Statlete Project Plan

* Graph sections should be hover-able to display section's value.
* Graph sections should be clickable to toggle section's value on and off.
* Graph sections hover and click states should be distinguishable from each other.
* App should include at least two types of graphs (bar, pie, line, etc).
* App may use an external charting package.

[React highcharts](•%09App%20should%20include%20at%20least%20two%20types%20of%20graphs%20(bar,%20pie,%20line,%20etc).) should help me cover the majority of the above. Worth noting that pulling large amounts of statistical data can often take a bit, do I need to use a loading indicator? What does the draw animation look like when rendering the chart?

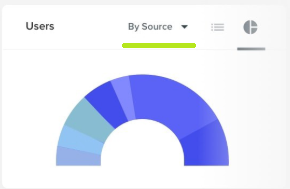
The representation of the data and what type of chart I use will be highly contextual to the sport of the athlete and the stat I’m displaying. For example if I’m comparing the points of an NBA player every year, a bar chart would likely be good. Though if I want to yearly scoring compared against career points a pie chart might actually be better.

* Page should contain a method to switch between the types of graphs.

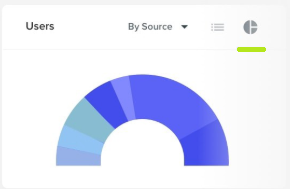
What’s good usability here?  
  
I could use a dropdown (2 clicks) to load up graph options by doing something like:

**Thoughts:** the user knows what they are explicitly selecting, they could also search graph types if there are a lot. Might play a little nicer from a screen reader perspective

Cons: higher amount of resistance just to switch a graph



I could use icons of the graph type to allow them to switch graphs (1 click)



**Thoughts:** Somewhat explicit and visual, could take up quite a bit of space with many graphs

I could use arrows (1 click) to change graph types

**Thoughts:** Only 2 options, allows them to go back and forth. Not explicit in what’s next. Might be a bit clunky on mobile devices.

* Page should automatically load the graph data on project load.

Probably want to pull the data on [componentDidMount](https://reactjs.org/docs/react-component.html#componentdidmount) React lifecycle event. See note about loading indicator above. Maybe there is a way, I can cache some of the data even when they refresh/logout so that they see stats instantly when coming back to the dashboard. May be out of scope for this work sample, though it would be a lot better user experience to only have to “visibly” pull the data once and then update it as needed.

* Application should import graph data from a third-party API of your choice.

This one for

Using ESPN’s would probably be the best option, but it looks like it’s shut down now.. <http://www.espn.com/apis/devcenter/overview.html>

This seems like the most solid bet thus far:

<https://sportsdata.io/developers/getting-started>

This one looks pretty good overall: <https://rapidapi.com/api-sports/api/api-nba>

Although it would be nice if I could find a free API with many player stats for many sports so I don’t have to deal with multiple requests and variant response data.

Given the scope of a work sample though, I will probably make the choice to only go with one sport.

Here are some other options:

<https://www.thesportsdb.com/api.php>

<https://www.mysportsfeeds.com/login>

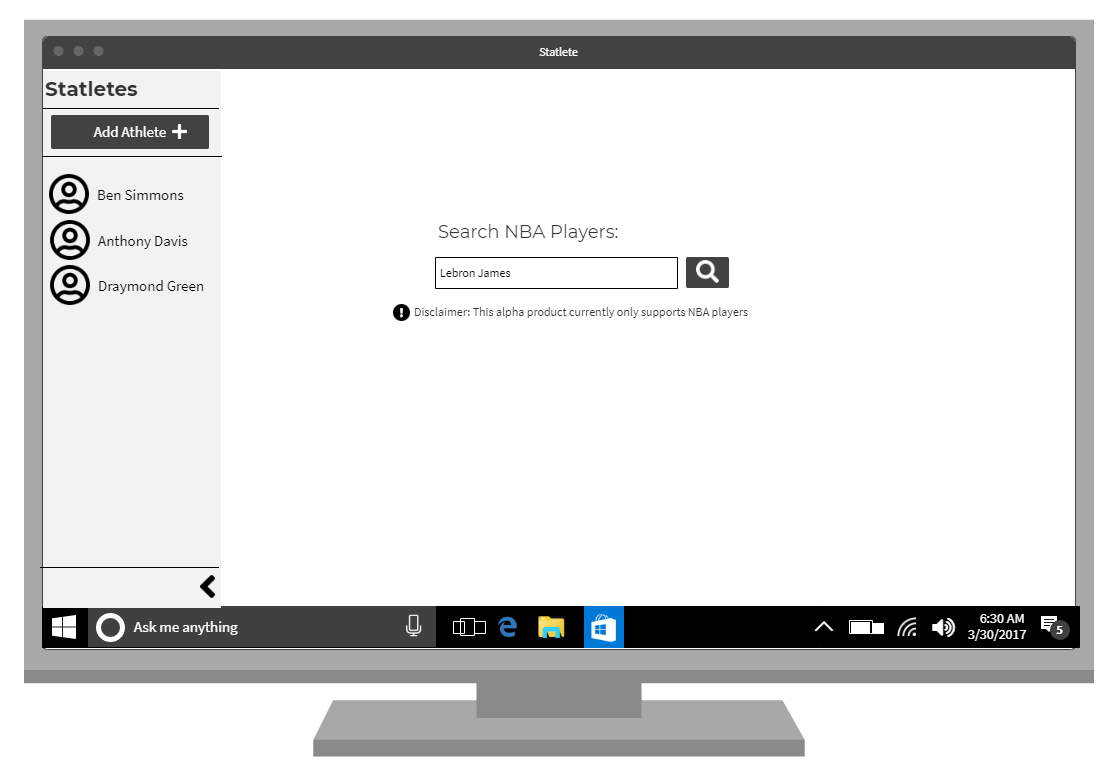
<https://developer.sportradar.com/member/register>

