

Website/Front End

libraries:

- Chakra UI
- react/node

App.jsx

useState variables:

- listURL
 - String
 - setlist.fm URL submitted by user in input box
- songList
 - Array[{songname:tally}]
 - All songs from an artists tour, tallied and sorted by number of plays

functions

- fetchSetlist
 - Happens onClick for the “submit” button
 - Calls extractSetlistID (jazzy - I hate JS!! Do you write this as extractSetlistID or extractSetlistID()?? Not sure if these should be referred to as functions or variables?!) to extract the setlistID for a specific concert from its user-submitted URL
 - Sends fetch request to server. Server will process request. Response will be an Array[Object]. Each Object will be in the form {song: "All To Well", count: 6}. Array is sorted by count.
 - Sets Response as songList useState variable
 - (maybe should rename to fetch songList)
- extractSetlistID
 - Takes listURL useState variable and extracts the numeric setlist.fm numeric ID for further API queries

HTML DOM

- Input Bar
 - Sets listURL (user submitted)
- Button
 - onClick calls fetchSetlist to get songList
- Flex Column
 - Displays parsed songList once it is set

Server/Back End

libraries:

- Axios
- Express

Server.js

Single endpoint “/”

- Takes in request from front end. Makes 2 sequential setlist.fm API queries and stores the response in two variables tourInfo and allTourInfo. Parses data from allTourInfo, then stores parsed data in songList.
 - tourInfo
 - Queries setlist.fm via getTourName (in Utils).
 - Stores answer in tourInfo variable
 - If setlist API doesn't have a tourname for the show, sets 400 status error
 - allTourInfo
 - Uses tourInfo to query setlist.fm via getAllTourSongs (in Utils)
 - Stores response in allTourInfo
 - (Can't remember wtf lin 27-30 is doing).
 - allTourInfo is a nasty convoluted object of every show from a band's tour. It's fed into...
 - songList
 - Which is an ordered array created by getSongTally (in formatData.js)

Utils/setlistAPIRequests.js

- getTourName
 - Function to API query for band and tour info based on numeric setList ID (from user submitted URL)
 - Calls setlist.fm API w/ user submitted setList numeric ID
 - Returns an object {bandName: namestring, tourName: tourname}
- getAllTourSongs
 - Queries setlist API for all songs played on tour (has to be done in batches of 20 per setList.fm's rules, so also has internal calculations to query the right number of batches)
 - Returns a nasty convoluted object of every show from a band's tour. It's fed into...

Utils/formatData.js

- getSongTally
 - Parses the response from getAllTourSongs orders them in an ordered array based on # of times played on tour

Website/Front End

- Input Bar
 - Sets listURL (user submitted)
- Button
 - onClick calls fetchSetlist to get songList
- Flex Column
 - Displays parsed songList once it is set

Server/Back End

getTourName

getAllTourSongs

getSongTally

setlist.fm API

[https://api.setlist.fm/rest/1.0/setlist/\\${listID}](https://api.setlist.fm/rest/1.0/setlist/${listID})

[https://api.setlist.fm/rest/1.0/search/
setlists?artistName=\\${artistName}
&p=1&tourName=\\${tourName}](https://api.setlist.fm/rest/1.0/search/setlists?artistName=${artistName}&p=1&tourName=${tourName})

