2021

<Concert Map Application>

CAB432 Cloud Computing
Assignment 1

<Jun Chen>
<n10240977>
9/5/2021

Contents

Introduction

Mashup Purpose & description

The web application I have implemented is a search engine using **SongKick, Leaflet and Flickr** that propose two main functions on the map which is searching by location and by artist name. Firstly, users can search by the city name to view the upcoming events that are being held in the geo location with the markers stating the events information. Another function is to search by artist name, users can enter name on the search box which will return all location markers of the specific artist on the map. The main objective of my application helps user to get a better visual context to where and what events are held, allowing users to map out the location markers and see how far it is from anywhere in the world.

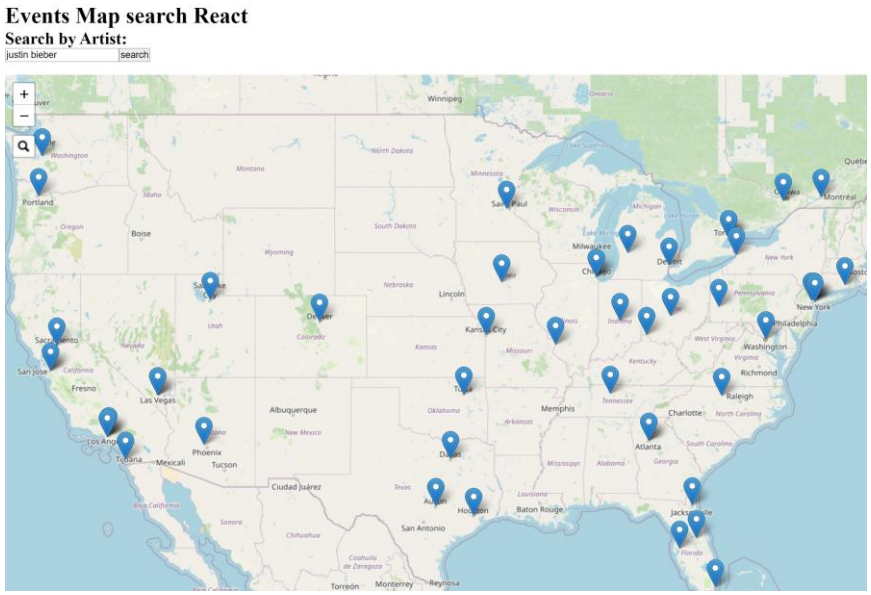


Figure 1. Search by artist

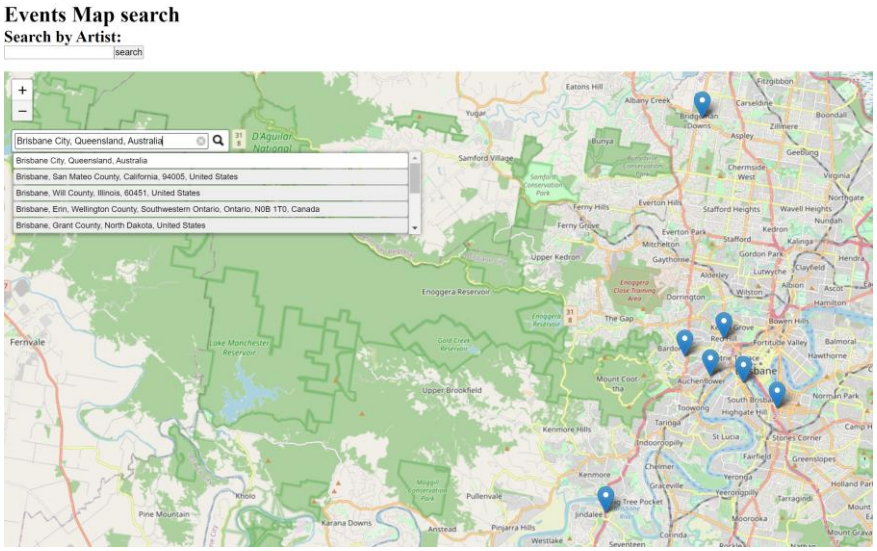


Figure 2. Search by city

Services used

1. SongKick API (v.3.0)

Returns a collection of events objects based on the input artist ID. It can be filtered to return the number of results per page to determine how many events you want to receive and retrieve.

Endpoints:

1. https://api.songkick.com/api/3.0/events.json?apikey=${SONGKICK_APIKEY}&location=?
2. https://api.songkick.com/api/3.0/events.json?apikey=${SONGKICK_APIKEY}&artist_name=?

Docs: https://www.songkick.com/developer/upcoming-events-for-artist

2. Flickr API

By calling a specific convention, it sends a request to Flickr endpoint specifying a method and some arguments to retrieve the Images based on the arguments. There will be many images in the data object, and it can be filtered to get a specific number of images tailored for the application.

Endpoint: https://api.flickr.com/services

Docs: https://www.flickr.com/services/api/

3. Leaflet API (v1.7.1.)

Leaflet returns an interactive map that provides view and markers of requested/clicked location on the map. Manipulation and options can be added to create a better visual and interaction of the map, adding information to popups and setting views.

Endpoint: https://nominatim.openstreetmap.org

Docs: https://leafletjs.com/reference-1.7.1.html

Mashup Use Cases and Services

User Story 1

As a	Concert fan
I want	A mapping service with markers
So that	I can view what concert events are held in that city or suburb when I clicked.

