ASSIGNMENT 1 – REACT APP

Danny O'Leary

20067817

Assignment 1 – React App – Premier League Application

Github link: https://github.com/dannyoleary1/Premier-League-App

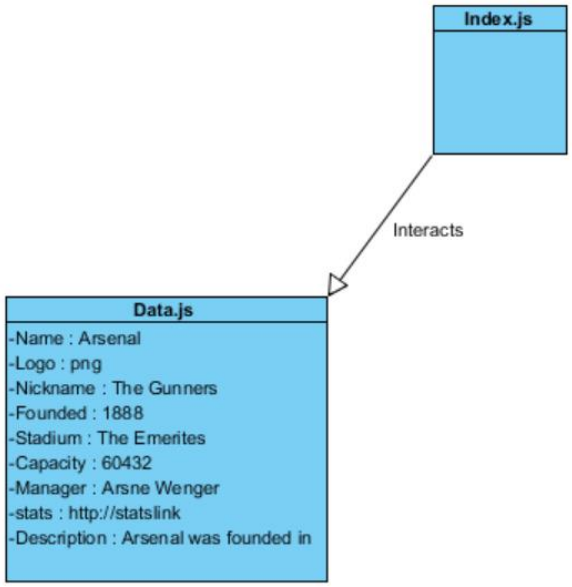
Overview

The aim of this React app is to demonstrate core react functionality to be able to produce a single page application on the Barclays Premier League. This will include a number of different features using React technologies, JSX and other different technologies. Some of these features will include React Routing to be able to use different pages without the need for re-rendering any of them. It will allow us to create links to different pages where more information will be displayed. A user will be able to access the site and get a list of all the different teams that are currently playing in the premier league. This information will be stored on the teams route and will give a brief description of the team. If they are looking for some more information on the teams, they can click into the team which will bring them to teams/name where name is the name of the team. On this page, it will contain information about the different teams in more detail. It will contain a widget that will display information on the different statistics that the team currently have this year. Another widget will be used that will show the different news stories for that team and when clicked will redirect a user to the story itself. As well as that on the table route, it will contain information about the current standings that are happening in the Barclay's Premier League. This is used from an external API called: http://soccer.sportsopendata.net/ . This will implement Reactable to make a table to display all the information which will be fully searchable and fully sortable on whichever field that you want to sort. The app in itself will use extensive hyperlinking so that the pages are easy to find, it will have multiple routes (20+, one for each team). I aims good styling and also implements a json server which stores the information on the web application (However it doesn't currently use that for its data set, instead uses the external data set and a data set that I have made myself). It is a single page application so it is tuned for performance and should run very quickly. The app is secure and features cannot be accessed without a login. Some of the independent learning that was done on this application would be:

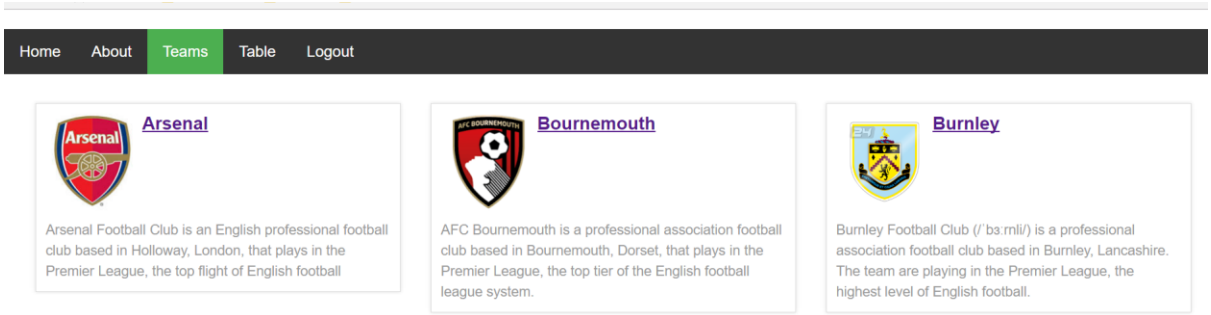
- Auth0 – Using this to store login information securely. This provides authentication for the app and was something I implemented myself.
- External API's – I used many different technologies to get an external API to work. I used componentDidMount which is a React Dom technology that allows a component to be rendered when the component is mounted and then it doesn't need to be used again. I used jquery to actually make the API call. I called to json data and was able to convert it into an array of data to use with a table.
- Reactable and the table. I used this technology to make it easy to use a table with the data from the external API.

