

I have a video explanation of the app. It is available here:

<https://www.youtube.com/watch?v=QvTB0sAhsXo>

1. Purpose.

The App functions by allowing a user to:

- A. Input a band and receive concert information.
- B. Input a song and receive artist information.
- C. Input a movie and receive information about the movie.
- D. Run a random search that will consist of one of A, B, or C.

2. How it works:

- A. Liri Bot starts by grabbing the user's command input and text input. It stores the command line input. It then slices the text input (in case it's more than one word) and joins the words together into one value.

```
var command = process.argv[2];
console.log(command);

var value = process.argv.slice(3).join(" ");
```

Now the bot has the user command and the user input.

- B. The code is structured to run a function depending on the command line. There are four options, with a default option if the user types a command the bot doesn't recognize:

```
switch (command) {
  case "concert-this":
    // If no user input, need a default.
    if (!value) {
      value = "U2"
    }
    concertThis();
    break;

  case "spotify-this-song":
    if (!value) {
      value = "The Sign Ace of Base"
    }
    spotifyThis(value);
    // data.track.items
    break;

  case "movie-this":
    if (!value) {
      value = "Mr Nobody"
    }
    movieThis();
    break;

  case "do-what-it-says":
    doWhatItSays();
    break;

  default:
    console.log("Make sure you use one of the four commands!")
}
```

C. The user's command triggers a function.

a. Concert-this ("input") = Run a Bands in Town API call for the input band:

```
function concertThis() {  
  
  var axios = require("axios");  
  var queryUrl = "https://rest.bandsintown.com/artists/" +  
    value + "/events?app_id=codingbootcamp";  
  console.log("Here's the URL search: " + queryUrl);  
  
  axios.get(queryUrl).then(  
    function (response) {  
      console.log("-----");  
      console.log("Here are the concerts coming up for: "  
        + value);  
      var bandResults = response.data;  
      for (i = 0; i < bandResults.length; i++) {  
        // console.log(response.data);  
        console.log(JSON.stringify("They're Playing at: "  
          + bandResults[i].venue.name, null, 2));  
        console.log(JSON.stringify("City: " + bandResults  
          [i].venue.city, null, 2));  
        console.log(JSON.stringify("Region: " + bandResults  
          [i].venue.region, null, 2));  
        var concertDate = moment(bandResults[i].datetime)  
          .format('LL');  
        console.log(JSON.stringify("on: " + concertDate,  
          null, 2));  
        console.log("-----")  
      }  
    }  
  )  
}
```

The bot outputs the results in the console log, organized in a readable format.

b. spotify-this-song("input")= Run the spotify API call for the song:

```
function spotifyThis(value) {  
  // This function needs to pass an argument of value so  
  // that it will work with the "do what it says" function.  
  // Otherwise I will get an error message.  
  spotify.search({ type: 'track', query: value },  
    function (err, data) {  
      if (err) {  
        return console.log('Error occurred: ' + err);  
      }  
      console.log(value)  
      var results = data.tracks.items[0];  
      console.log  
        ("-----")  
      console.log(JSON.stringify("Song: " + results.name,  
        null, 2))  
      console.log(JSON.stringify("Album: " +  
        results.album.name, null, 2))  
      console.log(JSON.stringify("Artist: " +  
        results.artists[0].name, null, 2))  
      console.log(JSON.stringify("Preview Link: " +  
        results.preview_url, null, 2))  
      console.log  
        ("-----")  
    }  
  )  
}
```

- c. movie-this("input") = Run the OMDB API call for a movie:

```
function movieThis() {
  // Here's the variable for axios.
  var axios = require("axios");

  axios.get("http://www.omdbapi.com/?t=" + value + "&y=&plot=short&apikey=trilogy").then(
    function (response) {
      console.log("-----")
      // * Title of the movie.
      console.log("Title: " + response.data.Title);
      // * Year the movie came out.
      console.log("Release Year: " + response.data.Year);
      // * IMDB Rating of the movie.
      console.log("IMDB Rating: " + response.data.Ratings[0].Value);
      // * Rotten Tomatoes Rating of the movie.
      console.log("Rotten Tomatoes Rating: " + response.data.Ratings[1].Value);
      // * Country where the movie was produced.
      console.log("Country of Production: " + response.data.Country);
      // * Language of the movie.
      console.log("Language: " + response.data.Language);
      // * Plot of the movie.
      console.log("Plot: " + response.data.Plot);
      // * Actors in the movie.
      console.log("Actors: " + response.data.actors);
      console.log("-----")
    }
  );
}
```

- d. do-what-it-says = Run whatever command is contained within the random.txt file.

```
function doWhatItSays() {
  fs.readFile("random.txt", "utf8", function (error, data) {
    if (error) {
      return console.log(error);
    }
    var newValue = data.split(" ").slice(1).join(" ");
    spotifyThis(newValue);
  });
}
```

- D. The bot presents the information to the user in the terminal:

```
Thomass-MacBook-Pro-3:liri-node-app TFlynn$ node liri.js movie-this The Dark Knight
-----
Title: The Dark Knight
Release Year: 2008
IMDB Rating: 9.0/10
Rotten Tomatoes Rating: 94%
Country of Production: USA, UK
Language: English, Mandarin
Plot: When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.
Actors: Christian Bale, Heath Ledger, Aaron Eckhart, Michael Caine
-----
```

And also records the data in a file called log.txt using an appendfile function:

```
Title: The Dark Knight
Release Year: 2008
IMDB Rating: 9.0/10
Rotten Tomatoes Rating: 94%
Country of Production: USA, UK
Language: English, Mandarin
Plot: When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.
Actors: Christian Bale, Heath Ledger, Aaron Eckhart, Michael Caine

-----
Title: Some Like It Hot
Release Year: 1959
IMDB Rating: 8.2/10
Rotten Tomatoes Rating: 97%
Country of Production: USA
Language: English
Plot: When two male musicians witness a mob hit, they flee the state in an all-female band disguised as women, but further complications set in.
Actors: Marilyn Monroe, Tony Curtis, Jack Lemmon, George Raft

-----
Title: Inception
Release Year: 2010
IMDB Rating: 8.8/10
Rotten Tomatoes Rating: 87%
Country of Production: USA, UK
Language: English, Japanese, French
Plot: A thief who steals corporate secrets through the use of dream-sharing technology is given the inverse task of planting an idea into the mind of a C.E.O.
Actors: Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen Page, Tom Hardy
-----
```

3. To run the APP:

- Use one of four commands:
 - "concert-this"
 - "spotify-this-song"
 - "movie-this"
 - "do-what-it-says"
 - If you attempt to use a command beside those four, you will receive a message saying: "Make sure you use one of the four commands!"
- For the first three commands, include your search. For example:
 - "concert-this" The Killers
 - "spotify-this-song" Dance Yrself Clean
 - "movie-this" The Dark Knight
- For the random variable, you do not need to include a search. For example:
 - "do-what-it-says"
- If you fail to provide a proper search term, the app will use a default search.

4. The app uses:

- Javascript
- APIS:
 - Bands in Town
 - Spotify
 - Online Movie Database
- Node.js

```
○ "dependencies": {  
○   "axios": "^0.19.0",  
○   "dotenv": "^8.2.0",  
○   "moment": "^2.24.0",  
○   "node-spotify-api": "^1.1.1"
```

5. Created by Tommy Flynn.