# Work@ Membrane

SENIOR BACKEND ENGINEER

## Market Status API REST

The goal of this project is to create a public API REST that retrieves market information for trading pairs.

Specifications:

- Use Express framework to set up the server.
- The API should expose two endpoints:
  - One that receives a pair name, and retrieves the tips of the orderbook (i.e. the better prices for bid-ask). Response should include both the total amounts and prices.
  - Other endpoint that is called with the pair name, the operation type (buy/sell) and the amount to be traded. Should return the effective price that will result if the order is executed (i.e. evaluate Market Depth).
- API should return market values for the following pairs: BTC-USD and ETH-USD. We expect to handle unexpected pairs.

This engine must be written in Node.js and it must use websockets, without persistent storage. It should also support a HTTP interface to fetch the endpoints.

The backend should consume data from an external exchange. You could use the orderbook websocket stream from Bittrex API or Bitfinex API.

Submit your solution to a private github repository. Keep the organization of branches and commits as if you were working in a team. Give access to the following list of github users:

- **fedecaccia**
- **francomangussi**
- **mainqueg**
- **ramirocarra**
- **Pablofedericosantillan**
- **Fabenedetti**
- **Oter2901**
- **gonza4**

**Bonus:**

- In the second endpoint, include a parameter to set a limit for the effective price and retrieves the maximum order size that could be executed.
- Create a set of unit tests for the logic used in the Market Depth implementation.

# Work@ Membrane

SENIOR BACKEND ENGINEER

## Evaluation

- Show us how you create clean, maintainable code.
- Focus your solution on performance and scalability.
- Make the code aesthetic, clean code (use linter)
- Senseful folder structure
- Scalable solution
- Error handling and logging

Upon completion, send a .zip file **without node_modules** to your recruiter
Include any env vars required for running the project and some pictures of it working