Data Model Design

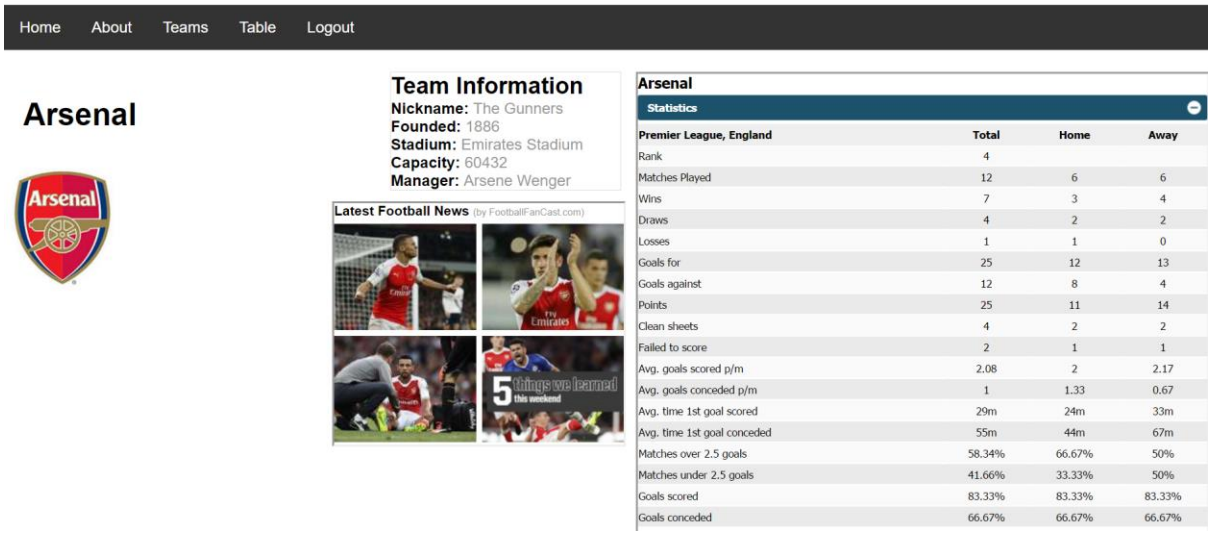
Data Model 1



This data is then used on the teams page to display this:

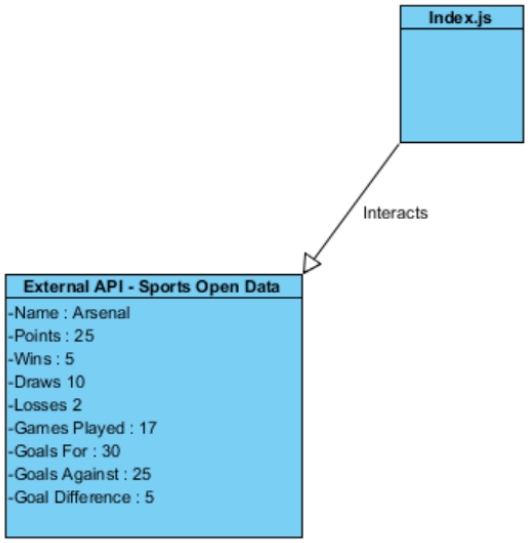


This uses the logo, name and description. It's also used on the individual team pages e.g. teams/Arsenal displays:



This example uses a lot of the different data such as the team info which uses the fields: nickname, founded, stadium, capacity, manager. It also uses the logo and stats.

