Developer Test Project

The purpose of this project is twofold. One purpose is to test your ability to work independently to solve problems in areas you may not have experience in, without direct support from other developers. The other is to test your ability to work with the basic technologies we are using here. To that end we will mandate certain technologies to use in completing the tasks below.

In addition, this is a timed test, and there may be a tradeoff to be made between completing all of the tasks and writing well designed, clean code. Part of the evaluation will be on how you manage that balance. If you do not feel you have time to complete all tasks you may choose to work on tasks out of order.

Note: BitMEX provides code samples and connectors for their APIs in various languages including Java. You may find some of this code useful as an example, but do **NOT** use their provided code in your submission.

Technologies and Environment

We use the current Spring Boot 2.0 release, with Maven for dependency and build management. You will be expected to use those technologies in your application as well. You may assume that a local MySQL database is available and that it will accept connections as “root” without a password, that Maven 3 will be installed and the mvn command will be available, that Java 8 JDK commands will be available, and that your application will be run on regular Linux user account (not root).

Deliverables

You should email an archive of a project folder containing your Maven project. There should be a README file in the root of your project folder explaining how to run the tasks in your project. Expanding your archive and following the instructions in your README should be sufficient to successfully run each task you have completed.

You have 24 hrs from the time this document was sent to you to respond with your completed submission. If you respond sooner the time saved will be taken into account when evaluating your submission.

Tasks

Complete the following tasks. These tasks will require you to use the testnet services provided by BitMEX at testnet.bitmex.com. You may need to create an account and load it from a Bitcoin Testnet faucet to complete some of the tasks.

Task 1: REST API

Connect to the BitMEX REST API. Query the API for the current bid price for the Bitcoin USD pair and print it to standard out.

Task 2: WebSocket API

Connect to the BitMEX WebSocket API. Print a running stream of bid price updates for the Bitcoin USD pair to standard out until the process is manually killed.

Task 3: Create A Limit Order

Create a limit order to buy 1 contract of the Bitcoin USD that will not fill (set a price far below the current market). Print out the HTTP response status and “orderID” of the resulting order to standard out. You may store the order details in the database if you wish.

Task 4: Lookup Order Details

Using the REST API, lookup the order details of an existing order. Print the HTTP response status and order fields to standard out.

Task 5: Modify An Order

Modify the price of an existing limit order into a range that it will clear. Print out the HTTP response status, and the order details before and the after modification to standard out.

Task 6: Cancel An Order

Cancel an existing limit order. Print out the HTTP response status, and the order details before and after cancellation.

Task 7: Create and Close A Position

Create a new market order. Once it has filled, create a reciprocal market order to close the position opened. Print the final order details of each order to standard out.

Task 8: Stream Order Updates from WebSocket API

Submit a new market order. Using the WebSocket API, store the order details in the database and update the database record when further order updates occur. Print the order details to standard out each time the database record is inserted or updated.