require("dotenv").config();

console.log('dotenv: this is loaded');

// load spotify and assign it to a variable

var Spotify = require("node-spotify-api");

// getting axios going/linked

var axios = require("axios");

// link to keys.js and assing a variable to it

var keys = require("./keys.js");

// Load the fs package to read and write

var fs = require("fs");

// Take in the argument from the user

// The first will be the action (i.e. "concert-this", "spotify-this-song", etc.)

// The second will be the track/movie/concert/thing to do that will be added, spotify-this-song, etc.

var action = process.argv[2];

var value = process.argv[3];

var spotify = new Spotify(keys.spotify);

// Creation of the log.txt file in order to log my code results

var logger = fs.createWriteStream('log.txt', {

flags: 'a' // 'a' means appending (old data will be preserved)

})

// The switch-case will direct which function gets run.

switch (action) {

case "concert-this":

concertThis(value);

break;

case "spotify-this-song":

spotifyThisSong(value);

break;

case "movie-this":

movieThis(value);

break;

case "do-what-it-says":

doWhatItSays(value);

break;

}

// each command should be like this

//node liri.js concert-this <artist/I Want it That Way;

// This function will contact BandsInTown to get a concert played called by user

function concertThis() {

var bandsInTnKey = "7e9d74149cb19a07ef1d023000b73376"

//var bandsILike = require("./bands.js");

value = process.argv[3];

axios.get("https://rest.bandsintown.com/artists/" + value + "/events?app\_id=codingbootcamp")

.then(

function (response) {

console.log("--------------------------");

console.log("The list is all bands");

console.log(response.id);

console.log(response.data.venue);

console.log(response.data.datetime);

console.log("--------------------------");

// Otherwise, it will print: "log.txt was updated!"

logger.write("This is the Band " + response.id);

console.log("log.txt was updated!");

})

}

// This function will contact Spotify in order to play a song--------------------------------

function spotifyThisSong() {

value = process.argv[3];

if (value === undefined || value === " ") {

value = "Love"

};

spotify.search({

type: "track",

query: value

},

function (err, data) {

if (err) {

console.log("Error occurred: ", err);

return;

}

var songs = data.tracks.items;

for (let i = 0; i < songs.length; i++) {

console.log("Number: ", i, "/", songs.length);

console.log("artist(s): ", songs[i].artists.map(getArtistNames));

console.log("song name: ", songs[i].preview\_url);

console.log("album: ", songs[i].album.name);

console.log("-----------------------------------");

}

}

)

// if no song play "The Sign" by Ace of Base

console.log("finally arrived in spotifyThisSong");

if (err) {

return console.log(err);

}

var output = data.split(",");

if (value === undefined) {

console.log(output);

value = "The Sign";

}

const play = ({

spotify\_uri,

playerInstance: {

\_options: {

getOAuthToken,

id

}

}

}) => {

getOAuthToken(access\_token => {

fetch(`https://api.spotify.com/v1/me/player/play?device\_id=${id}`, {

method: 'PUT',

body: JSON.stringify({

uris: [spotify\_uri]

}),

headers: {

'Content-Type': 'application/json',

'Authorization': `Bearer ${access\_token}`

},

});

});

};

// play({

// playerInstance: new Spotify.Player({

// name: "value"

// }),

// spotify\_uri: 'spotify:track:7xGfFoTpQ2E7fRF5lN10tr',

// });

logger.write(", " + value)

};

//`node liri.js movie-this "movie name here-------------------------------"`

// default move is Mr. Nobody

function movieThis(value) {

console.log("This is value " + value);

var omdbKey = "8c2420ae"

// Then run a request with axios to the OMDB API with the movie specified

axios.get("http://www.omdbapi.com/?t=" + value + "&plot=full&apikey=8c2420ae").then(

function (response) {

console.log("Title: " + response.data.Title);

console.log("IMDB Rating: " + response.data.imdbRating);

console.log("The movie's rating is: " + response.data.Rated);

console.log("The movie came out in this year: " + response.data.Year);

console.log("The actors in the movie are " + response.data.Actors);

console.log("The language in the film is " + response.data.Language);

console.log("The movie is from " + response.data.Country);

console.log("Plot: " + response.data.Plot);

while (response.data.Ratings) {

if ((grandchild() === "Rotten Tomatoes")) {

console.log("This might be Rotten Tomatoes " + this);

}

}

// console.log("The movie Rotton Tomato rating is " + response.data.Ratings);

logger.write(", " + response.data.Title) // append string to your file

logger.write(", " + response.data.Actors) // again

logger.write(", " + response.data.Year) // again )

// Otherwise, it will print: "log.txt was updated!"

console.log("log.txt was updated with " + response.data.Year + "!");

});

}

function doWhatItSays(value) {

fs.readFile("random.txt", "utf8", function(error, data) {

// If the code experiences any errors it will log the error to the console.

if (error) {

return console.log(error);

}

// print the contents of data

console.log(data);

// Then split it by commas (to make it more readable)

var dataArr = data.split(",");

// We will then re-display the content as an array for later use.

console.log(dataArr);

});

}