

NBA Fantasy Projector Journal

Santiago Tobon & Mack Cooper

What Worked

- We were able to retrieve the information needed for the projection from the NBA api.
- The program allows the user to input an player in the NBA and add them to their roster.
- Auto completes when the user inputs the player.
- The program projects the total stat line that the team will have over the course of the number of games that are inputted by the user.

What didn't Work

- Doesn't finish fetching from the api when in the browser with security restrictions.
To get this working you must remove the security restrictions.
- We had to last minute scrape the date range because we still couldn't get information about each players schedule from the api. Instead the user inputs the number of games each player is going to play over the period of time.
- We can't deploy the webpage because of the cors restrictions on the api.

What we had to install:

- node js
- using node package manager to install:
- jquery
- NBA
- node-sass
- react
- react-dom
- react-scripts
- sass

What issues did we run into:

- Trouble fetching data from the NBA api in the terminal and browser.
- Getting undefined variables from the api because it returned a promise.
- Api documentation wasn't clear as to what some of the functions in the api did, making it more challenging to figure out how to fetch properly.
- We tried using a class to manage everything in react but had to change to a more state based react model.
- Trouble getting the fg percentage correct in the projection.

How we fixed our issues:

- We fixed the undefined issue from the promises by utilizing callbacks that copied the info from the json object returned in the promise.

-We fixed the issue with fetching in the terminal by reducing the number of times we tried to retrieve the data from the api. We did this by reducing the number functions we had by retrieving once and grabbing all of the data we needed in one function.

-We fixed the fetch issue in the browser by removing the security restrictions on the browser. This was because of the cors restrictions it was holding up the fetch to the api.

What we would do differently:

-Use a different api or switch from a project related to the NBA. This is because there aren't very many NBA APIs that work well because of restrictions imposed by the NBA.

-We would have used react bootstrap because it would have simplified parts of the project for us. Unfortunately it was too late for us to incorporate it by the time we thought of it.

-Better manage our time so we could get to other functionalities.

Link to GitHub repo

<https://github.com/mackkcooper/NBA-Fantasy-Projector>

Journal Log of our process

Jan 28

-Added favicons and linked them in index.html.

-Using CSS skeleton as our CSS framework for the project.

Jan 29

-Updated .gitignore

-Implemented visual outline of the web page such as the headers and buttons.

Feb 22

-Met up together to do some pair programming.

-Made the decision to switch over to Bootstrap for CSS instead of Skeleton.

-Added react template to our project.

-Added the ability for the user to input a player to their roster.

-We also had to modularize our js code since we originally planned on using one big class but realized that would be problematic.

Feb 25

-Started working on the README, but need to wait until the program is done to add tests and deployment process to the README.

Mar 2-6

-Met up together again to do pair programming.

-Started working on functions to retrieve data from the NBA api.

-Trouble understanding the api documentation, spent time missing around with functions to better understand it.

- Worked on player search, roster output and removal of players in react.
- Set up map for future use with players
- Added html code for the date range component.

Mar 7

- Continued to work on functions for NBA api.
- Ran into issue with trying to return the information that was gathered from a promise that was returned from the api.
- Solution was to pass in a helper function to copy the information over into a json object.

Mar 9

- Worked more on the date range, being able to update the dates correctly and handle error checking if the dates were off.
- Worked on the Projector functionality, such as being able to remove players, search and add players to the map of players.
- Issue where api fetch inconsistent in the fetch, sometimes working other times not.

Mar 10-11

- Fixed api fetch issue by reducing the number of times we send a request to the api.
- Did this by compacting multiple functions that all use the same request function into one function call.
- Api fetch still fails when testing in the browser.

Mar 12-13

- Switched to user inputting the number of games their players play for the projection.
- This being because of the issue with getting the dates from the api.
- Fixed api fetch issue by removing security restrictions from the browser because of a cors restriction in place with the NBA api.
- This means we can't deploy the website because this problem would occur otherwise.