

Portfolio Backend Runbook

Application

Description

Owner

Service

Version

Version Date

Portfolio Backend

Data Storage

Dillon Coleman

SaaS

1.0

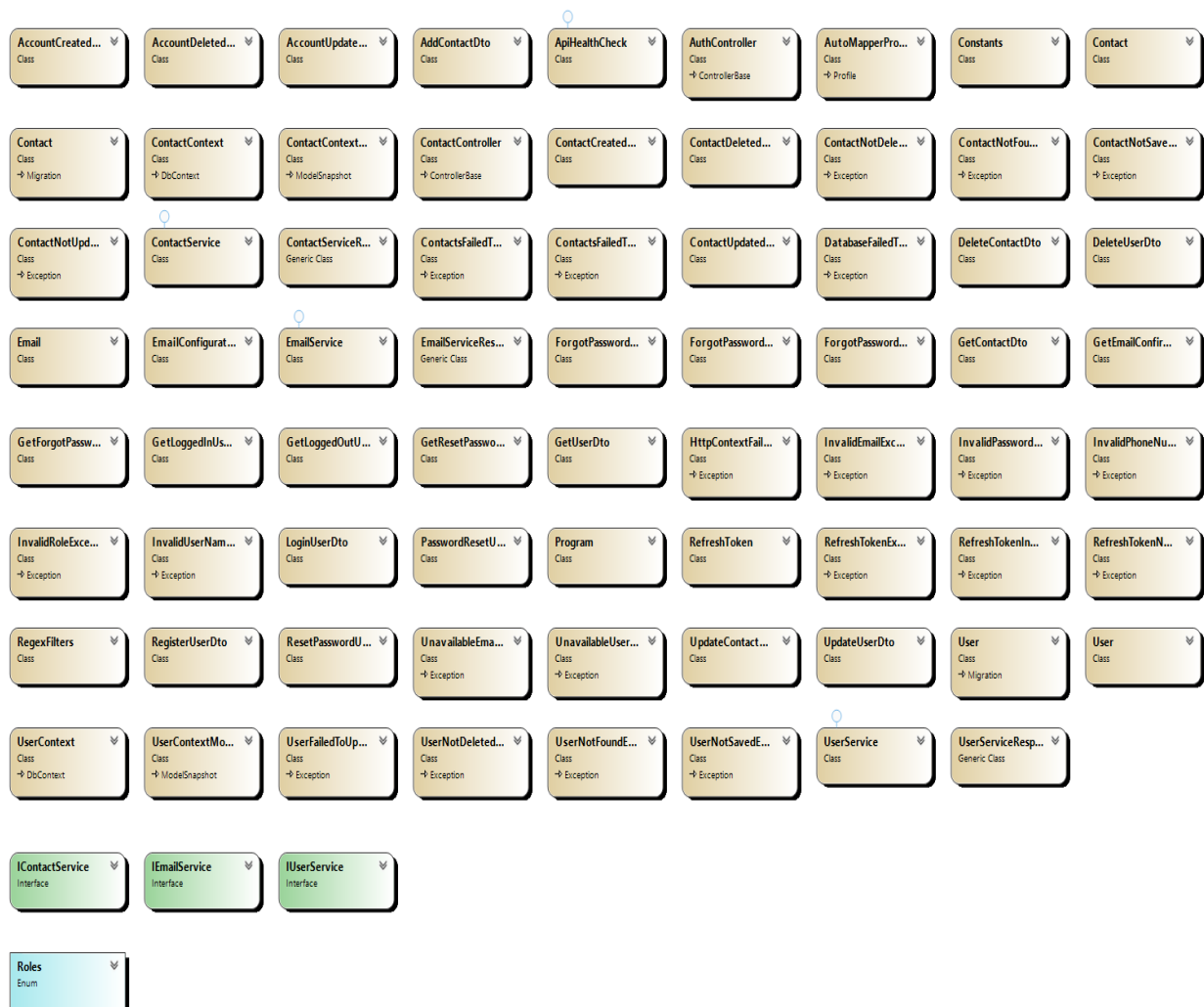
8.7.23

Architecture

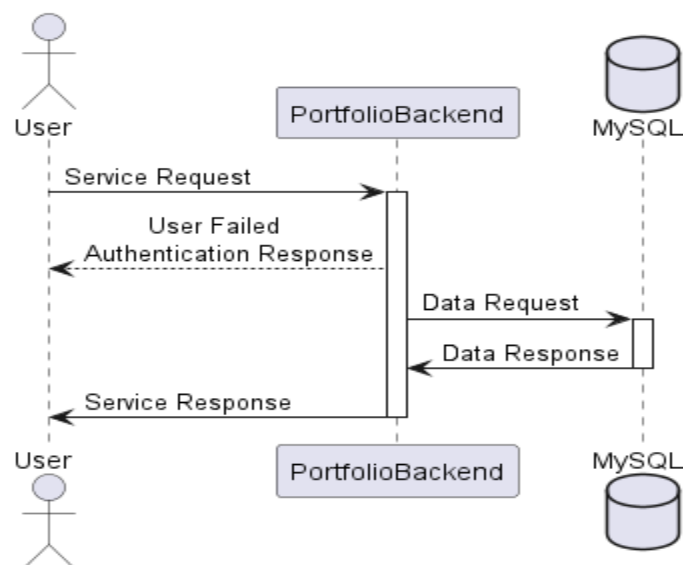
Description: A .NET 7 API for storing contact information and user data, including CRUD operations, login/logout functionality, using a MySQL database, and hosted on Azure.

Diagrams:

