FinChatter

Simple chat application with a bot that allows getting stock quotes.

Introduction

This is a simple web chat application built with .Net Technologies For its development, best development practices were taken into consideration and it is based on clean architecture.

The application allows several users to communicate using different chatrooms and also to get stock quotes with the built-it chat boot.

Features

- Use .NET identity for user registration to log in and talk with other members in different chat rooms.
- Allow user to use a chat bot for query stock quotes.
- The bot use RabbitMQ for messaging.
- Download stock info from the API https://stooq.com as CSV file and parse the information to send it to the chatroom that was requested.
- Has independent services for API, WEB, Chat Boot.
- Health-check for RabbitMQ monitoring.
- All messages are ordered by their timestamps and show up to 50 messages.
- Allow users to create more chatrooms.
- Unit Test main functionalities.

Technologies

- NET 6.0
- Blazor Webassembly for UI
- MSTest
- SQLite
- RabbitMQ
- SignalR
- Bootstrap 5

Getting Started

The easiest way to get started is using Visual Studio 2022 or 2019, but you can use any other IDE of your preference. There is some stuff that we need to install before we can run our project, so let's start with those.

1. Install the latest .NET 6 SDK
2. Install Docker Desktop

   > Docker is only required if you want to use RabbitMQ docker images. That's my default and easiest choice, but you can install it directly if you want.

3. A instance of RabbitMQ running. As I said before my choice is create a container using the official RabbitMQ docker image.

   We have to choices:

   - Use the batch file include in the project.
   - Run the following command:

     ```
     docker run -d --hostname finchattermq --name finchatter-rabbit -p
     5672:5672 -p 15672:15672 -e RABBITMQ_DEFAULT_USER=usr -e
     RABBITMQ_DEFAULT_PASS=Qwerty123$ rabbitmq:3-management
     ```

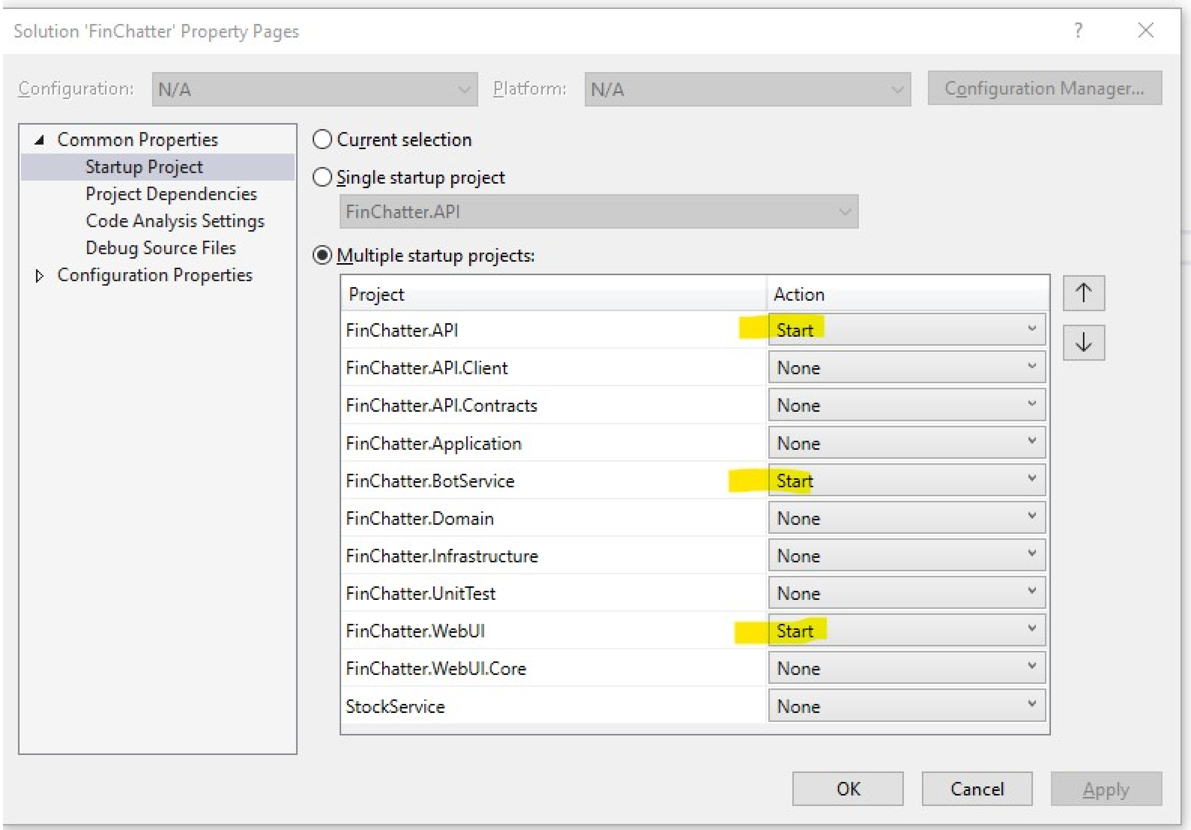
     Please notice that I am using all parameter with the exact value I have in the appsettings.json for the API. That's something you can change, but if you do in the next you need to update the appsettings.json.