Data Model 2

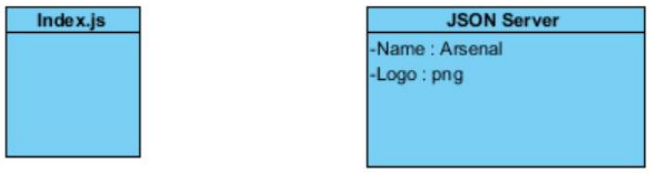


This is an external API call using jquery to be able to display the up to date data about a team. This is updated weekly so it means that I wouldn't have to. An example of how it looks:

TEAM	GP	W	D	L	GF	GA	GD	PTS
Liverpool	11	8	2	1	30	14	16	26
Chelsea	11	8	1	2	26	9	17	25
Man City	11	7	3	1	25	10	15	24
Arsenal	11	7	3	1	24	11	13	24
Tottenham	11	5	6	0	15	6	9	21
Man United	11	5	3	3	16	13	3	18
Everton	11	5	3	3	15	13	2	18
Watford	11	4	3	4	15	19	-4	15
Burnley	11	4	2	5	11	15	-4	14
Southampton	11	3	4	4	12	12	0	13
West Bromwich	11	3	4	4	12	15	-3	13
Stoke City	11	3	4	4	13	18	-5	13
Bournemouth	11	3	3	5	13	16	-3	12
Leicester City	11	3	3	5	13	18	-5	12
Middlesbrough	11	2	5	4	10	12	-2	11
Crystal Palace	11	3	2	6	16	19	-3	11
West Ham	11	3	2	6	11	20	-9	11
Hull City	11	3	1	7	10	24	-14	10
Swansea City	11	1	2	8	10	21	-11	5
Sunderland	11	1	2	8	9	21	-12	5

All of this data in the table rows is retrieved from the API.

Data Model 3



This is the JSON Server that I set up but it isn't connecting to the index.js because the index is using the data model used in data model 1. In a real scenario it would connect to the JSON server to update the data model on the fly. It would use post requests to update and get requests to return the data.