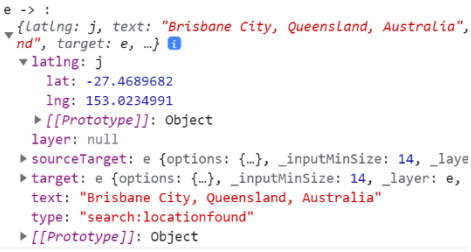


Figure 3. Leaflet search result

Searching for a particular location or city allows me to view the events from Songkick with the requested location longitude and latitude from Leaflet. Firstly, I input the location on the map with leaflet search control, I then get the **result** with specific longitude and latitude value of the location. Input the location to Songkick API to add the location markers and sort the results to specify the event. By clicking the markers, server will get the request and respond valid data to the client and on the client side I can manipulate and display the result on a popup icon stating the information of the concert in the location. The data will consist a image from **Flickr** and events infromation from **Songkick** to show the concert venue and relevant information.

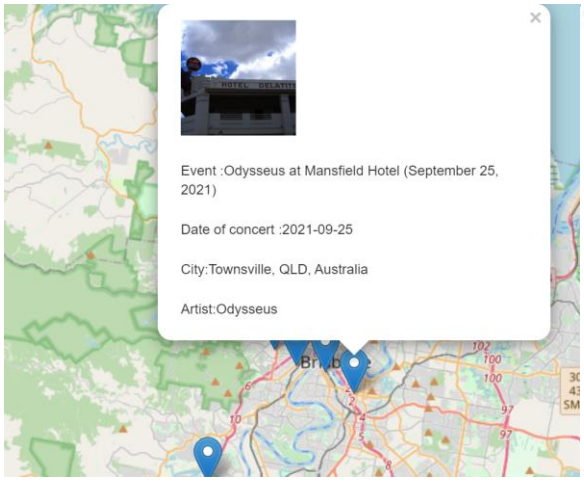


Figure 4. Display Pop up

Figure 5. Songkick API data

User Story 2

As a	Concert fan
I want	To search the event of my favourite artist
So that	I can view all of his concert location on the map and when it will be held.

Users can search for artist name to get all available concert related to the artist, the name of the search param will be sent from the user to the server. The search param will be passed to Songkick API to request the data of the artist and returned to the client with the respective data including

events and flickr image. The data received on the client side will be processed and mapped to output the display accordingly on the map. I first use the leaflet marker function to set the location on the map of all the concerts with the respective lng and lat values and then display the information on a pop up icon with the relevant events information.


```
data -> :                                                           index.e1334ff9.js:1
  (50) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
▼ {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
  ▼ 0:
      ageRestriction: null
      displayName: "The Event - Hosted by The Shaquille O'Neal Foundation 2021"
    ▶ end: {date: "2021-10-02", time: null, datetime: null}
      flaggedAsEnded: false
      id: 40008613
    ▶ location: {city: "Las Vegas, NV, US", lat: 36.10225, lng: -115.16997}
    ▶ performance: (5) [{…}, {…}, {…}, {…}, {…}]
      popularity: 0.73057
    ▶ series: {displayName: "The Event - Hosted by The Shaquille O'Neal Foundation"}
    ▶ start: {date: "2021-10-02", datetime: null, time: null}
      status: "ok"
      type: "Festival"
      uri: "https://www.songkick.com/festivals/3388955-event-hosted-by-the-shaquille-oneal-foun…
    ▶ venue: {id: 33948, displayName: "MGM Grand Garden Arena", uri: "https://www.songkick.com/…
    ▶ [[Prototype]]: Object
  ▶ 1: {id: 39380326, displayName: "Justin Bieber at Pechanga Arena (February 18, 2022)", type:…
  ▶ 2: {id: 39362103, displayName: "Justin Bieber at T-Mobile Arena (February 20, 2022)", type:…
  ▶ 3: {id: 39380332, displayName: "Justin Bieber at Gila River Arena (February 22, 2022)", typ…
  ▶ 4: {id: 39649059, displayName: "Justin Bieber at The Forum (February 23, 2022)", type: "Con…
  ▶ 5: {id: 39649044, displayName: "Justin Bieber at Tacoma Dome (February 26, 2022)", type: "C…
  ▶ 6: {id: 39649042, displayName: "Justin Bieber at SAP Center at San Jose (February 28, 2022)…
  ▶ 7: {id: 39649043, displayName: "Justin Bieber at SAP Center at San Jose (March 2, 2022)", t…
```


Figure 6. Songkick API artist name

Technical breakdown

Architecture and Data Flow

The application mashup comprises two main components which is the Server and Client request and response. The client will be making many requests for location and artist information from the server and getting response to process the data for users to view. I have created an express server with a set of function calls to get data from the APIs with the information given from the client. In the server, there are 3 routes, /getLocation, /getMarker and /getName which performs operation to get data from APIs for the client. This organization clarifies and separates the set of client operation, for instance when client requests the artist name, it will go through getName route and server will hit the APIs with the functions implemented and output the results for client side to process and display. The results will consist of the events information and the flickr image organized for client side.

```
//get artist name from client
app.get("/api/searchName", function (req, res) {
  var { geo, name } = req.query;
  getNameMarketByGeo(name).then(async (rs) => {
    var data =
      rs.resultsPage.results.event && rs.resultsPage.results.event.length
        ? rs.resultsPage.results.event
        : [];
    data.forEach(async (element) => {
      try {
        var imgRes = await getImg(element.venue.displayName);
        element.imgSrc = imgRes.photos.photo.length ? imgRes.photos.photo[0].url_q : '';
      } catch (error) {
        element.imgSrc = ''
      }
    });
    setTimeout(() => {
      res.json({ data });
    }, 1500)
```