[https://fantasydata.com/api/api-documentation/nba#](https://fantasydata.com/api/api-documentation/nba)

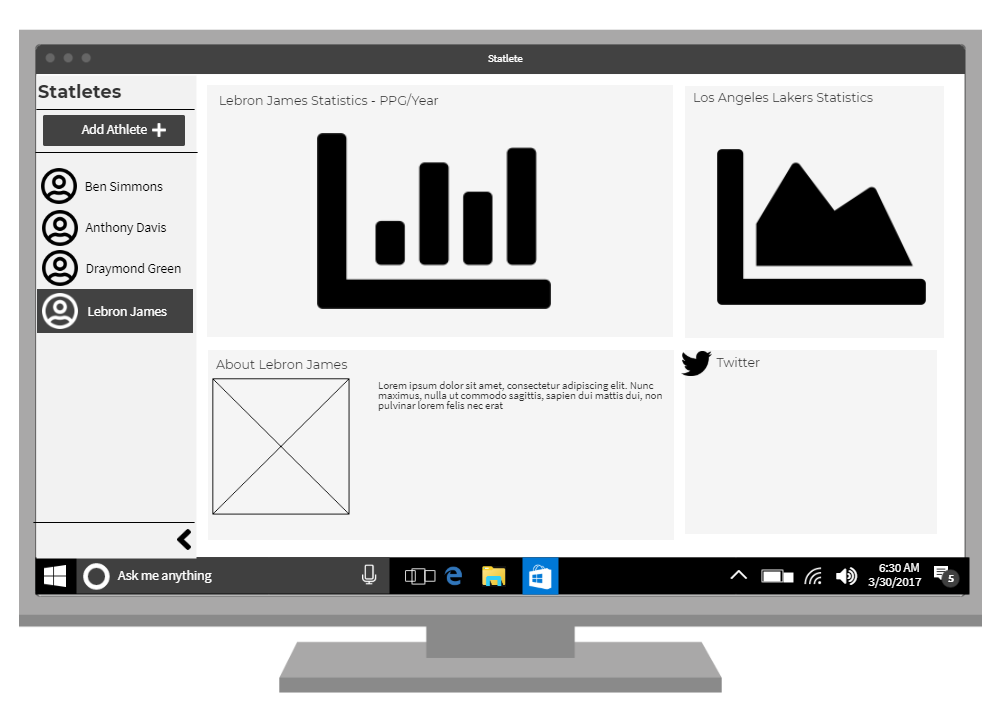
**Update:** The above API’s didn’t allow a great means of searching for athlete data via a name search. Also, they have rate/query limits. Which would make me have to pull all data for every player just to get their player id (this takes too long). Accordingly I found the NBA api that’s free below and allows me to search by player name:  
<https://ball-dont-lie.herokuapp.com/#teams>

Low-Fi Wireframe Concepts:

**Screen 1:**



**Screen 2:**

****

**Technical Bulletpoints**

* Make Side Nav collapsible and have it only show player pictures on mobile. Pictures go to left and “Add Athlete” button goes to the right
* We are only using a single sport and league for the scope of the work sample since free quality API prospects are limited and the data can vary quite a bit between sports even though there are shared entities. Still going to try and build out the app so it isn’t built specifically for NBA and basketball though, so adding other sports and leagues in the future isn’t a massive lift.
* Search won’t have autocomplete since that is too difficult to constrain down to only NBA players if using a search service like google. If we use the api to pull back the full list of players it might be achievable but add extra bloat to the load. Just going to go with a static search button + enter on this one.
* If possible it would be nice to at least persist the athlete names on refresh using browser local storage

For athlete view here is how we should be able to fetch the data for the corresponding information squares

1. Player Statistics – <https://ball-dont-lie.herokuapp.com/#teams>
2. Team Statistics – <https://ball-dont-lie.herokuapp.com/#teams>
3. About Info - <https://developers.google.com/knowledge-graph/>
4. Twitter - <https://developer.twitter.com/en.html>
   1. Still thinking through if I want to show their personal tweets or mentions/trending information. Likely going to go with whatever is easiest to find based on their name.