UI Design

Home Page

Home | About | Teams | Table | Logout

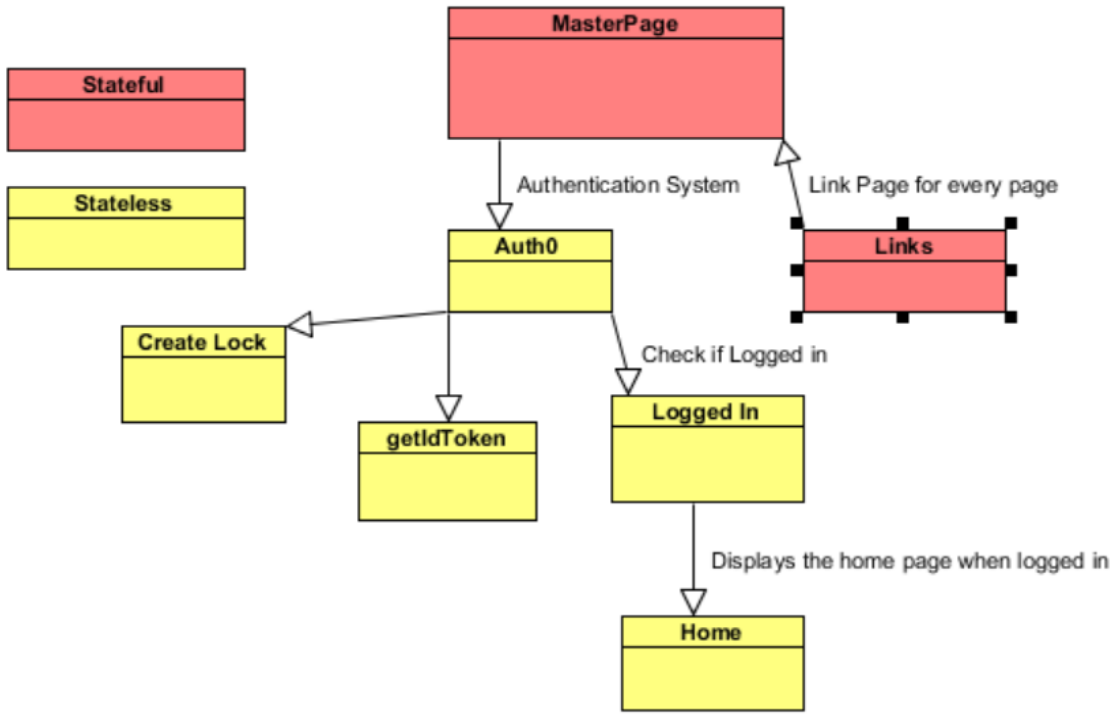
Welcome to Danny's Premier League App

Click on the button below to sign in or sign up to unlock features

Sign In

The URL Pattern: http://localhost:3000

The URL Pattern: http://localhost:3000/home



Text Form

Links

MasterPage

|Auth0

||CreateLock

||getIdToken

||LoggedIn

|||Home

MasterPage State: {lock, idToken, authHash, }

Links:{ActivePage}

AboutPage

Home | About | Teams | Table | Logout

About

The aim of this app is to give information on the different aspects of the Barclays Premier League. It is broken down into different sections: teams, individual teams, and also a table. This about page will also show the different technologies that I used to make this app.

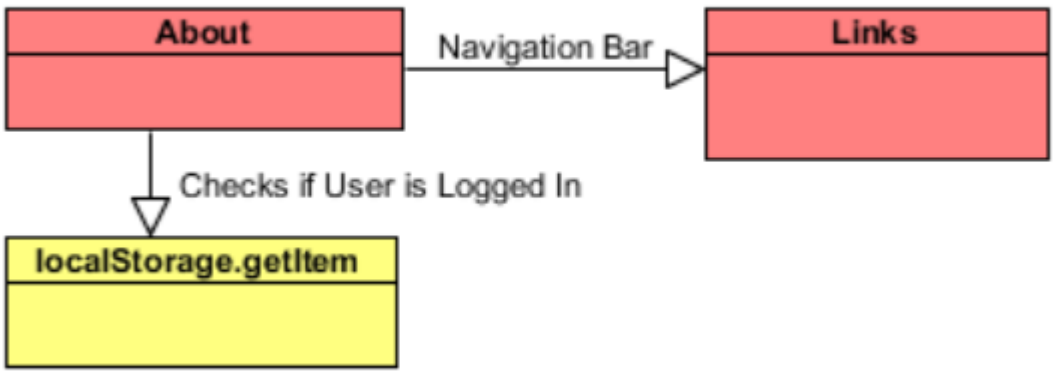
Teams

Teams will be the section where it will contain all the different teams in the Premier League. It will have a hyperlink to their page which will have more information about the team. It will also give a brief description of the club in question.

Teams Individual Pages

The individual page will contain more detailed information and give different stats on the teams. It will contain a a news feed for each team which will link to stories about that team.