Figure 7. Configure results from API

```
//search by artist name
const search = (params) => {
  var params = {
    name: artist_name,
  };
  //clear layer
  markersLayer.clearLayers();
  map.setZoom(2);
  axios.get("/api/searchName", { params }).then((res) => {
    var data = res.data.data;
    console.log("data -> :", data);
    if (!data.length) {
      //invalid artist search, display error message
      alert("There is no artist of this name");
      return;
    }
    data = data
      .map((it, index) => ({
        ...it,
        loc: [it.location.lat, it.location.lng],
      }))
      .map((it, index) => {
        var marker = new L.Marker(new L.latLng(it.loc)).on("click", (e) => {
          if (it.displayName) {
            marker.bindPopup(`
            <div className="pop">
            ${it.imgSrc ? `<img
              src=${it.imgSrc}
              width="100"
              height="100"
```

Figure 8. Search name of artist code

The above code snapshot highlights how the operation flows, client side first inputs the name and hits the route and when data is obtained, I can process it and display the results, for instance when there is no matched name, it will show a error alert or else display the event's location with marker and use the popup for event information. Due to how different APIs result are presented, it is necessary for me to analyse the data structure and only get the ones I need to present a well-organized display format.

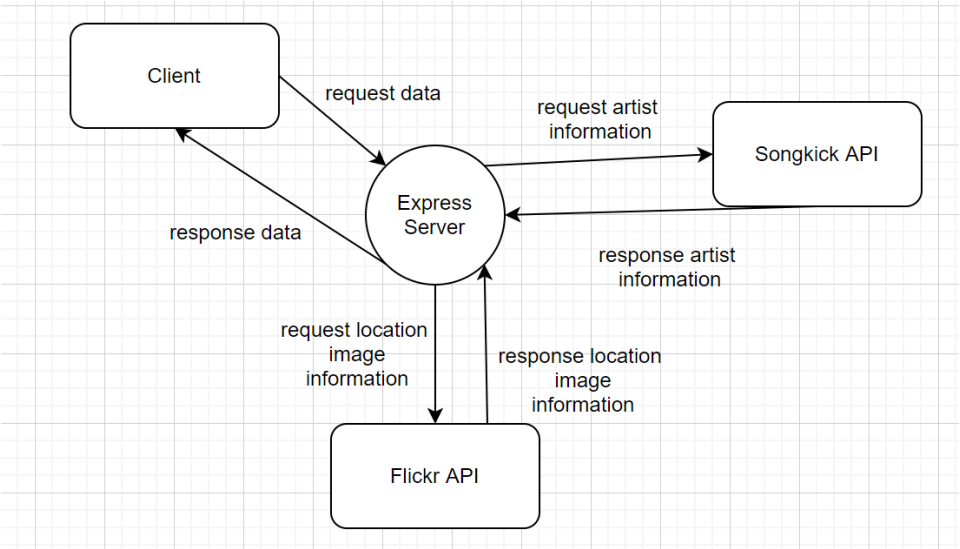


Figure 9. Data flow diagram

Deployment and the Use of Docker

I have used a Dockerfile to assemble the image by creating the App directory and copying all the files over such as package.json, server.js and app.js etc with the required npm methods to install the packages and run build. I have used node:16 tag for my image. I have also add a dockerignore file to prevent reading of huge and unnecessary files such as node modules and Dockerfile in the project. The docker file snapshot is included in Appendix A below.

Test plan

Task	Expected Outcome	Result	Appendix B
Search location	Markers displayed on the map	Pass	01
Search artist	Markers displayed on the map	Pass	02
Limit number of results	Search results are 50 and below	Pass	02
Enter invalid artist name	Error message popped up	Pass	03
Enter invalid location name	Error message popped up	Pass	04
Click marker to view info	Events information displayed on pop up	Pass	05
Handle flickr no image found	No image preview shows text message 'No image'	Pass	06
Check no duplicate markers	Markers with same ID are removed	Pass	07
Check markers all have popup	All markers must display text or events when clicked	Pass	08
Check no events markers	*No events marker will show 'No event'*	Pass	*09*

Difficulties / Exclusions / unresolved & persistent errors /

During the creation of mashup, Leaflet API returns all the markers of the search location with the longitude and latitude values which makes it difficult for me to compare the locations of the events from Songkick API which is why some markers have no events information instead of removing the markers. I improvised by adding text messages to show no events and images for this problem. Another major problem I encountered was iterating the locations on the leaflet map due to the syntax problem, I was stuck at how values can be placed on the markers and took me a long while to try the syntax to get it working.

Following, there is one bug on Leaflet map that I cannot overcome which is one specific marker in Alice Spring that responded with data, but it cannot read the value of the displayName. All other markers worked fine but this marker does have the data but it is just not reading it properly.

Screenshot of the problem is listed in the figure below:

Extensions (Optional)

With more time and research, I would like to add more components to the application, the one most interesting use would be adding videos to the display to show how the events are like. This will be added at the side div with the list showcasing the event held previously using other video APIs. In addition, adding spotify to play music related to the artist would be an interesting idea as well, adding a option to listen to the artist album.

User guide

The application has two distinct uses which is searching artist name and location to find the events. When the app starts up, you can see the map with two search boxes for users to enter (Figure 10). Enter the respective search parameters to get the results displayed on the map.

