Import Data Files (xlsx, csv) into SQL Server using ASP.NET Core

Table of Contents

1. Introduction

This documentation outlines the process of importing data files (XLSX and CSV) into a SQL Server database using ASP.NET Core 7. The project aims to create a robust and efficient web application that provides users with a seamless experience for importing large volumes of data from various file formats.

2. Prerequisites

Before you begin, ensure you have the following prerequisites:

- Visual Studio or Visual Studio Code
- .NET 7 SDK
- SQL Server installed and running
- Basic knowledge of ASP.NET Core, C#, SQL

3. Project Overview

The project involves creating a web application with the following technologies and features:

- ASP.NET Core 7 for building the web API.
- SQL Server for data storage.
- XML Bulk Insert for efficient data insertion.
- Dapper for database operations.
- CsvHelper for CSV file processing.
- EPPlus for XLSX file processing.
- File validation, error handling, and data transformation.

4. Key Features

XLSX and CSV File Support: The application supports importing data from both XLSX and CSV file formats.

File Validation and Error Handling: Before data import, files are validated for format and structure. Detailed error handling provides informative messages to users.

Mapping and Transformation: Data mapping and transformation allow users to map columns from files to database fields and perform necessary transformations during import.

Data Import Performance Optimization: The application optimizes data import performance using XML Bulk Insert and Dapper to efficiently insert large volumes of data into the SQL Server database.

5. Implementation Steps

Step 1: Create a New ASP.NET Core Web API Project

1. Open Visual Studio or Visual Studio Code.
2. Create a new ASP.NET Core Web API project targeting .NET 7.

Step 2: Configure Database Connection

1. Configure a connection string to your SQL Server database.

Step 3: Install Required NuGet Packages

Install the necessary NuGet packages:

- CsvHelper
- EPPlus
- Dapper

Use the Package Manager Console or the .csproj file to add these packages to your project.

Step 4: Model Creation

Create a model class that represents the data you want to import.

Step 5: File Upload API

Create an API endpoint for file upload. Use `[FromForm]` to accept files.

Step 6: File Validation and Error Handling

Implement file validation and error handling logic within the `UploadFile` action.

- Check file size and extension.
- Provide detailed error messages to the user if validation fails.

