How to get started with the ATM project?

Introduction

Last semester we started on the development of the Automated Trading Platform (ATM). This platform should be developed so that users can create an account and use their own developed stock trading strategies to test them on the platform with different variable ranges like time, risk factors, etc. Eventually the platform will be able to run real-time analysis and keep testing with recent data and make prediction. But first it will need to be able to test with historical data.

Tech stack

Our project is developed using Golang with RabbitMQ as a message broker and MongoDB as a NoSQL database. The front end is developed in Angular and we use Docker to containerize our applications.

Prior knowledge

Before starting and continuing the project, there are some topics that are relevant to know. We will create a list as a starting point that is needed for this project and to understand our code:

- Basic messaging, we are using RabbitMQ, our code creates queues and consumes and produces messages from that queue.
- Basic functionality of MetaTrader 5, what a strategy file looks like, what are some settings you can adjust, how a config file looks like, how to start a test from the command line.
- Very basic knowledge about wine, what it is, how can I use it to run windows applications on Linux distros.
- Very basic knowledge of xvfb or an alternative to run applications with a virtual display so you don’t need a GUI.
- Basic knowledge about the tech stack.
- Basic Knowledge about Kubernetes jobs
- Basic knowledge about API’s

Prerequisites

To continue working on the ATM project you need to do a few things to get started:

1. Golang version V1.20
2. Angular V15.1
3. Docker V4.18.0
4. Kubernetes V1.24
5. Create an CloudAMQP account with a free RabbitMQ broker
6. Create a MongoDB account with a free database cluster
7. Pull ATM repos
8. Change RabbitMQ broker url and MongoDB url

MongoDB

This is what your database structure should look like:

- Results
  - csv
  - entries
- Stockbrood_testmanager
  - strategies
  - tests
- Testing
  - strategies

Docker hub

All our repos have a Github Actions that pushed the latest version to Docker hub. See header links.

All documentation related to each service can be found in the README in the corresponding Github repository.

Repositories

More info about all the repositories can be found in the organisation

Platform (frontend)

- Log in functionality (creating account or google login)
- Added a screen to import EA's and then run a test.
- https://github.com/S-A-RB05/platform

Strategy service

- Expert advisor (EA) storing functionality
- Passing strategy to test manager
- https://github.com/S-A-RB05/StratService

Test Manager service

- Generate config from inputs
- Can run tests by starting a Kubernetes job inside cluster that runs the MT5 container
- https://github.com/S-A-RB05/TestManager

Result service

- Receive results from jobs
- Save results to MongoDB
- https://github.com/S-A-RB05/ResultService

Links

- All our code are online available in our organization: https://github.com/S-A-RB05
- All of our relevant research can be found here: https://github.com/S-A-RB05/.github/tree/main/Research
- MetaTrader 5: https://www.metatrader5.com/
- Free MongoDB: https://www.mongodb.com/
- Free RabbitMQ: https://www.cloudamqp.com/
- Docker hub: https://hub.docker.com/search?q=stockbrood