**Crypto Convertor**

A simple, not so ambiguous application to convert Cryptocurrency to currency, which is made as an technical assignment.

**How to run application**

Navigate to source code direct, and then navigate to CryptoConvertor folder. You should be able to see the docker-compose.yaml file. Make sure docker is running on your machine, probably on Linux mode. Then run the following command: docker-compose up –build

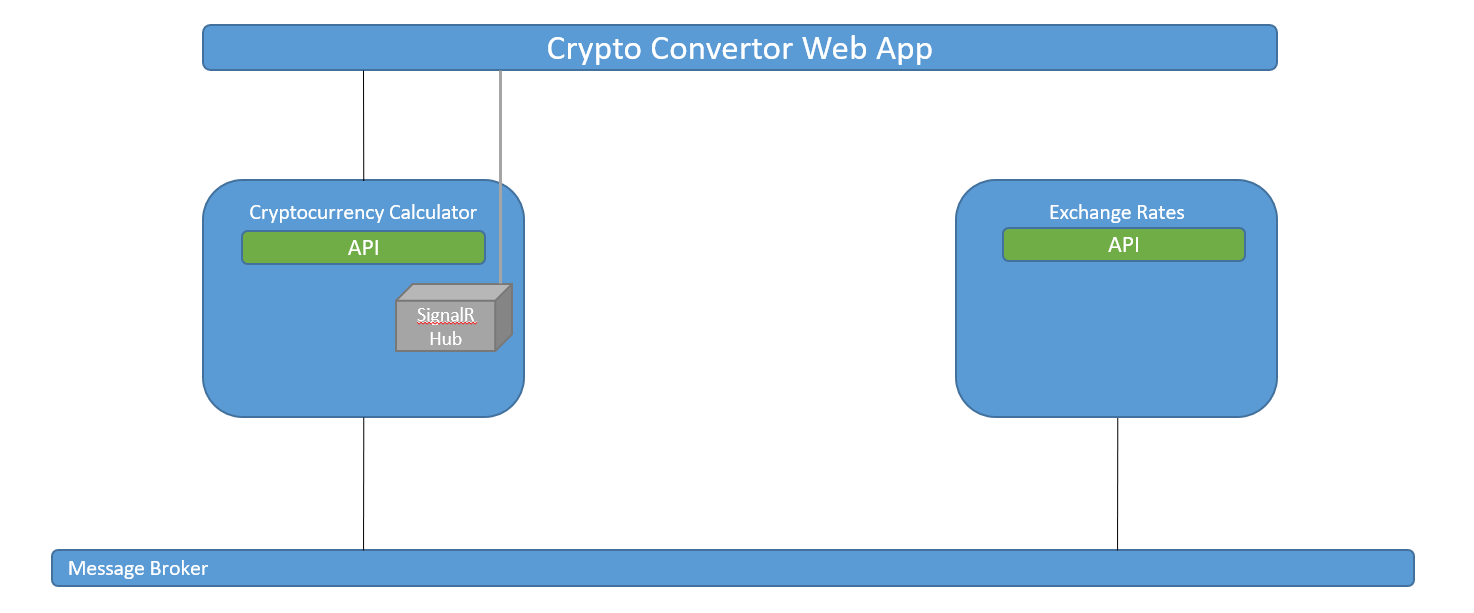
After images downloaded and containers are created, you can navigate to webui using this link: <http://localhost:5012>

**Tech Stack**

* Dotnet Core 2.2
* SignalR
* Angular
* MS Test, Moq
* Rabbitmq
* Docker
* Docker-Compose

**Solution Design**

A simple solution design with using Microservice and Dockerise concepts. I tried to keep it quit simple and minimal.



The solution contains the following projects:

* **Web App**: is a simple UI which triggers the cryptocurrency calculations.
* **Cryptocurrency Calculator**: It gets the exchange rates from Exchange Rates service and calculate the quote for incoming cryptocurrency symbol.
* **Exchange Rate**: An Api which load and distribute the list of exchange rates against base currency, per request from the Message Broker. This project is an Api Core project, but it could be an console app as well.

**Technical Question**

Question 1:

* I have spent round 16 hours on doing this assignment. I also had a look on two training on Microservices and Dockerising .Net application.
* If I had more time,
  + I would have use an API Gateway between UI and services
  + Add more unit testing to cover rabbitmq side, using MassTrannsit Test Frame work
  + Improve exception handling and logging policies
  + I would not store API key in appsetting directly, but read it from vault or CICD pipeline
  + I would like to see in deployed in AWS or at least Azure
  + I would manage users for SignalR, right now it just broadcast to all

Question 2:

* I think the solution I made is quite modern, but I have not used a lot of new-new features, however I used some new c# syntax feature like
  + Expression Bodied Members

public string Uri => $"rabbitmq://{UserName}:{Password}@{HostName}:{Port}";

I also used new docker extension for VS Code which was so pleasant.

Question 3:

* I have had some experiences with performance issues. First thing I try to do, is to identify where is the bottle neck, is it related to application logic, 3rd party component or infrastructure. Logging and monitoring tools would be quite useful to direct you toward the root cause, but some times you need to collect some telemetry data from external service to indicate if they have the issue. For the applications I traced, a lot of time it was Infar and networking issue.

Question 4:

* For this assignment I wanted to have a better understanding of Microservice, docker and also clean architecture, so I did the following course on Pluralsight
  + Modernizing .NET Framework Apps with Docker
  + Clean Architecture: Patterns, Practices, and Principles

I also watched and amazing video on microservice and dotnet core, it was more theory but I liked it and it was presented by a Dutch guy, I guess. <https://www.youtube.com/watch?v=-AfZxdXa7yc>

Question 5:

* I liked this technical assignment, it was small and straight forward. I tried to do it aligned with what Knab needs for the job. It was already beneficial for me because it pushed me to learn new approach, which I found it quit fun but also challenging. I hope I can hear how you see it as well. I also enjoy this technical question, I guess they are quite simple and tricky 😊.

Also the result of the assignment, Quote of BTC to US Dollar, reminds me I should have bought some Bitcoins years ago. It could be a bit more rewarding than coding 😉

Question 6:

* Please me.json in the root directory of the project.