

TopUp Beneficiary Solution - NET6

Created a robust backend solution for the topup beneficiary application, enabling users to efficiently manage their top-up beneficiaries, explore available top-up options, and execute top-up transactions for their UAE phone numbers. Implements a maintainable enterprise-level API application with Middleware, EntityFrameworkCore, Serilog, and Swagger using Domain Driven Design (DDD) and architecture principles.

Table of Contents

- [TopUp Beneficiary Solution - NET6](#)
 - [Table of Contents](#)
 - [Prerequisites](#)
 - [Architecture Overview](#)
 - [Instructions](#)
 - [Installation:](#)
 - [Database Setup](#)
 - [Third Party Libraries](#)
 - [Future Enhancements](#)

Prerequisites

You will need the following tools:

- [Visual Studio Code](#) or [Visual Studio 2022](#) (version 17.5.0 or later)
- [.NET Core SDK 6.0](#)
- Microsoft SQL Server

Architecture Overview

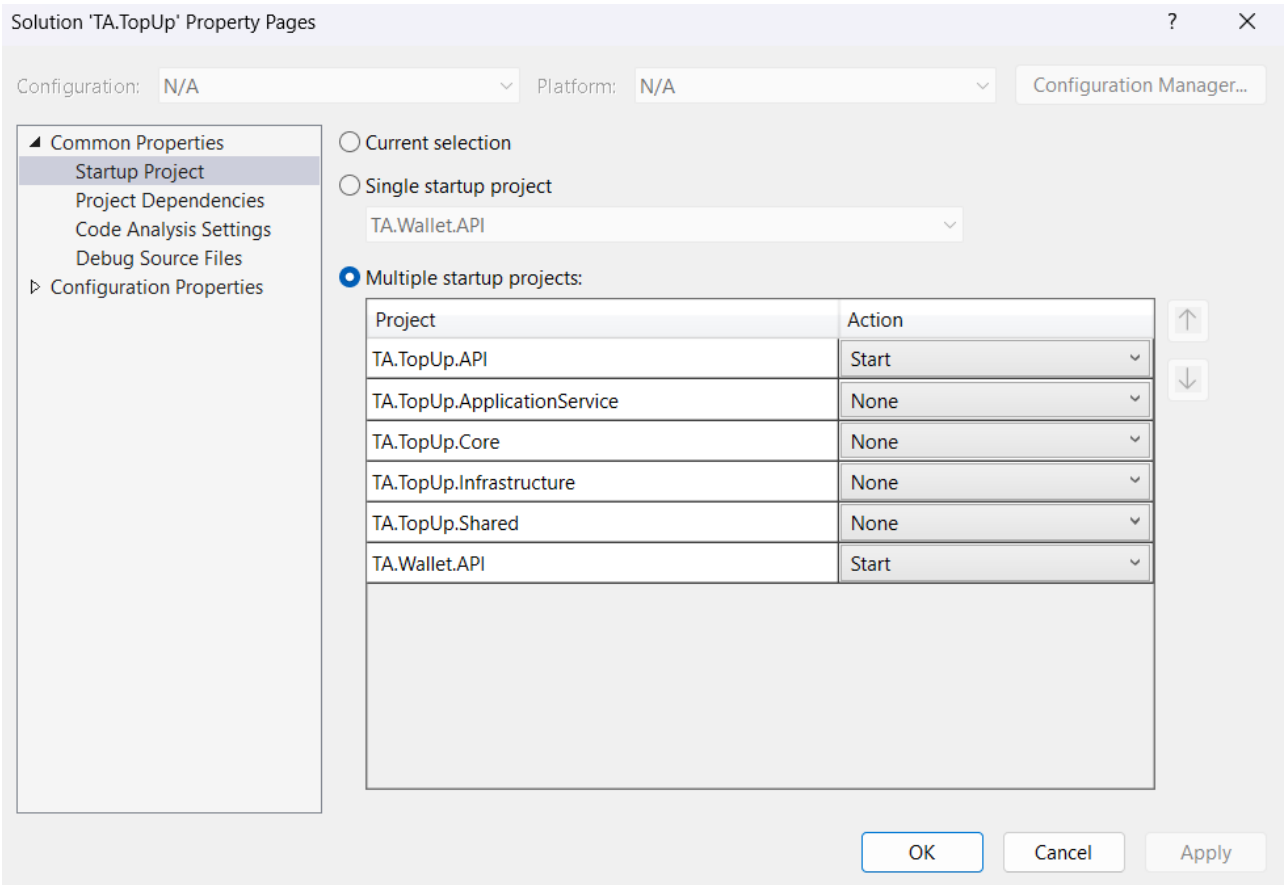
This is a multi-project solution that utilizes Domain Driven Design (DDD) to create a maintainable web API application using .NET 6 that allows it to run on Linux or Windows and in Docker environments.

Instructions

Installation:

1. Install the latest [.NET Core 6 SDK](#).
2. Used Database first approach, so kindly execute the attached SQL script.
3. Clone Github repo <https://github.com/Shivvvvvvvv/TechnicalAssessment.git>
4. Restore Nuget Packages

5. Rebuild and run the program by setting below projects as startup,



You should be able to browse and test the application api by using the below URL :

```
Swagger - TopUp Service: https://localhost:7024/swagger/index.html
Swagger - Wallet Service: https://localhost:7085/swagger/index.html
```

Database Setup

To setup the SQL Server database following the instructions below:

- 1. Reveiw the **connection string** in **appsettings.json** in below projects and update.

```
..\TA.TopUp.API\appsettings.json
..\TA.Wallet.API\appsettings.json
```

- 2. Execute the Script uploaded in below path `./Script/ScriptWithMasterData.sql`

	UId	FirstName	LastName	IsVerified
1	1	Tom	A	1
2	2	Jez	B	1
3	4	Jacob	C	0

Master Data to test #User Table

UId	UserId	Balance	CurrencyId
1	1	14.00	1
3	2	10.00	1
4	4	39.00	1

#Wallet Balance

NickName	MobileNumber	CountryCode	UserId
TomB	0544066653	+971	1
ZenA	0169758085	+971	2
BenB	4973441455	+971	4
TomA	8357730789	+971	1
TomC	8357730789	+971	2
TomD	5449444432	+971	2

#Beneficiary

#Sample request data added above each api. Kindly check in the controller.

Third Party Libraries

- Serilog

Future Enhancements

- Implement migration concept to create application database on initial run
- Implement polly
- Implement circuit breaker
- Refractor Topup Service
- Write Unit test and Integration test