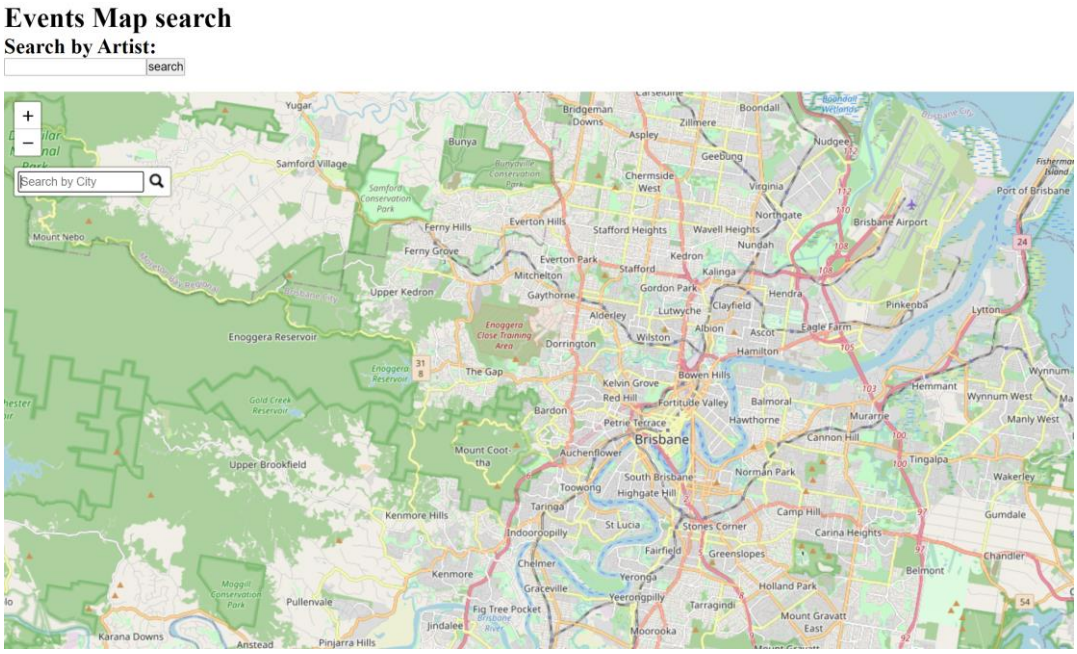
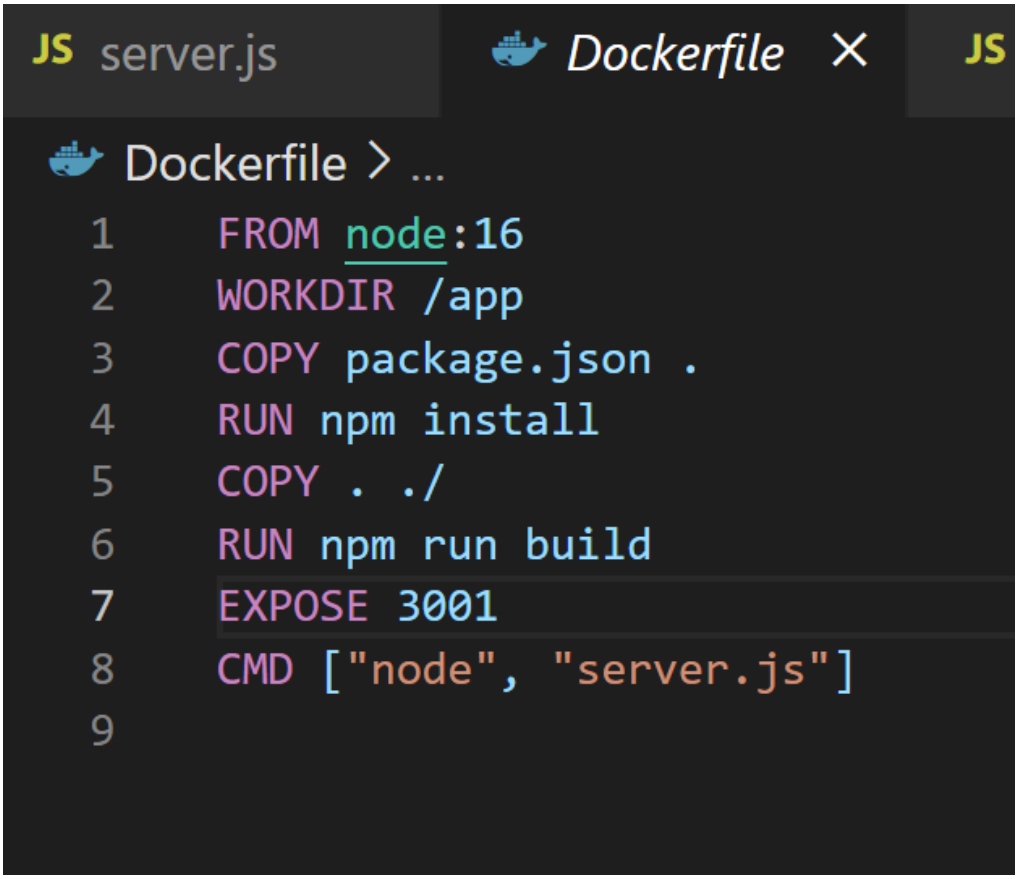