The URL Pattern: http://localhost:3000/about



Text Form

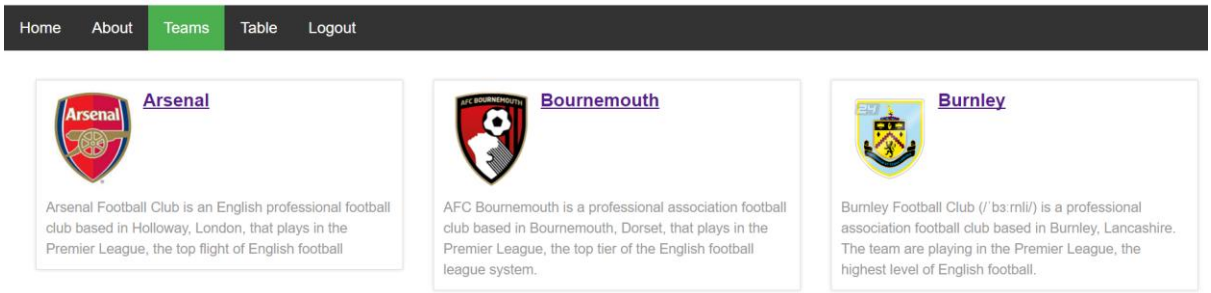
Links

About

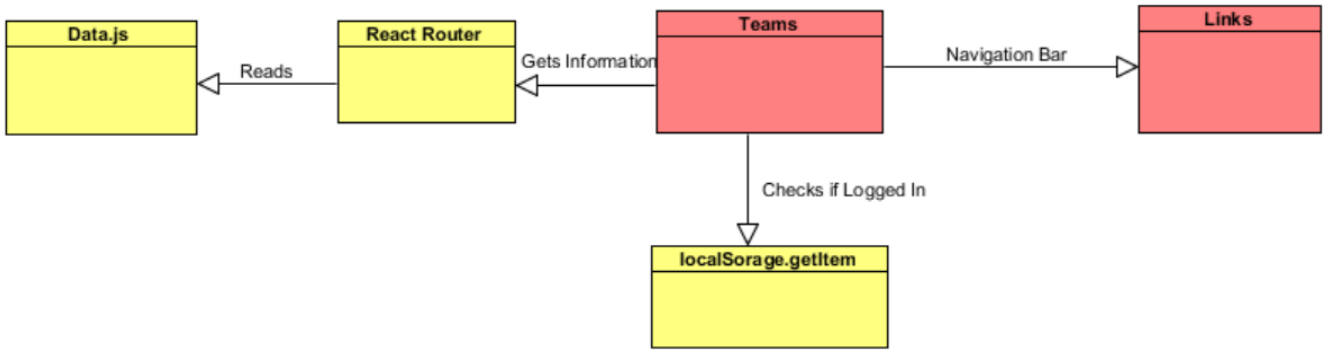
|localStorage.getItem

Links: {ActivePage}

Teams Page



The URL Pattern: http://localhost:3000/teams



Text Form:

Links

Teams
|localStorage.getItem

|React Router

||Data.js

Teams: {Teams from data.js, default link, namesList(what information is being displayed)}

Links: {ActivePage}

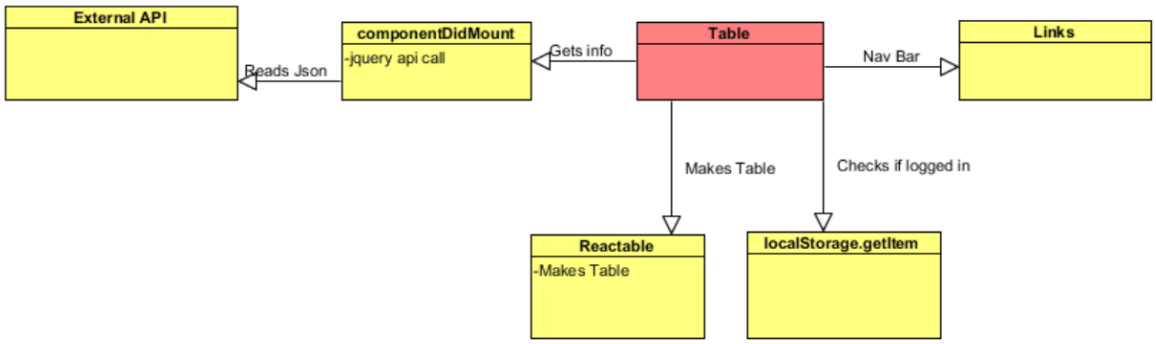
Table Page

Home About Teams Table Logout

Premier League Table

TEAM	GP	W	D	L	GF	GA	GD	PTS
Chelsea	12	9	1	2	27	9	18	28
Liverpool	12	8	3	1	30	14	16	27
Man City	12	8	3	1	27	11	16	27
Arsenal	12	7	4	1	25	12	13	25
Tottenham	12	6	6	0	18	8	10	24
Man United	12	5	4	3	17	14	3	19
Everton	12	5	4	3	16	14	2	19
Watford	12	5	3	4	17	20	-3	18
Bournemouth	12	4	3	5	14	16	-2	15
Southampton	12	3	5	4	12	12	0	14
Burnley	11	4	2	5	11	15	-4	14
West Bromwich	11	3	4	4	12	15	-3	13
Stoke City	12	3	4	5	13	19	-6	13
Leicester City	12	3	3	6	14	20	-6	12
Middlesbrough	12	2	5	5	10	13	-3	11
Crystal Palace	12	3	2	7	17	21	-4	11
West Ham	12	3	2	7	13	23	-10	11
Hull City	12	3	1	8	10	27	-17	10
Sunderland	12	2	2	8	12	21	-9	8
Swansea City	12	1	3	8	11	22	-11	6

