# LIRI Bot

### Overview

In this assignment, you will make LIRI. LIRI is like iPhone's SIRI. However, while SIRI is a Speech Interpretation and Recognition Interface, LIRI is a \_Language\_ Interpretation and Recognition Interface. LIRI will be a command line node app that takes in parameters and gives you back data.

### Before You Begin

1. LIRI will display your latest tweets. As we do not want to display your personal account, or its keys, please make an alias account and add a few tweets to it!

2. Make a new GitHub repository called liri-node-app and clone it to your computer.

3. To retrieve the data that will power this app, you'll need to send requests to the Twitter, Spotify and OMDB APIs. You'll find these Node packages crucial for your assignment.

\* [Twitter](https://www.npmjs.com/package/twitter)

\* [Spotify](https://www.npmjs.com/package/node-spotify-api)

\* [Request](https://www.npmjs.com/package/request)

\* You'll use Request to grab data from the [OMDB API](http://www.omdbapi.com).

### Instructions

1. Initialize a `package.json` file at your project root. Be sure to save all of the npm packages you'll be using to this file. If you fail to initialize a `package.json` file and save your dependencies to it, it will be troublesome, and at times almost impossible for anyone else to run your code.

2. Make a .gitignore file and add the following lines to it. This will tell git not to track these files, and thus they won't be committed to Github.

```

node\_modules

.DS\_Store

```

3. Make a JavaScript file named `keys.js`. \*\*Do Not\*\* add this file to the .gitignore. This would be a good thing to do in the real world, but it makes grading this assignment a challenge.

Inside keys.js your file will look like this:

```JavaScript

console.log('this is loaded');

var twitterKeys = {

consumer\_key: '<input here>',

consumer\_secret: '<input here>',

access\_token\_key: '<input here>',

access\_token\_secret: '<input here>',

}

module.exports = twitterKeys;

```

4. Get your Twitter API keys by following these steps:

\* Step One: Visit <https://apps.twitter.com/app/new>

\* Step Two: Fill out the form with dummy data. Type `http://google.com` in the Website input. Don't fill out the Callback URL input. Then submit the form.

\* Step Three: On the next screen, click the Keys and Access Tokens tab to get your consume key and secret.

\* Copy and paste them where the `<input here>` tags are inside your keys.js file.

\* Step Four: At the bottom of the page, click the `Create my access token` button to get your access token key and secret.

\* Copy the access token key and secret displayed at the bottom of the next screen. Paste them where the `<input here>` tags are inside your keys.js file.

5. Make a file called `random.txt`.

\* Inside of `random.txt` put the following in with no extra characters or white space:

\* spotify-this-song,"I Want it That Way"

6. Make a JavaScript file named `liri.js`.

7. At the top of the `liri.js` file, write the code you need to grab the data from keys.js. Then store the keys in a variable.

8. Make it so liri.js can take in one of the following commands:

\* `my-tweets`

\* `spotify-this-song`

\* `movie-this`

\* `do-what-it-says`

### What Each Command Should Do

1. `node liri.js my-tweets`

\* This will show your last 20 tweets and when they were created at in your terminal/bash window.

2. `node liri.js spotify-this-song '<song name here>'`

\* This will show the following information about the song in your terminal/bash window

\* Artist(s)

\* The song's name

\* A preview link of the song from Spotify

\* The album that the song is from

\* If no song is provided then your program will default to "The Sign" by Ace of Base.

\* You will utilize the [node-spotify-api](https://www.npmjs.com/package/node-spotify-api) package in order to retrieve song information from the Spotify API.

\* Like the Twitter API, the Spotify API requires you sign up as a developer to generate the necessary credentials. You can follow these steps in order to generate a \*\*client id\*\* and \*\*client secret\*\*:

\* Step One: Visit <https://developer.spotify.com/my-applications/#!/>

\* Step Two: Either login to your existing Spotify account or create a new one (a free account is fine) and log in.

\* Step Three: Once logged in, navigate to <https://developer.spotify.com/my-applications/#!/applications/create> to register a new application to be used with the Spotify API. You can fill in whatever you'd like for these fields. When finished, click the "complete" button.

\* Step Four: On the next screen, scroll down to where you see your client id and client secret. Copy these values down somewhere, you'll need them to use the Spotify API and the [node-spotify-api package](https://www.npmjs.com/package/node-spotify-api).

3. `node liri.js movie-this '<movie name here>'`

\* This will output the following information to your terminal/bash window:

```

\* Title of the movie.

\* Year the movie came out.

\* IMDB Rating of the movie.

\* Rotten Tomatoes Rating of the movie.

\* Country where the movie was produced.

\* Language of the movie.

\* Plot of the movie.

\* Actors in the movie.

```

\* If the user doesn't type a movie in, the program will output data for the movie 'Mr. Nobody.'

\* If you haven't watched "Mr. Nobody," then you should: <http://www.imdb.com/title/tt0485947/>

\* It's on Netflix!

\* You'll use the request package to retrieve data from the OMDB API. Like all of the in-class activities, the OMDB API requires an API key. You may use `trilogy`.

4. `node liri.js do-what-it-says`

\* Using the `fs` Node package, LIRI will take the text inside of random.txt and then use it to call one of LIRI's commands.

\* It should run `spotify-this-song` for "I Want it That Way," as follows the text in `random.txt`.

\* Feel free to change the text in that document to test out the feature for other commands.

### BONUS

\* In addition to logging the data to your terminal/bash window, output the data to a .txt file called `log.txt`.

\* Make sure you append each command you run to the `log.txt` file.

\* Do not overwrite your file each time you run a command.

- - -

### Minimum Requirements

Attempt to complete homework assignment as described in instructions. If unable to complete certain portions, please pseudocode these portions to describe what remains to be completed.

- - -

### One More Thing

If you have any questions about this project or the material we have covered, please post them in the community channels in slack so that your fellow developers can help you! If you're still having trouble, you can come to office hours for assistance from your instructor and TAs.

\*\*Good Luck!\*\*

## Copyright

Coding Boot Camp (C) 2016. All Rights Reserved.