Figure 10. User guide

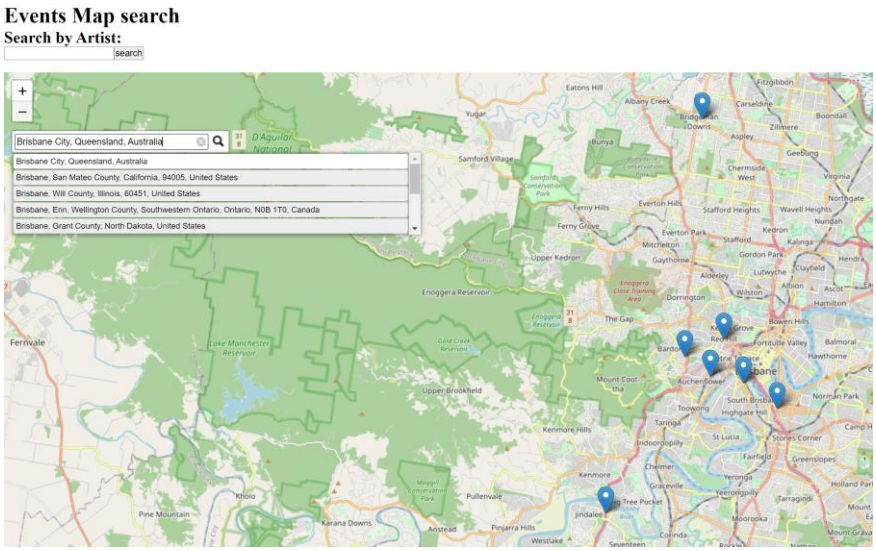
References

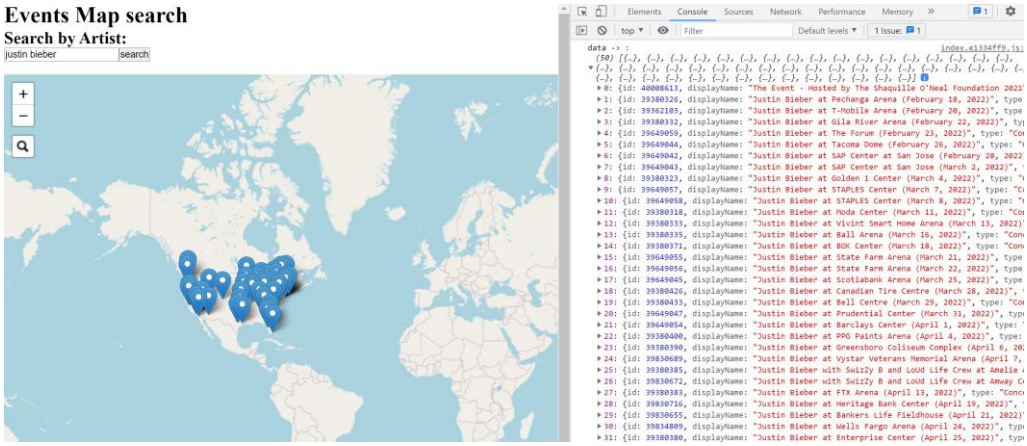
Appendices

Appendix A

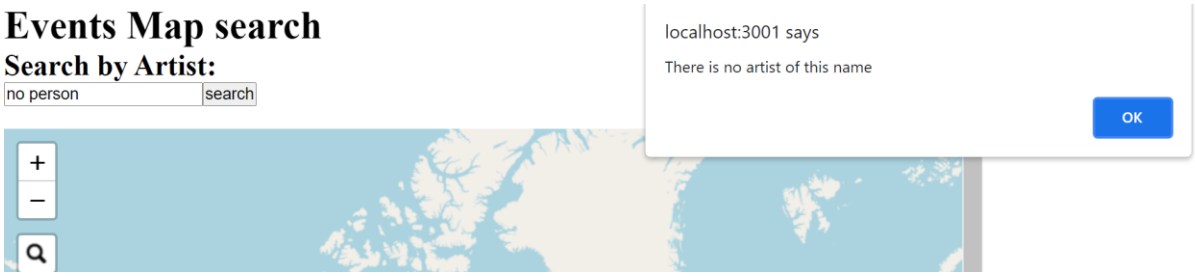


Appendix B

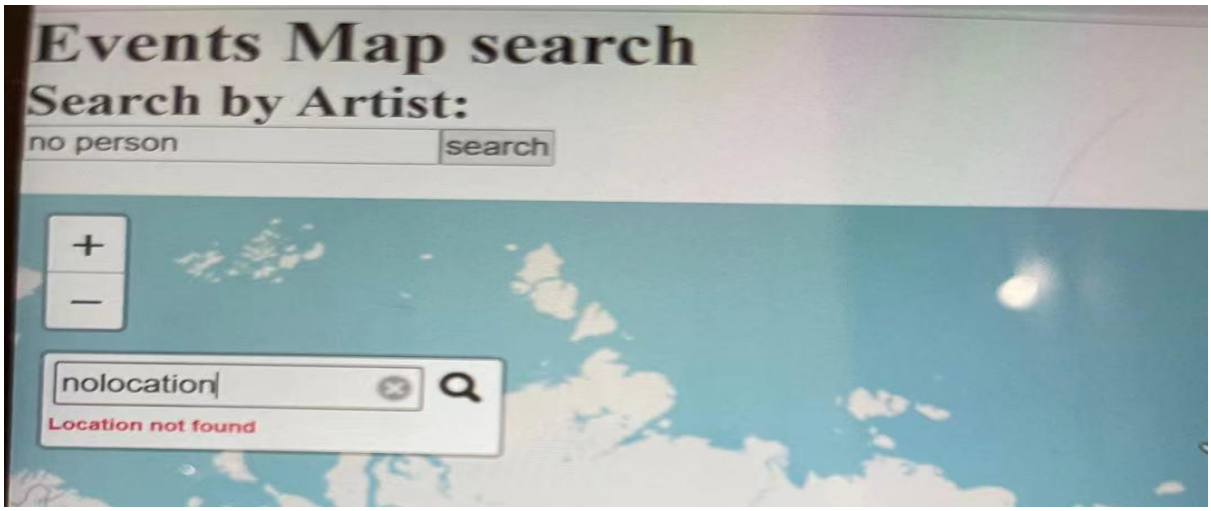



Events Map search
Search by Artist:
justin bieber
search
Elements
Console
Sources
Network