The URL Pattern: http://localhost:3000/table



Text Form

Links

Table

|localStorage.getItem

|Reactable

|componentDidMount

||External API

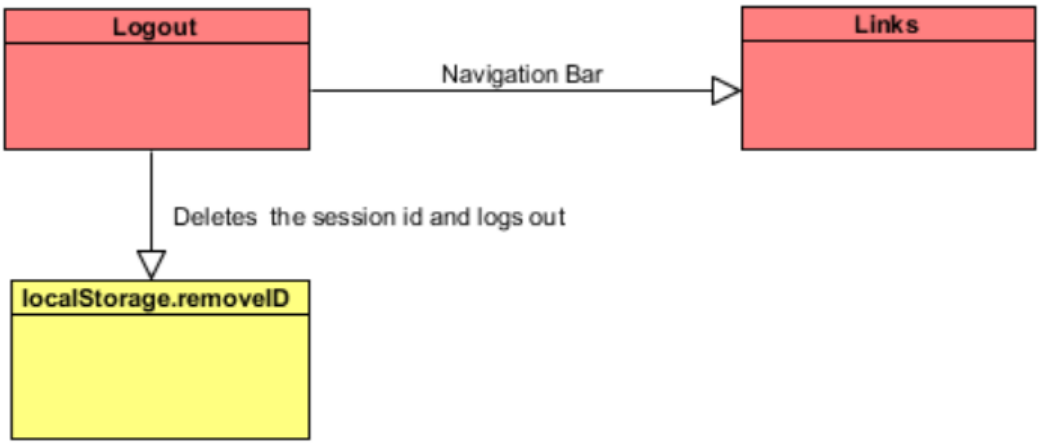
Table: {bplTeams(information from api), Table information}

