **Get Student by ID:**
  - First, I check if the student exists in the database.
  - If found, I return the data from the database.
  - If not found, I call the external API to get that specific student's details, save it to the database, and return it.

## Frameworks and Libraries Used

- **ASP.NET Core Web API** - To build a RESTful API ( GET, POST, PUT, DELETE)
- **Newtonsoft.Json** - Convert the JSON text into a list of objects.
- **ADO.NET (System.Data.SqlClient)** - To access the database by SQL queries without using EF
- **HttpClient** - Get details from the API

## Folder Structure

- **appsettings.json** - Configuration data like DB connection string and external APIs
- **Controllers** - To create methods to handle HTTP requests.
- **Models** - Define the data structure ( Show the table fields in DB )

## Project Setup

### 1. Create a External API

- This is the API I used to get student details.
- <https://mocki.io/v1/f15b358b-5d58-46c4-9153-c39577ab60b9>

### 2. Clone the Project

- Clone or download the repository.
- Open the project in Visual Studio.

### 3. Set configuration

- Open the appsettings.json file.
- Add the database connection and url of the API.

```
{  
  
  "ConnectionStrings": {  
  
    "DefaultConnection":  
    "Server=MSI\\SQLEXPRESS;Database=AssesmentDB;Trusted_Connection=True;TrustServerCertificate=True;"  
  
  },  
  
  "APISettings": {  
  
    "BaseUrl": "https://mocki.io/v1/",  
  
    "Endpoint": "f15b358b-5d58-46c4-9153-c39577ab60b9"  
  
  }  
}
```

In ConnectionString **AssesmentDB** is my database name in SQL. Add your database name from the SQL.

When adding the url :

- Base Url - The first part of the API with HTTP (<https://mocki.io/v1/>)
- Endpoint - the last part of the URL ([f15b358b-5d58-46c4-9153-c39577ab60b9](https://mocki.io/v1/f15b358b-5d58-46c4-9153-c39577ab60b9))

## Set up the SQL

Run the following SQL queries to set up the database and table:

```
CREATE DATABASE AssessmentDB
```

```
CREATE TABLE Students (
```

```
    Id INT PRIMARY KEY,
```

```
    StudentID NVARCHAR(50),
```

```
    StudentName NVARCHAR(100),
```

```
    StudentAge INT
```

```
);
```

## Running the Application

1. Run the project from Visual Studio.
2. Use **Postman** or swagger to test the endpoints.

## Generated EndPoints

**GET /api/Students/FetchAndSaveStudents** - Fetch data from API and save to SQL DB

**GET /api/Students/GetStudentList** - Get all students from the database

**GET /api/Students/FindById/{id}** - Get student by ID; if not found in DB, fetch from API and save

Example:

- <https://localhost:44334/api/Students/FetchAndSaveStudents>
- <https://localhost:44334/api/Students/GetStudentList>
- <https://localhost:44334/api/Students/FindById/5>

Replace the port number (44334) with the one your local project is running on.