Performance
Memory
Filter
Default levels
1 Issue: 1

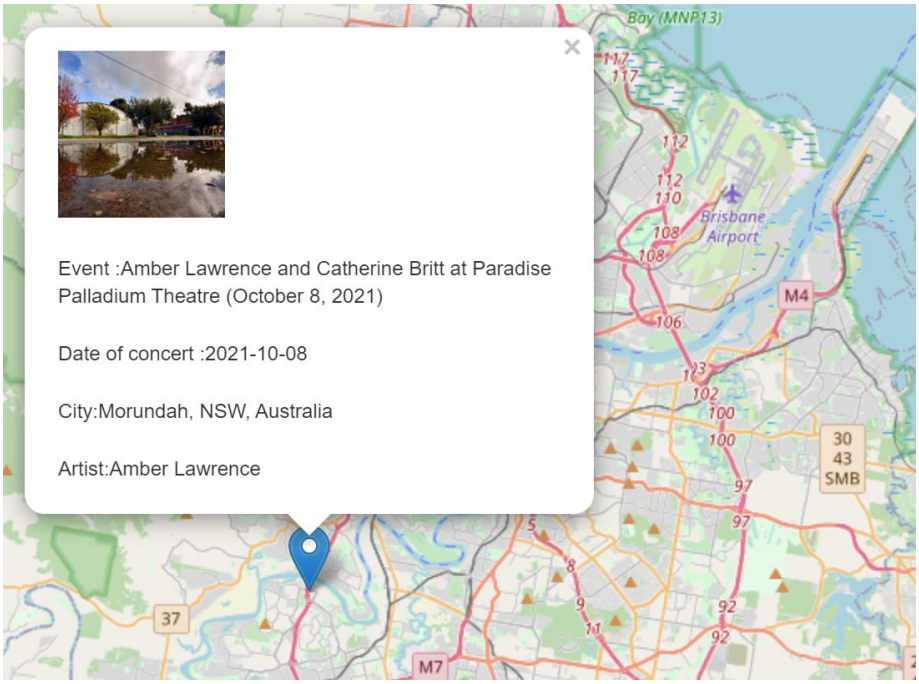
02


Events Map search
Search by Artist:
no person
search
localhost:3001 says
There is no artist of this name
OK

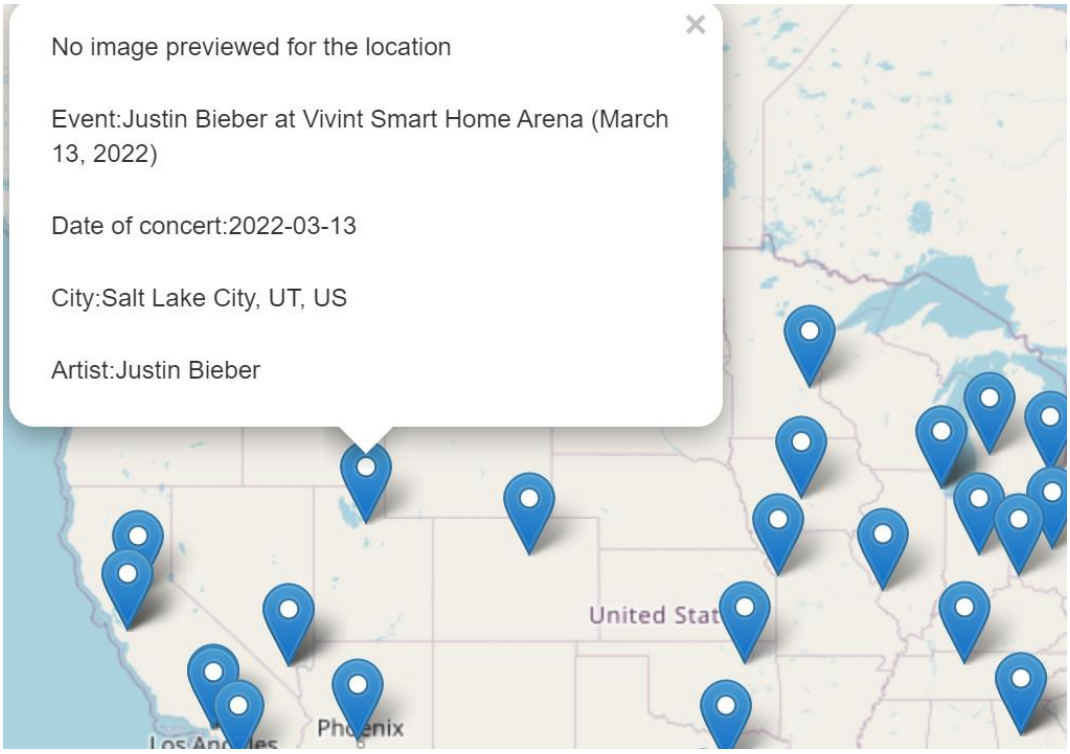
03


Events Map search
Search by Artist:
no person
search
nolocation
Location not found