Links: {ActivePage}

Logout Page

You have succesfully logged out

The URL Pattern: http://localhost:3000/logout



Text Form

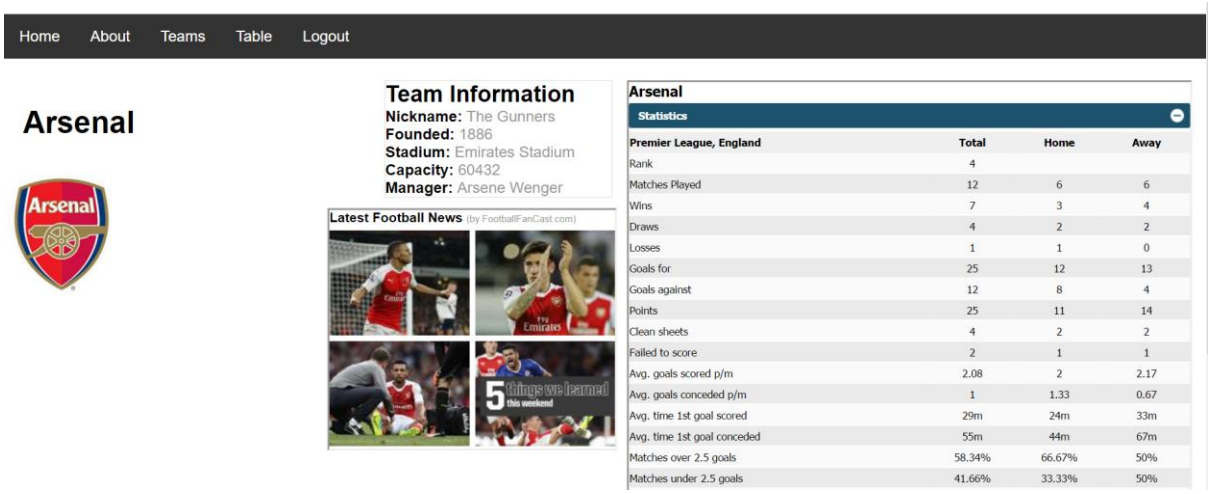
Links

Logout

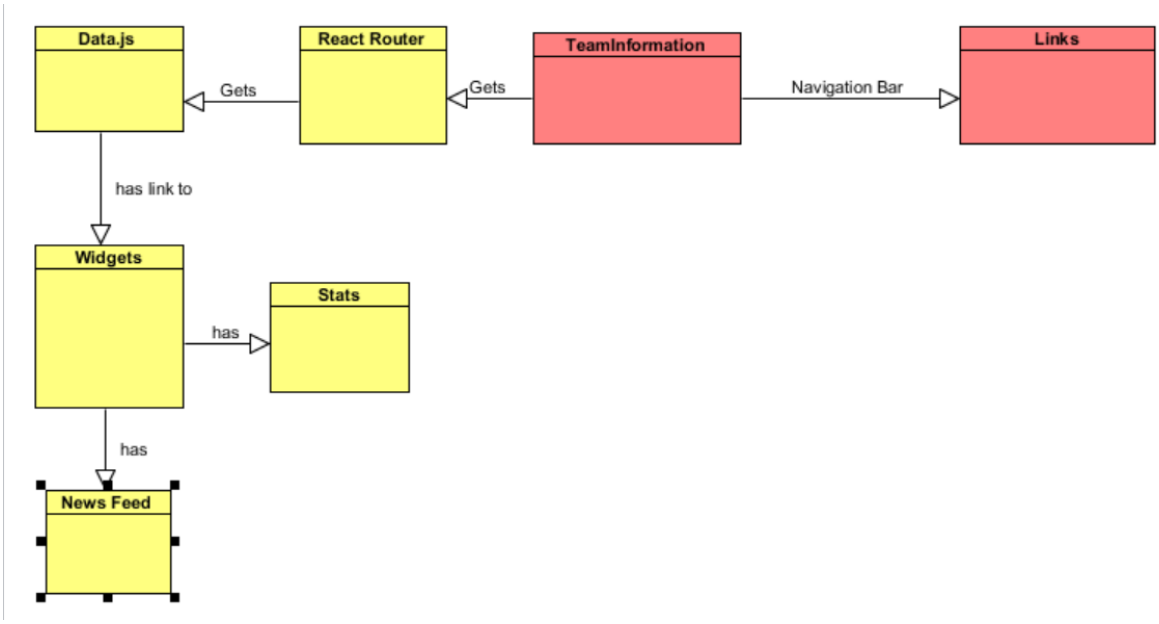
|localStorage.removeID

Links: {ActivePage}

Teams/:name Page



The URL Pattern: http://localhost/3000/teams/:name



Text Form

Links

TeamInformation

|React Router

||Data.js

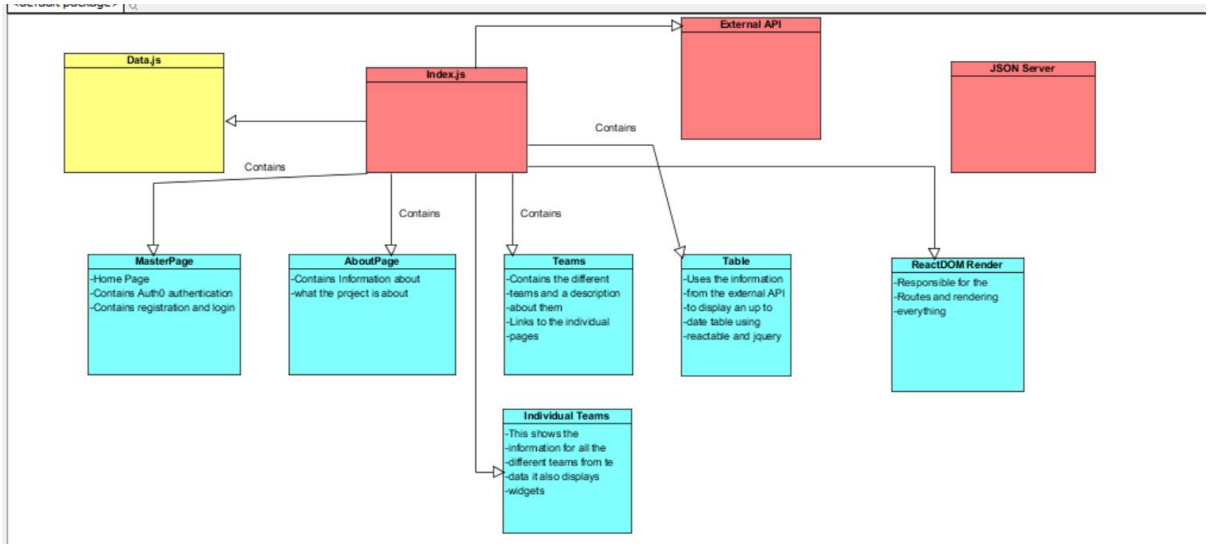
|||Widgets(Links)