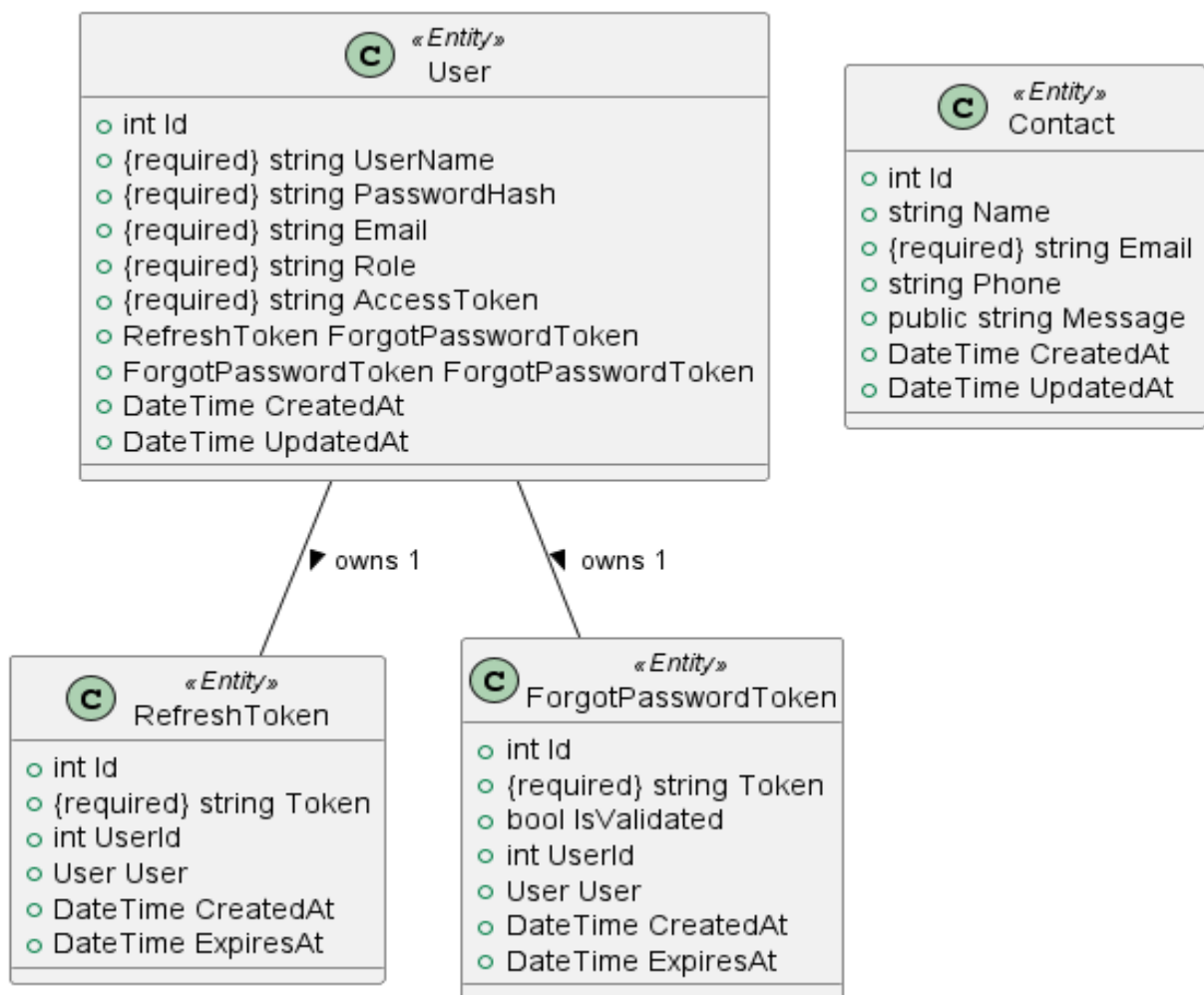
Class Diagram



Sequence Diagram



Entity Diagram



Operational Procedures

Startup Procedure:

Locally:

1. Open terminal
2. Navigate to project folder
3. Enter “dotnet run”, and press enter

Production:

1. Navigate to PortfolioWebsiteBackend Web App in Azure
2. Select “Overview” from the left navigation panel
3. Select “Start” from the top navigation panel

Shutdown Procedure:

Locally:

1. Open working terminal
2. Press Ctrl + C

If you’ve close or cannot find the working terminal press the red square icon next to the play button in your ide.

Production:

1. Navigate to PortfolioWebsiteBackend Web App in Azure
2. Select “Overview” from the left navigation panel

Select “Stop” from the top navigation panel

Health Check Procedure:

1. Navigate to {Base URL}/healthcheck

Environment Details

Environment Configuration:

In Development, the service uses Mocks as a substitute database. Additional Users and Contacts can be added when needed. This was used as a cost effective and reliable backend. In future updates, I would like to move the Mocks out and replace them with a MySQL test container.

In Productions, all calls become asynchronous. There are actively two instances of the application always running.

Release Procedure

Deployment Steps:

Notes: Deployment using Docker and Azure Container Registry

1. In Visual Studio, right click "PortfolioBackend" and select "Publish"

*Note - If you haven't set up a publish configuration yet a Publish UI will appear. Follow the below step to create publish configuration.

1. Target = Azure
2. Specify target = Azure Container Registry
3. Registry = Create new and select it
4. Container build = Docker Desktop
