TopUp Project Technical Documentation

Overview

The project is designed to manage mobile top-up transactions for users. It allows users to add beneficiaries, view their beneficiaries, and top up their beneficiaries' mobile numbers securely and reliably. The project is built using microservices architecture to ensure scalability, flexibility, and maintainability.

Onion Architecture

The services follow Onion Architecture to ensure a clean separation of concerns:

- **Domain Layer:** Contains core business logic and domain entities.
- **Application Layer:** Contains service interfaces and implementations.
- **Infrastructure Layer:** Contains data access, external services, and repository implementations.
- **Presentation Layer:** Contains API controllers and middleware.

Technology Stack

- **ASP.NET Core:** Framework for building the web API.
- **Entity Framework Core:** ORM for database interactions.
- **SQL Server:** Database management system.
- **JWT (JSON Web Tokens):** For secure authentication and authorization.

Unit Testing Implementation

Unit testing for the BeneficiaryService in TopUp Project was implemented using xUnit and Moq to ensure the service's functionality is reliable and meets the expected behavior. The in-memory database provided by Entity Framework Core was used to simulate database interactions, enabling isolation of the service logic from the actual database.

Repositories

https://github.com/dev92anas/TopUp.git

Using the Implemented API (POSTMAN COLLECTION)

To help users interact with the API, a Postman collection is provided. This collection includes predefined requests for all the endpoints, making it easy to test and use the API. The Postman collection file is included in the project files.

Cloud Deployment

The SQL databases for each service are deployed in the cloud (Azure SQL Database). Each service has its own dedicated database to ensure data separation and integrity.

===========

Server Name: topup92.database.windows.net

Username: topup

Password: ToPo@1234

===========

Build Instructions

To build and run the project using Visual Studio 2022, follow these steps:

1. **Clone the Repository: git clone https://github.com/dev92anas/TopUp.git**
2. **Open Visual Studio 2022 Go to File > Open > Project/Solution. Navigate to the cloned repository folder and open the solutions file.**
3. **Restore NuGet Packages Right-click on the solution in the Solution Explorer. Select Restore NuGet Packages. for each Solution (TopUpService, BalanceService, SecurityService, APIGateway)**
4. **Update Connection Strings Open the appsettings.json file for each service solution project (TopUpService.API, BalanceService.API, SecurityService.API). Replace the connection strings with the ones provided in the attachment.**
5. **Build the Solution Right-click on the solution in the Solution Explorer. Select Build Solution.**
6. **Run the Application Set multiple startup projects (TopUpService.API, BalanceService.API, SecurityService.API, ApiGateway) by right-clicking the solution and selecting Proprties and update the multiple startup projects.**
7. **Press F5 to run the application. Ensure that all services (TopUpService, BalanceService, SecurityService, APIGateway) are running together.**
8. **Import Postman Collection Import the provided Postman collection into your Postman application.**
9. **Steps to Use the API**

   - **Import Postman Collection: Import the provided Postman collection into your Postman application.**
   - **Authenticate: Use the SecurityAPI to obtain a JWT token by logging in.**
   - **Add a Beneficiary: Use the Add Beneficiary endpoint to add a new beneficiary.**
   - **Get Lookups: Retrieve your list of lookups Amount using the "get-topup-options" endpoint, you will use the lookupId on topup.**
   - **Top-Up a Beneficiary:**
     - **Generate Idempotency Key: Enter a unique idempotency key for the top-up request.**
     - **Perform Top-Up: Use the Top-Up Beneficiary endpoint, including the lookup Id for the amount and idempotency key in the request to ensure that the operation is idempotent.**

Conclusion

The TopUpService project implements a robust system for managing mobile top-up transactions using microservices and Onion Architecture. By leveraging ASP.NET Core, Entity Framework Core, and JWT for secure authentication, the project ensures reliable and secure operations. The provided Postman collection facilitates easy interaction with the API, and the cloud deployment of databases ensures reliability and accessibility.