||||Stats

||||News Feed

TeamInformation: {Teams(Data.js info), Widget info, param entered, }

Architecture



Extra Features

I have implemented a few different extra features which I briefly mentioned above. Some of these include:

1. The use of JSON Server to just show it implemented. This has full functionality (Can GET and POST to it), but it is currently not actually being used along side the application and is totally separate. If I had more time I would have changed from using a js file full of data to using the JSON Server to store it all.
2. The use of jquery to make an API call. I used jquery to make a call to an API which was quite difficult to find because a lot of the API's don't allow requests to come from servers such as the one we are using and gives an error about CORS not being enabled. This made it difficult to implement and at first, I had to implement it using a different format other than JSON which was JSONP but in the end I found a better API that allowed me to use JSON format with jquery.
3. I used an external API which allowed me to input up-to-date data on the latest Premier League table without doing anything manually. This was an important feature because if it didn't exist I would have had to manually update the whole table myself. It used jquery as discussed above to read all the data that the API had to offer, and I used the ones that I thought were relevant.
4. For the table that was used alongside the external API I used a package called 'Reactable' which made the use of tables very easy. It allowed me to search for the teams by name with a search bar along the top of the table. It also allowed me to sort the table by whichever column I wanted.
5. I used widgets to display different information about the individual teams which was a news feed for each team and also a statistics table for how they are doing this season. Both of these were contained in iframes which I implemented in the TeamInformation class and then stored each teams individual links in the dataset itself.
6. I used auth0 to be able to authenticate the application and add security to it. This means that anyone who hasn't signed up to the service will not be able to access any of the information on it. This means that the external API is harder to abuse and it's harder to abuse in general.
7. I also used auth0's lock system to be able to set up a registration, log in and logout. This allows people to create accounts using a number of different things:
    a. Email – Will also send a confirmation email to verify it's used. This does not need to be accepted to use the site though.
    b. Google – You can simply login and register using your gmail account.
    c. Twitter – You can simply login and register using a twitter account.
    d. GitHub – You can simply login and register using a GitHub account.
    e. Facebook – You can simple login and register using a Facebook account.

Independent Learning

For this assignment, I done a lot of Independent Learning which involved a lot of research and looking things up (All will be referenced below). One of the main areas of my Independent Learning came from looking at the different capabilities that react has and I've implemented a number of them into the project itself. There was also a lot of them that I didn't implement because they weren't really relevant to what I was trying to do.

I learned about auth0 and the role it plays it trying to keep data and applications secure and implemented a lock feature from them which is being used to authenticate, login and register users in a safe, secure manner.

I learned about JSON Server and its potential uses that it can have when it is working alongside a single page application. I implemented my own JSON server that is fully functional in this project.

I learned about jquery and the role it plays in retrieving data from API's. I also learned about some of the other ways you can do it such as fetch.

I learned about Reactable which is used to produce tables on data sets. I learned that it makes searching and sorting very easy and simple to do.

Links to other resources

https://scotch.io/tutorials/routing-react-apps-the-complete-guide

http://www.davidmeents.com/tables-react-you-can-sort-filter-paginate/

https://facebook.github.io/react/tutorial/tutorial.html

https://davidwalsh.name/react-authentication

http://alijafarian.com/responsive-image-grids-using-css/