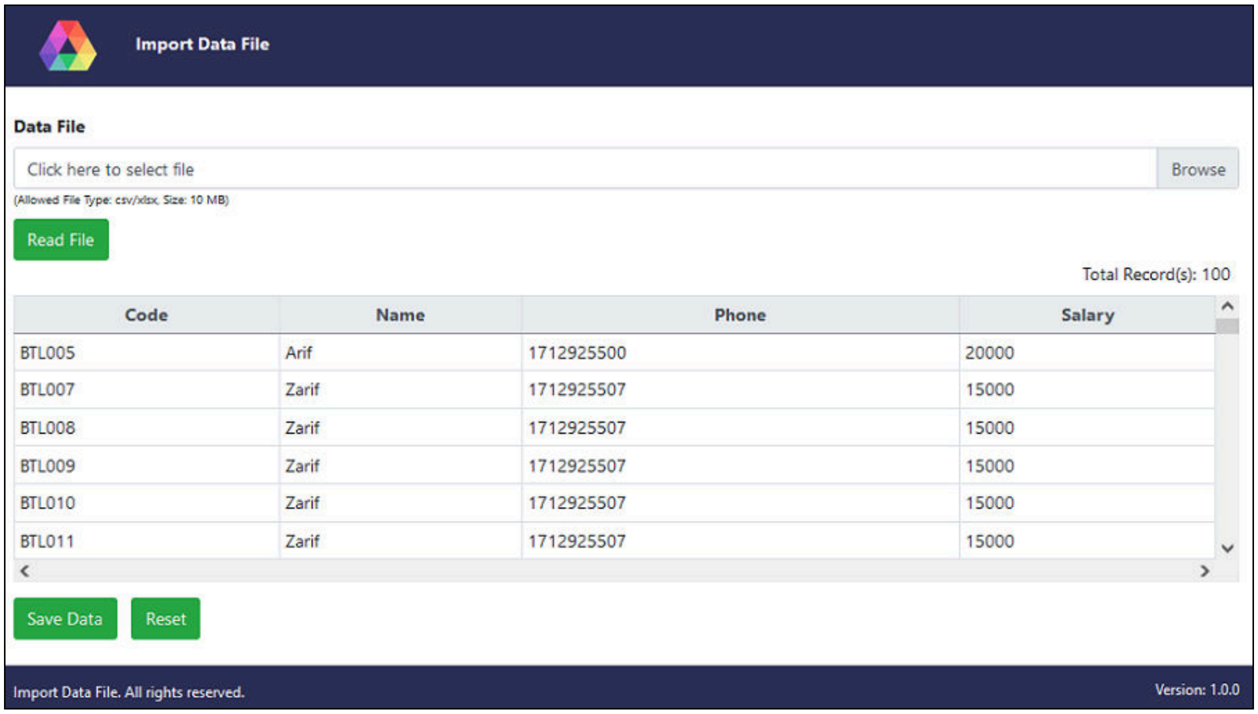
Step 7: Data Mapping and Transformation

Implement data mapping and transformation logic to map columns from the uploaded file to your database model.

Step 8: Data Import Performance Optimization

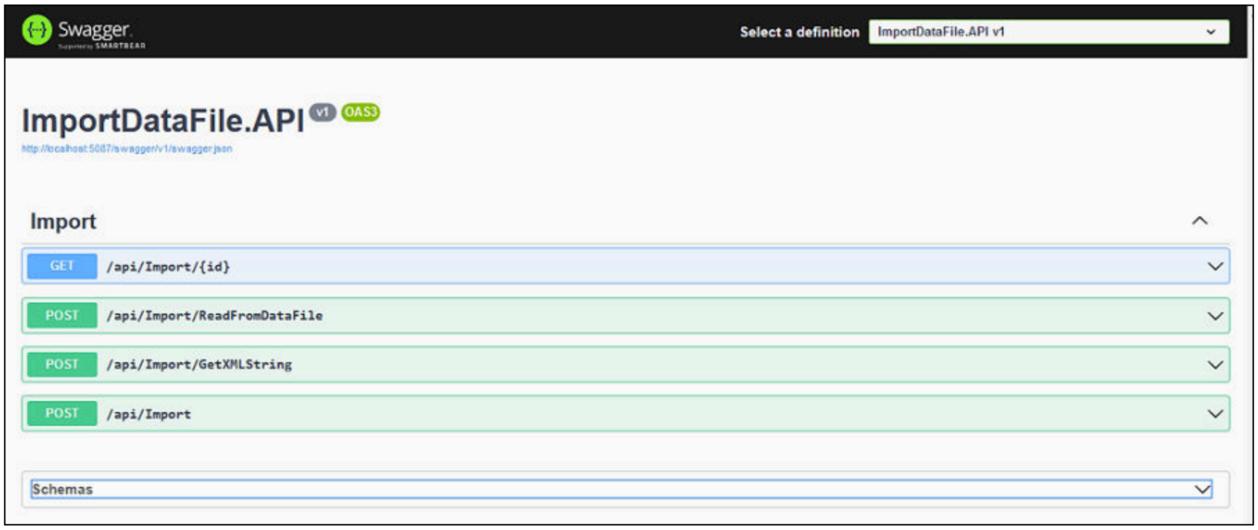
Optimize data import performance using XML Bulk Insert and Dapper to efficiently insert data into the SQL Server database.

6. Screenshots


Import Data File
Data File
Click here to select file
Browse
(Allowed File Type: csv/xlsx, Size: 10 MB)
Read File
Total Record(s): 100
Save Data
Reset
Import Data File. All rights reserved.
Version: 1.0.0

Code	Name	Phone	Salary
BTL005	Arif	1712925500	20000
BTL007	Zarif	1712925507	15000
BTL008	Zarif	1712925507	15000
BTL009	Zarif	1712925507	15000
BTL010	Zarif	1712925507	15000
BTL011	Zarif	1712925507	15000

