



BAO Front End Challenge

Introduction

Hi there! 🎉

Congratulations on making it this far. We really appreciate the time you have taken to participate in our interviews thus far. This front-end challenge should be completed in **2 hours** but you will have **7 days** to turn in your submission.

Task Description

The goal of this challenge is to test your front-end skills by building a simple React UI for a news web application. The application has two pages:

1. A main page that shows a list of news items.
2. A second (admin) page for loading news data from data sources.

Main Page

This is the page that is displayed to the user when the app **launches**.

1. If news data has not been loaded, then news list should be **empty**, with a prompt to **load** news data.
2. If there is news data, this page should **list** all available news items.
3. The news item list displays only the **title** of the news story.
4. The news items should be sorted by **score** (details below).
5. **Bonus:** Clicking on the news story should show the details as a popup/dialog.

Admin Page

This is the page for loading news items.

1. There will be two sources where data can be pulled from:
 - a. an API (see technical details)
 - b. a csv file (see technical details)
2. Once the data is loaded, it should be available on the main page.

Technical details

Admin Page

API

1. We would use HackerNews as our API source ([Documentation](#))
2. The `/v0/topstories/` endpoint provides a list of ids of the top 500 stories. Sample response:

```
[
  34617844,
  34624795,
  34623007,
  34623604,
  ...
]
```

3. Individual news items are available at `/v0/item/<id>.json`. For example, story `34617844` is available here and the response is:

```
{
  "by": "kgwgk",
  "descendants": 256,
  "id": 34617844,
  "kids": [
    34618087,
    34618030,
    ...
  ],
  "score": 160,
  "time": 1675285994,
  "title": "Meta Reports Fourth Quarter and Full Year 2022 Results",
  "type": "story",
  "url": "https://investor.fb.com/investor-news/press-release-details/2023/Meta-Reports-Fourth-Quarter-and-Full-Year-2022-Results/default.aspx"
}
```

4. Task 1:

- a. Provide a button on the admin page for loading data from the API.
- b. When this button is clicked, make an API call to get the ids of the 500 top stories, then make follow-up requests to get the details of the top stories.
- c. **Bonus:** Provide a notification when the data load is done. See [react-hot-toast](#) for example.

CSV

1. Extra news items are provided in csv format in the source code `public` folder (`news.csv`).
2. Task 2:
 - a. Provide a second button on the admin page for loading data from this csv file.
 - b. When the button is clicked, load the data from the csv file and parse it into a json structure.

Notes:

1. You can use any library of your choice for making the API requests.
2. You can use any library of your choice to parse the csv. You can have a look at [papaparse](#).
3. If the 2 data source buttons are clicked, ensure to combine the data into one array.

Main Page

1. The main page should show a list of news stories after the data has been loaded.
2. The items should be sorted by `score` in descending order.
3. **Bonus:** Once implemented, there will be a lot of items on the main page. Does this impact the responsiveness of the page? If so, what can you do to improve responsiveness?
4. **Bonus:** When a list item is clicked, display only the following fields in your popup:

- a. Title (`title`)
- b. Author (`by`)
- c. Source (`url`)
- d. Popularity (`score`)

Getting Started

1. Fork the [bao-ap-react-take-home](#) repo.
2. Run `npm install` to install dependencies.
3. Run `npm start` to start the application.
4. Make your changes as necessary.

Considerations

1. Your application should include a header bar for switching between the Main page and the Admin page.
2. Please use any UI library ([material-ui](#), [antd](#), etc.) of your choice or plain CSS (or both).
3. The application state does not need to be persistent across page reloads.
4. Proper error handling and basic unit tests will be rewarded.
5. All recommended libraries are installed but you can use alternatives if you prefer.
6. Bonus questions are optional. Attempt them if you wish and only after all other requirements have been completed.

Submitting

1. Push all your changes to your forked repo and ensure the repo is public.
2. Send an email with the link to your repo.

If you have any technical questions, please send an email to jimiolaniyan@baosystems.com.