4. Now we need to clone this repo after that we will have the file FinChatter.sln that is our solution. Let's open it to find several projects:

Project name	Folder	Short description
FinChatter.API	src	Main API project for Authentication and Chat base on SignalR.
FinChatter.API.Client	src	Client for FinChatter.API is use for FinChatter.WebUI.Core
FinChatter.API.Contracts	src	All contracts related with the FinChatter.API. Is shared with WebUI.
FinChatter.Application	src	Defines interfaces that are implemented by outside layers.
FinChatter.BotService	src	Decoupled boot service. Also includes health-check for RabbitMQ.
FinChatter.Domain	src	Everything related with the domain.
FinChatter.Infrastructure	src	Contains implementation for external resources.
FinChatter.WebUI	src	Our main UI project in Blazor Webassembly.
FinChatter.WebUI.Core	src	All components for the UI Blazor project.
StockService	src	Client service to call https://stooq.com API
FinChatter.UnitTest	test	All Unit test

Usage

When you first start the project in Visual Studio all NuGet packages start to update. After that the first thing you need to do is start all the projects going to the solution property window by doing right click in the solution and then choosing the option "Properties".



Solution 'FinChatter' Property Pages

Configuration: N/A

Platform: N/A

○ Current selection

○ Single startup project

● Multiple startup projects:

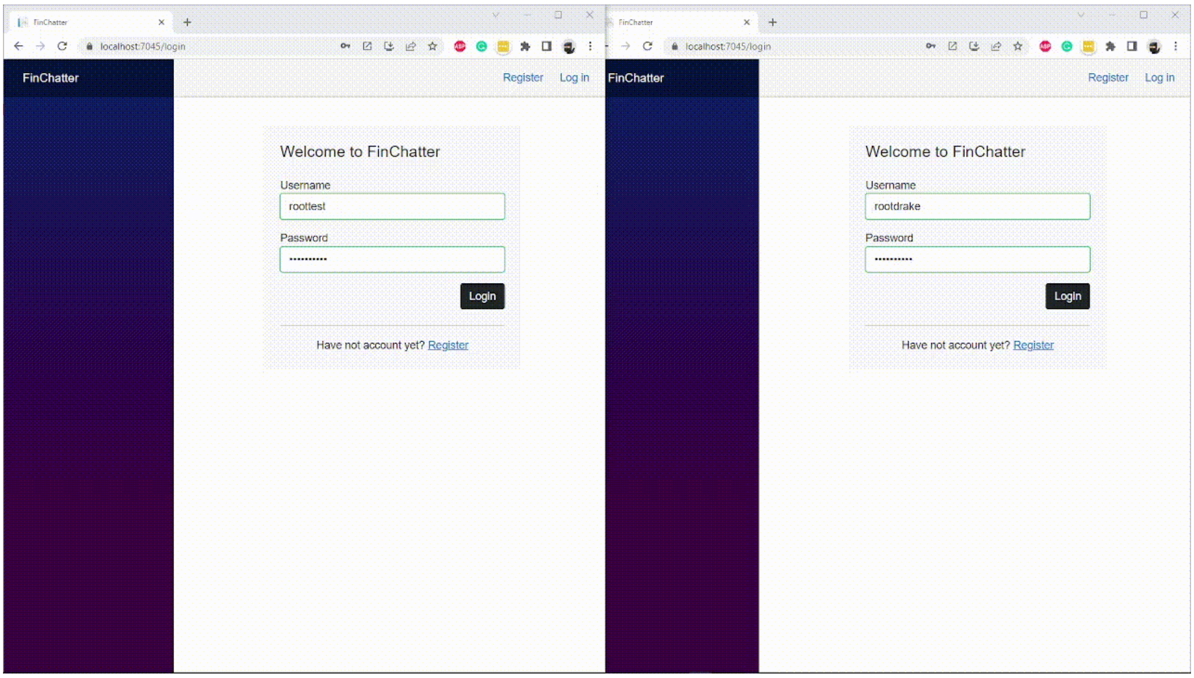
Project	Action
FinChatter.API	Start
FinChatter.API.Client	None
FinChatter.API.Contracts	None
FinChatter.Application	None
FinChatter.BotService	Start
FinChatter.Domain	None
FinChatter.Infrastructure	None
FinChatter.UnitTest	None
FinChatter.WebUI	Start
FinChatter.WebUI.Core	None
StockService	None

After that, you can start the projects and proceed to create a user to use the chat.

The database is in SQLite and you can found it the the following path src/FinChatter.API/finchatter.db. In case you wanna to delete the database you can do it and then go to Tools menu and chose Nuget Package Manager > Package Manager Console. After the windows is open choose the project FinChatter.API and run the command:

```
Update-Database
```

Demo



Welcome to FinChatter

Username: roottest

Password: ••••••••••

Have not account yet? Register

Welcome to FinChatter

Username: rootdrake

Password: ••••••••••

Have not account yet? Register