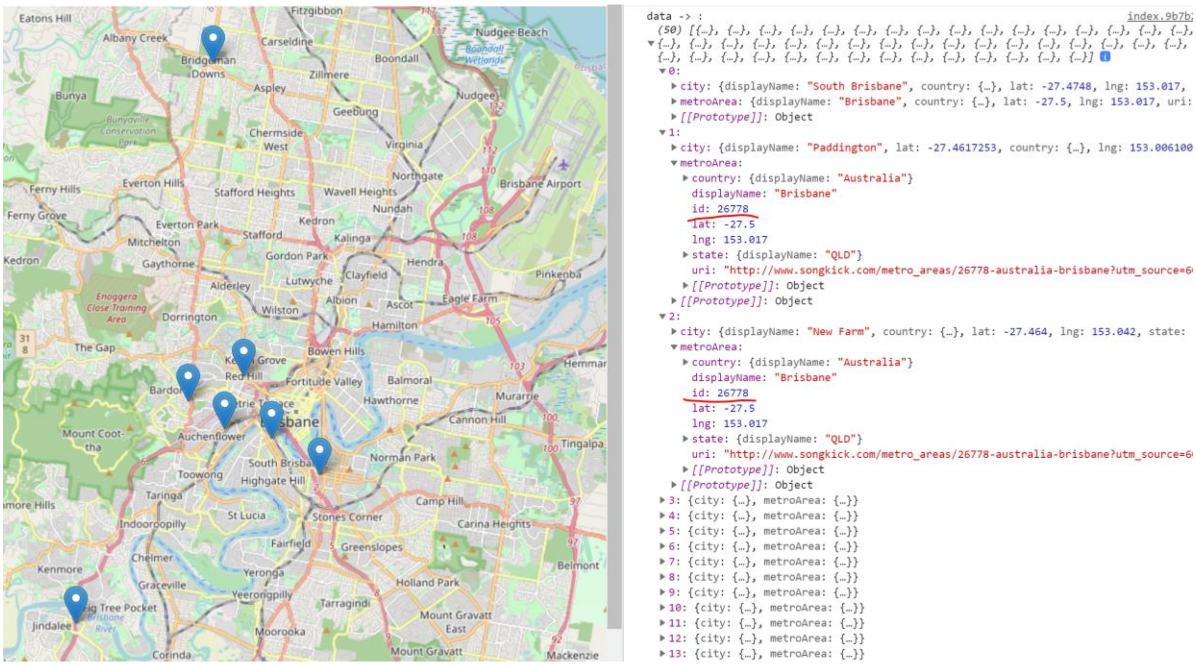
04


Bay (MNP13)
Brisbane Airport
M4
M7
37
30
43
SMB
Event :Amber Lawrence and Catherine Britt at Paradise Palladium Theatre (October 8, 2021)
Date of concert :2021-10-08
City:Morundah, NSW, Australia
Artist:Amber Lawrence

05

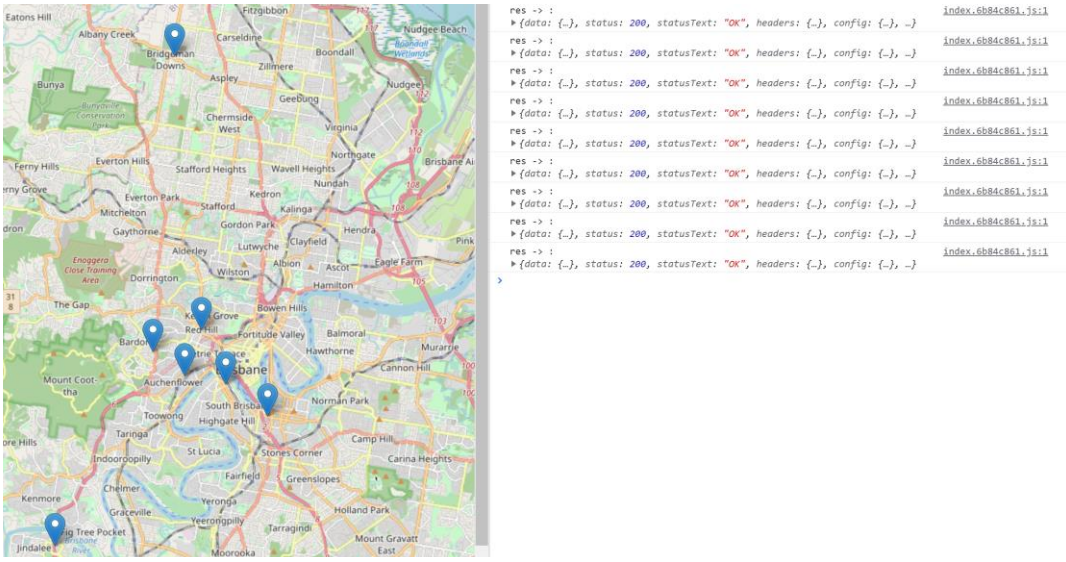

No image previewed for the location
Event:Justin Bieber at Vivint Smart Home Arena (March 13, 2022)
Date of concert:2022-03-13
City:Salt Lake City, UT, US
Artist:Justin Bieber
United Stat
Phoenix