Fig: View Data File



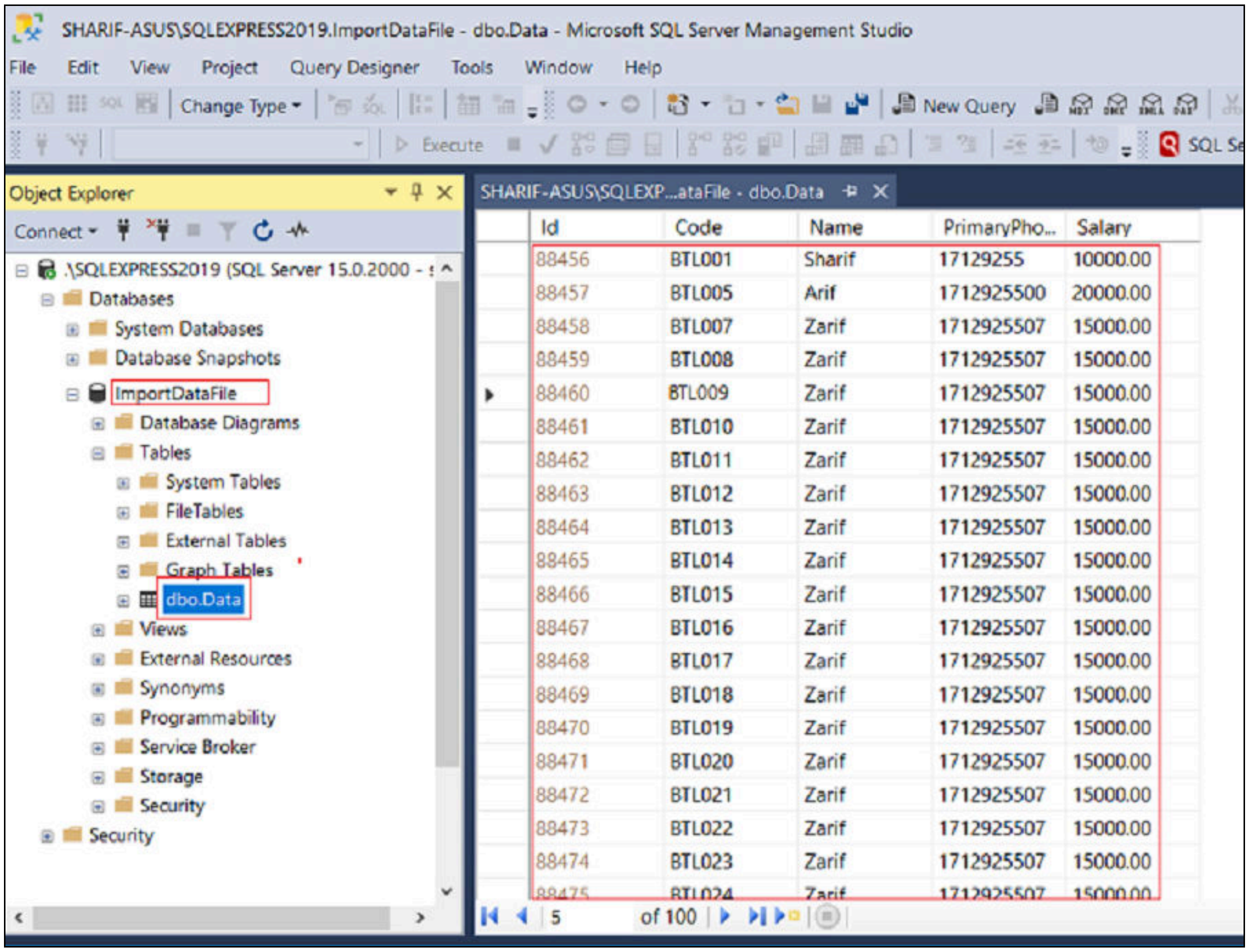
ImportDataFile.API v1 OAS3

http://localhost:5087/swagger/v1/swagger.json

Import

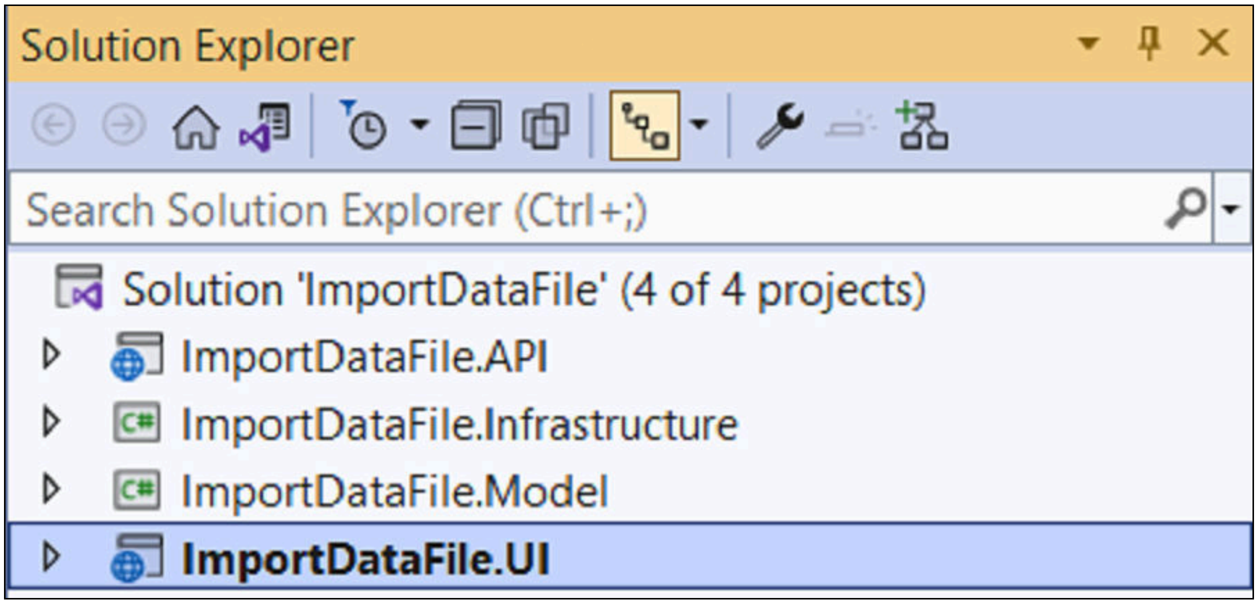
```
GET   /api/Import/{id}
POST  /api/Import/ReadFromDataFile
POST  /api/Import/GetXMLString
POST  /api/Import
```

Fig: API



Id	Code	Name	PrimaryPho...	Salary
88456	BTL001	Sharif	17129255	10000.00
88457	BTL005	Arif	1712925500	20000.00
88458	BTL007	Zarif	1712925507	15000.00
88459	BTL008	Zarif	1712925507	15000.00
88460	BTL009	Zarif	1712925507	15000.00
88461	BTL010	Zarif	1712925507	15000.00
88462	BTL011	Zarif	1712925507	15000.00
88463	BTL012	Zarif	1712925507	15000.00
88464	BTL013	Zarif	1712925507	15000.00
88465	BTL014	Zarif	1712925507	15000.00
88466	BTL015	Zarif	1712925507	15000.00
88467	BTL016	Zarif	1712925507	15000.00
88468	BTL017	Zarif	1712925507	15000.00
88469	BTL018	Zarif	1712925507	15000.00
88470	BTL019	Zarif	1712925507	15000.00
88471	BTL020	Zarif	1712925507	15000.00
88472	BTL021	Zarif	1712925507	15000.00
88473	BTL022	Zarif	1712925507	15000.00
88474	BTL023	Zarif	1712925507	15000.00
88475	BTL024	Zarif	1712925507	15000.00

Fig: Table with Imported Data



Solution 'ImportDataFile' (4 of 4 projects)
ImportDataFile.API
ImportDataFile.Infrastructure
ImportDataFile.Model
ImportDataFile.UI

Fig: Solution Explorer

7. Conclusion

This documentation has provided a detailed overview of creating a web application to import data files (XLSX and CSV) into a SQL Server database using ASP.NET Core 7. By following the outlined steps and utilizing the mentioned technologies and features, you can build a robust and efficient data import system tailored to your needs. Remember to implement proper error handling and data validation to ensure the reliability of your application.