LAB- Understanding Monolithic Application

In this lab you will be understanding about a monolithic application which we are going to use throughout the course

Before you proceed, make sure that you have installed mysql on your machine. You can use any database if you want to. But you need to modify the connection configurations in all projects.

I suggest you to use Mysql for this course.

Make sure that you also create a database with name micro

In this lab, you will be working on **Way2LearnTraderApp** project

Open pom.xml and observe that we have configured all the dependencies in that xml.

This is a Trading Application – Application will fetch prices of Stocks from Yahoo
There is Rest API provided by Yahoo
Trading Application will be used to buy and sell stocks

MONOLITHIC APPLICATION

This application has 3 services which will be broken down into microservices later so that they are independently deployable and manageable.

Steps

1) Run Way2LearnTrader.java which is in com.way2learnonline package. This also starts embedded Tomcat server.

2) Look at **application.yml** in src\main\resources. Notice the port of application.

3) In this yml , change the mysql database' username and password specific to your database

4) Give a request to http://localhost:6060

5) On the Login Page – Under **Registration :** Click on **Create One** and do registration

6) In mysql – micro database – check account table – you will find entry for user

7) Login with username and password that you have created (Spring security is used for all

this)

8) Portfolio Summary is shown : purchased value of stocks , sold value of stocks etc.

9) For new Account , Portfolio Summary shows no data – only Total Cash available is shown

10) Click on the **Trade** Link on the Above Menu

11) In the Trade Page , Enter any Stock name – lets say **IBM** and press **GET QUOTES**

12) This will hit yahoo service and get IBM's live quotes – High and Low
13) Enter the Quantity to buy and Click **"Buy" .**You will see that **"Order successful"**

14) And **List of Holdings** show IBM stocks are purchased and purchase price etc.

15) Now I want to **sell** 50 stocks of IBM
16) Enter 50 in text box and click on **Sell**
17) And **List of Holdings** show updated IBM stocks , purchase value, gain/loss etc.
18) Now go back to **Portfolio** page , check current value of held shares, Total purchase value , Total Sold value etc.

Code Overview : Way2learnTraderApp

1) UserController :
2) For Registration and Log-in
3) On the Home Page (Without Log-in also) – it displays MarketSummary , notice "/" over method
4) MarketSummaryService is used for the same by UserController
5) UserController is dependent on UserService which has logic for login and logout
6) UserService has logic to create an account , log-in , get Account details based on log-in user and logout

TradeController
7) For Trading, buying stocks etc.

8) It is dependent on MarketService

9) MarketService is using QuoteService

10) QuoteService is hitting yahoo URL to get quotes

11) To this yahoo URL , it is making REST request using RestTemplate

12) QuoteService has method getQuotes – it takes comma separated symbols and makes a RESTful call to the given yahoo URL , gets the Yahoo responses , creates a list of Quote objects
13) MarketService is using QuoteService to get list of all companies and list of quotes.

14) MarketService is also dependent on PortfolioService

15) PortfolioService is responsible for managing user's Porfolio
16) The list of holdings and all user's transactions will be managed by PortfolioService

17) PortfolioController
18) For Displaying Portfolio of log-in User
19) It is dependent on MarketService and MarketSummaryService
20) Gets Portfolio using MarketService which further interacts with PortfolioService for the same