Nooklyn Interview Challenge
Engineer Internship 2018

Part 1

We'd like to see a very simple React application that utilizes the API [described below] to render a two-pane layout that displays the list of NYC subway trips on one side and a map (Google, Apple Maps, MapBox, etc) on the other. Clicking on a subway trip should use the map to display the various stops that were made using the provided location and arrival time.

We've setup an API that is backed by data normalized from the MTA's real-time API for their subway countdown clocks. There are two API endpoints, one that will display a list of trips and another that will show the arrival times for that trip at each stop.

[LIST OF TRIPS]

https://nooklyn-interview.herokuapp.com/trips

[EXAMPLE: LIST OF F-TRAIN TRIPS]

https://nooklyn-interview.herokuapp.com/trips?filter[route]=F

[EXAMPLE: ARRIVALS FOR TRIP ID: 41]

https://nooklyn-interview.herokuapp.com/trips/41/arrivals

This API is structured according to the JSON API standard, which you can reference here: http://jsonapi.org . This API format makes traversal of the API easy by following the generated URLs in the response.

[EXAMPLE: TRIP PAGINATION]


```
"links": {
    "first": "https://nooklyn-interview.herokuapp.com/trips?
    filter%5Broute%5D=F&page%5Bnumber%5D=1&page%5Bsize%5D=20",
    "next": "https://nooklyn-interview.herokuapp.com/trips?
    filter%5Broute%5D=F&page%5Bnumber%5D=2&page%5Bsize%5D=20",
    "last": "https://nooklyn-interview.herokuapp.com/trips?
    filter%5Broute%5D=F&page%5Bnumber%5D=11&page%5Bsize%5D=20"
}
```


[EXAMPLE: TRIP'S ASSOCIATED ARRIVAL DATA]

```
"relationships": {
    "arrivals": {
        "links": {
            "self": "https://nooklyn-
            interview.herokuapp.com/trips/191/relationships/
            arrivals",
            "related": "https://nooklyn-
            interview.herokuapp.com/trips/191/arrivals"
        }
    }
}
```


The deliverable of this project will be a git repository and can be shared via a private Github repository. The README should provide any instructions necessary to configure and run the application locally. You may also include a link to a live-running version of your submission using Heroku.

You're free to use any public, open-source libraries you wish. You can consult Stackoverflow or any documentation as necessary. There's no time-limit or deadline on this project. Your time is valuable and we don't want it to be a burden or impose on other obligations you have.

Part 2

[The following is a bonus question on encryption / network security. Please answer it to the best of your ability without using external resources.]

Alice and Bob wish to send messages to each other using a symmetric encryption algorithm but have not agreed upon a shared key. They decide to use the Diffie-Hellman key exchange. A brief explanation of this protocol is explained below.

1) Alice and Bob agree to two prime numbers, **g** and **p** and share these publicly over the network.

2) Alice and Bob each select secret numbers denoted by **a** and **b** respectively.

3) Alice computes **A**, denoted by the equation $A = g^a \mod p$, and sends it to Bob.

4) Bob computes **B**, denoted by the equation $B = g^b \mod p$, and sends it to Alice.

5) Alice computes the shared key **S**, by computing the equation $S = B^a \mod p$. Likewise, Bob computes the shared key **S**, by computing the equation $S = A^b \mod p$.

6) Now, Alice and B have a shared key **S** they can use as an input to a symmetric encryption algorithm.

Please explain how encrypted messages between Alice and Bob can be compromised. How could you prevent this from happening?