

This api is used in conjunction with gorilla mux in order to connect both the front end and the back end. In this sprint, we were able to send requests to a certain url from the front end and then receive them on the backend and use some functionality on the backend side. We were able to get these requests and login in to the spotify api and get an authorization token to be used in both the spotify api and sdk. After the login, we were then redirected but are having difficulties with the final redirected back to the angular server we have running.

The run() function can be written in the main but is its own function for readability and compactnes.

The completeAuth() function is called upon the completion of the login into spotify. Once the login is completed, spotify redirects the page to a callback url we have specified in our spotify developer app. Once called, it creates a token using the login information. It has a couple error checks just to make sure everything is going through. At the end it crient a client from the token that can be treated as an object and lets us know we completed the login.

The healthCheck() function simply checks whether we can read the normal host server

The send RedirectURI() function is still a work in progres, but essentially redirects the current page we are on which should be the callback URI to the local server being hosted by Angular.

The sendToken() function creates a token json object to use for our project. The front end needs a token inorder to use the spotifySDK, so this function is what sends the token back to them so they can use it. It also has error checks within the function to make sure we can know wether the token is being ecnoded and used or not.