STOCK MARKET DOCUMENTATION

**PROBLEM STATEMENT**

E-StockMarket Application is a Restful Microservice application, where it allows users to manage the stocks like create, view stock price details and company details.

Microservices Used

1. Company
2. Stockprice
3. Authentication
4. Api gateway
5. eureka
6. Kafka consumer Component

Database Used

1. My SQL
2. MongoDB we are using cloud db from mongo db atlas

Micro service Descriptions

1. **Company**

Company service is used to add, delete, search and display company related details. Authentication filter is used to validate every request for. My SQL and Mongo Db is used for storing data.

Apis exposed ::

* /api/v1.0/market/company /all
* /api/v1.0/market/company /register
* /api/v1.0/market/company /delete/{companyCode}
* /api/v1.0/market/company /get/{companyCode}

Swagger url::

http://localhost: 8085/company/swagger-ui.html (through api gateway)

http://localhost: 8090/company/swagger-ui.html (Direct company service call)

1. **Stockprice**

Stock price service is used to add and retrieve stock information. Authentication filter is used to validate every request for. My SQL and Mongo Db is used for storing data.

API’s exposed ::

* /api/v1.0/market/stock /add
* /api/v1.0/market/stock /get/stockExchange
* /api/v1.0/market/stock /get/{companyCode}

<http://localhost:8085/stockprice/swagger-ui.html> (through api gateway)

http://localhost:8091/stockprice/swagger-ui.html (Direct stockprice service call)

1. **Authentication**

Authentication Service is used for user creation and token generation for valid users. Token will be used by the services to validate the user

API’s Exposed::

* /api/v1.0/market/authentication/user/ signup
* /api/v1.0/market/authentication/user/login

Swagger url ::

http://localhost:8085/ authentication-service/swagger-ui.html (through api gateway)

http://localhost: 8092/authentication-service/swagger-ui.html (Direct authentication-service call)

1. **Api gateway**

Api gateway act as the entry point to all our services. All incoming request will be routed to corresponding microservices.

1. **Erureka server**

Eureka server holds the information on the connected microservices.

Eureka endpoint :: http://localhost:8761/

1. **Kafka consumer Component**

Kafka consumer component will consume the messages published from stockprice service and company service and insert the data into mongodb. Which will then be used by the UI.

The company details are published to kafka from the company service and stock price details are published to kafka from the stock price service

Kafka Topic: stockmarket

Note : Attached sql script and refer sql data script file to insert pre defined stock exchange data into stock\_exchange table in db