

👉 Introduction

You are required to develop a Widget for one of our clients. This widget will be used to display several currency exchanges rates and it will allow the user to exchange USD (Note #1) for Crypto (Note #2). The application consists of a React front-end client and a Node back-end service (**NestJS**).

☰ Criteria

The application must meet all the following criteria:

- Front-end must have a toolbar ([design here](#)) with two dropdowns and two input fields where you can pick the currency that you want to exchange from and the Crypto that you want to exchange to.
- Front-end must have a table component ([design here](#)) to display historical data (Note #3).
- Back-end service must consume the rates from the API (Note #4) every few minutes (configurable number), store the data in the Database (Note #5) and stream the data to the front-end using WebSockets.
- Latest Node and React versions must be used.
- You need to upload the repository to GitHub and share the link.
- You need to create a short video to demo the application in a working state.
- Assessment will be rejected if no readme file is provided with instructions on how to run the project.

★ Bonus points

- Dockerize the application.
- Documentation of the code with comments.
- Unit-tests.
- Add a [Date Picker](#) filter (any Date Picker can be used) on the table to filter data by time.
- Add sorting for all columns on the table.
- Mobile view must be exactly as the design provided ([design here](#)).

Notes

- 1.** USD is mandatory. EUR, GBP and other fiat currencies are optional.
- 2.** BTC and ETH are mandatory. Other Crypto are optional.
- 3.** Table fields are "Date & Time", "Currency from", "Amount 1", "Currency to", "Amount 2", "Type".
 - a. There are two types of data, "Live Price" and "Exchanged"
 - b. "Live Price" indicates the rate coming from the API and it is the currency conversion of any currency to USD. For example, the live price of 1 BTC is 44000 USD.
 - c. "Exchanged" indicates the transaction happened when user clicked "SAVE" button. For example, the user exchanged 44000 USD to get 1 BTC.
- 4.** Any API can be used or even a mock server that returns fake data. Possible solutions include [CurrencyAPI](#), [CoinLayer](#), [CoinAPI](#) and [Coingecko](#).
- 5.** You can use either Mongo or MySQL.

Good luck! 😊