2. Select "Publish"

Rollback Procedure:

Notes: There are three branches, main, develop, and rollback

1. In Visual Studio, Publish the rollback branch

Monitoring & Alerts

Monitoring:

- Azure Application Insights
- Health Check
- Logging

Alert Procedures:

Currently there are only two alerts. One that sends an email and text when the application is unavailable, and one that tracks abnormal rise in the rate of HTTP requests or dependency calls that are reported as failed.

Known Issues

Common Issues:

- Mock User Tokens Expire – They are set to expire in 5 years BUT that doesn't mean something can't happen and they'll need to be recreated.

Troubleshooting

Debugging Steps:

Local:

- Verify User Secrets access
- Check the expiration date of tokens
- Check Logs

Production:

- Perform Health Check
- Check Application Insights
- Check Logs
- Monitor Performance
- Check Database

Files

Postman Collection:



Portfolio Website Backend Public.postman_collection.json

Living Doc:



LivingDoc-8.7.23.html

JMX file:



PortfolioWebsite_Backend_Performance.jmx

If the links do not work, I've added a copy of the files in the runbook folder.

Notes

Common Commands:

- dotnet watch run
- dotnet watch test
- dotnet ef migrations add <<NAME>> --context <<CONTEXT>>
- dotnet ef database update --context <<CONTEXT>>
- dotnet run --launch-profile
- livingdoc test-assembly
.\PortfolioBackend_ATDDs\bin\Debug\net7.0\PortfolioBackend_ATDDs.dll -t
.\PortfolioBackend_ATDDs\bin\Debug\net7.0\TestExecution.json -o .\LivingDoc.html

Contact Information

Author: Dillon Coleman

Email: coleman399@gmail.com

GitHub: <https://github.com/coleman399>

LinkedIn: <https://www.linkedin.com/in/dillonthedev/>