

# Info 253B Project Proposal

## NBA Data API

Info 253B Spring 2020

Aismit Das, Bryant Le, Rishabh Meswani, John Yang

Due 3/32/2020

The purpose of our API is to create an interface to our service that can scrape for NBA game data from websites or query from existing online databases for player data and team data to calculate advanced analytics for live sports games. Users will be able to send post requests for different players and receive game data and be able to do a live game comparison of stats, instead of having to search each player independently.

We want the API to either return previously known data (such as player and team statistics) as well as new analytics and insights based on previous data and live data. Some examples include basic metrics such as a player's points per game or total technical fouls. More advanced values could be values like player efficiency rating and real plus minus. There's a Medium article (link: <https://bit.ly/3adMilX>) that details some of the advanced analytics we'd like to implement. The minimum functionality we'd like our API to achieve is to query team and player data. However, if we nail those functions down, there'd be multiple directions we could extend into, such as comparisons of specific teams and players, data from different leagues (i.e. CBA, Euroleague), information regarding betting odds and money lines for certain games and outcomes, or other factual information about the league (i.e. schedule, playoffs, standings, news, player accolades, streaks, etc.).

We are entertaining the idea of calling external APIs that have historical basketball data along with livescore basketball data. For instance, while the popular basketball reference website, linked here: (<https://www.basketball-reference.com/>), is not an API, it does allow one to download CSVs, which we plan to target to scrape data. We also think popular sporting news and journal websites, particularly ESPN (<http://www.espn.com/apis/devcenter/docs/>) could be a great way to retrieve more live and recent data. If we implement requests that allow one to search for teams by region or state, we may use an API providing geographic data, such as Google Maps.

Two features of our API which require data that don't appear readily available through another API is 1. betting/odds data and 2. Certain advanced analytics. We believe this is our opportunity to implement the algorithms that calculate these values, then persist data in our own database so that they're available to be queried. We haven't researched which database library we'd like to use, but because this data will likely be hierarchical and structured in nature, we'd want to select whatever framework allows us to define custom structures and retrieve + mutate these values easily. Given this requirement, we'll explore MySQL as our first option. We might consider building multiple APIs if we want to keep the algorithms for calculating advanced analytics in a separate API, which is in turn called by the main NBA API. However, at the moment, this seems like it'd make things unnecessarily complicated. Per the requirements, our goal is to deliver both the API and a docker container it can run in.

### Tentative List of API Endpoints

This is a list of endpoints we're thinking about implementing. We might not do all of them, and we're sure there will be interesting endpoints that will come up as we write and design this API.

<b>Purpose</b>	<b>Endpoint</b>	<b>Input</b>	<b>Output (JSON)</b>
Get a player's information	GET /player	{ "first": "Kobe", "last": "Bryant" }	Player's historical data
Get a player's accolades	GET /player/awards	{ "first": "Kobe", "last": "Bryant" }	Player's accolades
Get a player's basic statistics	GET /player/stats/basic	{ "first": "Kobe", "last": "Bryant" }	Player stats (pts, reb, assist, etc.)
Get a player's advanced stats	GET /player/stats/adv	{ "first": "Kobe", "last": "Bryant" }	Advanced stats (PER, WS, etc.)
Get a player's news	GET /player/news	{ "first": "Kobe", "last": "Bryant" }	Summaries of latest headlines about player
Get a team's information	GET /team	{ "name": "LA Lakers" }	Team's historical data
Get team stats from specific year	GET /team/<year>	{ "name": "LA Lakers" }	List of data from specific year
Get specific team stat from specific year	GET /team/<year>/<stat>	{ "name": "LA Lakers" }	Stat data of team from given year
Get projected stats of team for next year	GET /team/future	{ "name": "LA Lakers" }	Projected stats of team for next season
Get news associated with specific team	GET/team/news	{ "name": "LA Lakers" }	Summaries of latest headlines about team
Get a list of today's games	GET /games	None	List of daily games
Get data about specific game	GET /games/<id>	None	Data of specific game
Get betting data about specific game	GET /games/<id>/odds	None	Betting odds of game
Get specific statistic about game	GET /games/<id>/<stat>	None	Stat data of game from given year