06



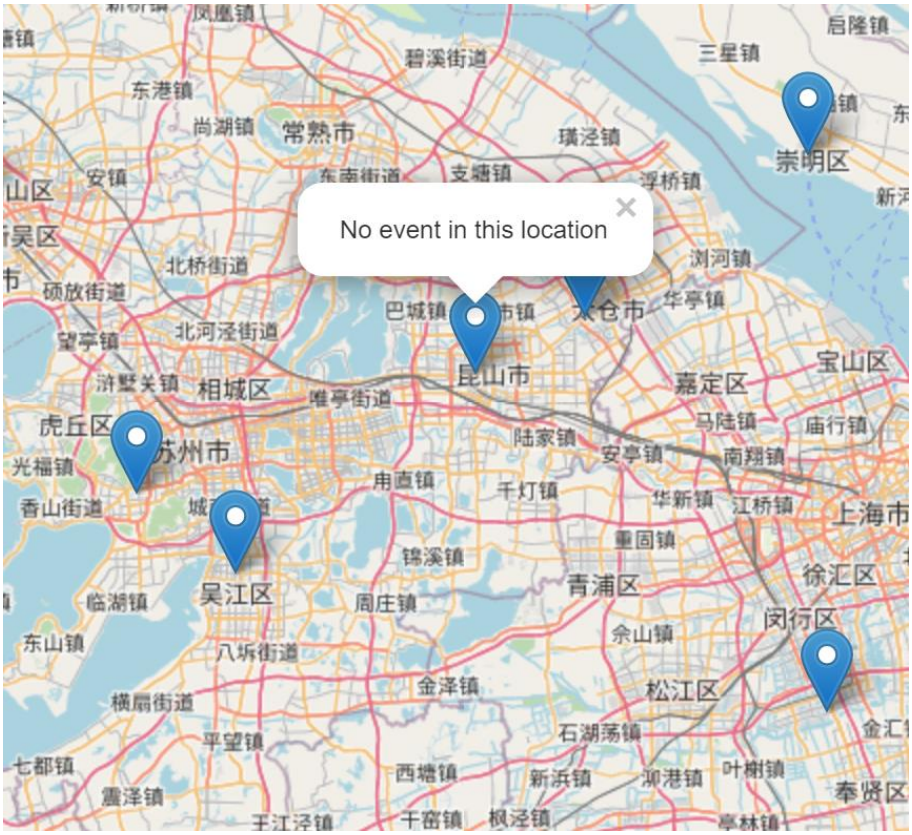
data -> :
index.9b7b
(50) [{…}, {…}]
0:
city: {displayName: "South Brisbane", country: {…}, lat: -27.4748, lng: 153.017,
metroArea: {displayName: "Brisbane", country: {…}, lat: -27.5, lng: 153.017, uri:
[[Prototype]]: Object
1:
city: {displayName: "Paddington", lat: -27.4617253, country: {…}, lng: 153.006100
metroArea:
country: {displayName: "Australia"}
displayName: "Brisbane"
id: 26778
lat: -27.5
lng: 153.017
state: {displayName: "QLD"}
uri: "http://www.songkick.com/metro_areas/26778-australia-brisbane?utm_source=6
[[Prototype]]: Object
[[Prototype]]: Object
2:
city: {displayName: "New Farm", country: {…}, lat: -27.464, lng: 153.042, state:
metroArea:
country: {displayName: "Australia"}
displayName: "Brisbane"
id: 26778
lat: -27.5
lng: 153.017
state: {displayName: "QLD"}
uri: "http://www.songkick.com/metro_areas/26778-australia-brisbane?utm_source=6
[[Prototype]]: Object
[[Prototype]]: Object
3: {city: {…}, metroArea: {…}}
4: {city: {…}, metroArea: {…}}
5: {city: {…}, metroArea: {…}}
6: {city: {…}, metroArea: {…}}
7: {city: {…}, metroArea: {…}}
8: {city: {…}, metroArea: {…}}
9: {city: {…}, metroArea: {…}}
10: {city: {…}, metroArea: {…}}
11: {city: {…}, metroArea: {…}}
12: {city: {…}, metroArea: {…}}
13: {city: {…}, metroArea: {…}}

07



res -> :
{data: {…}, status: 200, statusText: "OK", headers: {…}, config: {…}, …}
index.6b84c861.js:1

08


No event in this location
碧溪街道
三星镇
启隆镇
东港镇
尚湖镇
常熟市
璜泾镇
崇明区
安镇
支塘镇
浮桥镇
吴区
北桥街道
浏河镇
硕放街道
巴城镇
太仓市
华亭镇
望亭镇
北河泾街道
昆山市
宝山区
浒墅关镇
相城区
嘉定区
虎丘区
苏州市
陆家镇
安亭镇
南翔镇
光福镇
甪直镇
千灯镇
华新镇
江桥镇
上海市
香山街道
锦溪镇
重固镇
徐汇区
青浦区
临湖镇
吴江区
周庄镇
闵行区
东山镇
八坼街道
佘山镇
金泽镇
松江区
横扇街道
平望镇
石湖荡镇
七都镇
西塘镇
新浜镇
泖港镇
叶榭镇
震泽镇
千窑镇
枫泾镇
奉贤区