Mobile Engineer Coding Challenge/Take Home Test

Thank you for applying to tastytrade! As you are aware, we have a coding challenge for you. Take your time and spend up to a week to complete this project. We have outlined a few guidelines and tips for you, but feel free to get as creative as you'd like.

Please create a watchlist application:

- The goal of this app is to provide a way for a user to manage watchlists of quotes.
- This application must be written as a native iOS (Swift) or Android (Kotlin) app.
- React Native or any other JavaScript based frameworks are not allowed.
- Open-source libraries are fine to use.

Github

- Your code submission must be committed into a private github project.
- Your github project should include a README.md file that has clear instructions on how to setup and run your project.
- Github Username for submissions: @coreypett

Quote Data Source

Use the IEX Cloud API to retrieve information about stocks. Sign up for an account, they have a free trial plan (you don't need to enter any credit card information).

Some endpoints that are relevant are:

- https://www.iexcloud.io/docs/api/#quote
- https://www.iexcloud.io/docs/api/#batch-requests

Symbol Search Endpoint

https://api.tastyworks.com/symbols/search/<Search String>

Example:

https://api.tastyworks.com/symbols/search/GOO

Application Requirements

1. When the app first starts up, a Watchlist screen will be presented.
    a. A default watchlist should be created for the user
        - The default watchlist's name should be "My first list"
        - The default watchlist should have the following symbols: AAPL, MSFT, GOOG
2. The Watchlist screen should consisting of the following elements:
    a. A way to switch watchlists
    b. The watchlist table composed of the following columns:

        - Stock Symbol
        - Bid Price
        - Ask Price
        - Last Price

    c. A way to modify a watchlist

        - Add a symbol to an existing watchlist
        - Remove a symbol from an existing watchlist
        - Create a new watchlist with a given name
        - Delete a watchlist

3. Watchlists should be persisted locally.
4. When the user is adding a symbol to a watchlist, they should be presented with a symbol search screen to allow for auto-completion of a symbol.
5. The application should refresh the quotes on the active watchlist every 5 seconds.
6. When a user selects a symbol from a watchlist it should bring up a new view with that displays the Stock Symbol, Bid, Ask, and Last Price As well as a chart.

    a. The chart should show the last 30 days of the symbol’s history.

        - The stock price history chart should render a candlestick for each day
        - You can retrieve the last 30 days of data from IEX's chart api.
        - Feel free to use open source code to retrieve and display this data.

    b. The quote prices should update every 5 seconds as well.

Extra Credit - Create your own Symbol Search Server

If you want to provide an example of your server-side coding ability, implement your own symbol search server. Use whatever language you like.

You can get a CSV of listed stocks for each exchange from here:

http://www.nasdaq.com/screening/company-list.aspx

You must provide instructions on how to build and start your server locally and how to configure your mobile application to use a different search endpoint.

Good Luck!