

**Task:** Create a single-page application using NodeJS for the webserver and ReactJS for the front ui that will interact with the Coinbase Pro websocket API (https://docs.cdp.coinbase.com/exchange/websocket-feed/overview).

A user should be able to load your website and view a page that will allow them to interact with any of the following 4 products:

* BTC-USD
* ETH-USD
* XRP-USD
* LTC-USD

The webpage should consist of 4 components: Subscribe/Unsubscribe, Price View, Match View, and System Status.

**Subscribe/Unsubscribe**

This section is only for user interaction and should allow the user to subscribe or unsubscribe to a product, as well as display whether each product is subscribed or unsubscribed. Subscribing to a product would equate to sending a subscribe message on the API for the product and the *level2* and *matches*channels. An unsubscribe would equate to sending on unsubscribe to both channels for that product. For this project, do not worry about missed messages on the *match*channel so subscribing to *heartbeats* is not required.

**Price View**

Your application will need to maintain the level2 updates that are received from the websocket. This section should display the bids and asks for the subscribed products at a refresh rate of 50ms. Do not worry about displaying them in a specific way, as long as it is reasonable.

**Match View**

This section will be an order blotter that will display the timestamp, product, trade size, and price (use green text for buy and red text for sell) of each match that comes over the *matches*channel. You do not need to keep infinite scrolling on this trading blotter, use your best judgment for how much to display without impacting performance, but ensure that you always display the most recent match.

**System Status**

This section will display a list of all channels that the application is subscribed to (this information is provided to you as a response to a *subscribe* message). The displayed channels should be everything, and not limited to what the user has asked for.

**Miscellaneous**

You will need to be able to support multiple users, where each user can be subscribed to different currencies. A user should only see updates for the pairs that they are subscribed to.

The webserver must handle all incoming prices updates and then send them to the right subscribed user.

There are no design restrictions on the page, but you are also not expected to build something flashy and unique, just functional.

**Criteria**

We will be reviewing your coding style, logic, ability to handle updates, ease of use of the web page, and robustness from possible user error.