Project 4 - NBA Application
Yehuda Colton

API: API-NBA (2.2.5)
Documentation: https://api-sports.io/documentation/nba/v2
Playground:
https://rapidapi.com/api-sports/api/api-nba/playground/apiendpoint_b58617b2-1f52-4b56-9894-07e35465d3de

1. Includes begins with 2 EditText fields, and outputs TextViews and ImageView fields. The user inputs an NBA team name into the first EditText and a season year into the second EditText. It then sends an HTTP request to the web service and receives a JSON formatted reply. The app then displays the information to the user. The team logo is displayed in the ImageView and the formatted team roster is diplayed in the TextView. The user can repeatedly make searches without restarting the app.

   I had trouble creating the AndroidStudio project so I used the bones of the AndroidInterestingPicture lab. I was not able to change the name because there are too many dependancies, so it has remained the name of my Android Project.

   List of NBA teams: (Capitalization does not affect the search)
   Hawks
   Celtics
   Nets
   Hornets
   Bulls
   Cavaliers
   Mavericks
   Nuggets
   Pistons
   Warriors
   Rockets
   Pacers
   Clippers
   Lakers
   Grizzlies
   Heat
   Bucks
   Timberwolves
   Pelicans
   Knicks
   Thunder
   Magic

76ers *(The logo URL returned by the api is broken and will not display a logo in the app)
Suns
Trail Blazers
Kings
Spurs
Raptors
Jazz
Wizards

Options of NBA seasons:
2015 - 2024

2. The servlet has 2 url paths. The first path is the homepage that has text welcoming the user to the site and has a link to the analysis dashboard. The second url path is for the analytics page. The Servlett accepts a HTTP request, extracts the search values, and makes sure they are valid. If they are not, the response is sent telling the app user that it was not a valid search. If it is valid, the api URL is built with the search values, and the servlet queries the api, and extracts the appropriate fields. It then builds player objects with relevant details, places them in a roster object and puts both a roster and the team logo in a team object. The servlet the send a JSON response of the team object to the App.
3. N/A
4. For each successful request/reply pair from the App, there are several pieces of information that get logged in a MongoDB Atlas database. The user agent header, team name, season, and timestamps of the app request, api request, api responss, and app response.
   Additionally, the number of times each team and season have been searched as well as the total number of searches is logged. This data is used to calculate search frequencies for teams and seasons.
5. See #4
6. The Dashboard can be accessed directly from a hyperlink on the homepage jsp or by the url path /app/analytics.
   There is a team frequency table, season frequency table, the total number of searches and the request/reply logs.