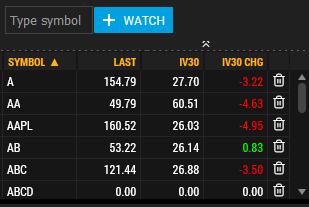
**Programming exercise**

Create a simple web application that retrieves market data using REST Web API and displays data in the browser.

UI may look like this:



User enters the stock symbol name, clicks “Watch”, the symbol is added to a watchlist grid below.

Application then calls the Web API asynchronously and updates information in the browser every 10 seconds.

Clicking the Trash icon removes the symbol from watchlist and updates for this symbol should not be requested from Web API anymore.

Make sure that the user entered a valid symbol (related Web API method will be described below).

You do not have to implement column sorting and conditional coloring that are present in the above image.

Use Django or Flask for server side and any of React/Agular/Vue for front-end side.

**API access**

You will need to use CBOE Livevol All Access APIs for your exercise.

Please review below link for more information https://www.livevol.com/apis/technical-reference/

Easy way to start is to review usage Examples, Response Codes, Authentication section,

then two methods you will need to use:

* Reference API – Symbols /allaccess/reference/symbols

Should be used for symbol validation

* Market at a Glance API – Underlying quotes /allaccess/market/underlying-quotes

Should be used to retrieve market data (symbol, implied\_underlying\_mid, iv30, iv30\_change are the fields you need to implement UI)

Valid access token should be provided with each Web API call. To obtain the token please use below credentials:

Client id: aaa\_client\_for\_test\_exercise

Client secret: Jfjbi6w!

**Additional comments**

* Access token lifetime is 4 hours, so you do not have to request it before each API call, instead retrieve it once and cache for the duration of the session.
* You have to use **delayed** endpoint for the purpose of this exercise.
* Your Web API account has usage limits, so if you call Web API too often, your requests will be throttled, and you will receive 429 HTTP error
* Note, every method page contains a link to “Try It” widget that may be used to try API calls without any coding, you just need to provide above mentioned credentials.