2020

Genevieve Richards

N9665544

9/15/2020

America Covid-19 Informer



Replace image with one with some relevance to your application here

CAB432 Assignment 1

Contents

[Introduction 3](#_Toc50539208)

[Mashup Purpose & description 3](#_Toc50539209)

[Services used 3](#_Toc50539210)

[Twitter Standard Search API (v.1.1) 3](#_Toc50539211)

[API 2 3](#_Toc50539212)

[API n 3](#_Toc50539213)

[Mashup Use Cases and Services 3](#_Toc50539214)

[Restaurant Search in a Foreign City 3](#_Toc50539215)

[US 2 4](#_Toc50539216)

[US n 4](#_Toc50539217)

[Technical breakdown 4](#_Toc50539218)

[Architecture and Data Flow 4](#_Toc50539219)

[Deployment and the Use of Docker 5](#_Toc50539220)

[Test plan 5](#_Toc50539221)

[Difficulties / Exclusions / unresolved & persistent errors 5](#_Toc50539222)

[Extensions (Optional) 6](#_Toc50539223)

[User guide 6](#_Toc50539224)

[Statement on Assignment Demo 6](#_Toc50539225)

[References 6](#_Toc50539226)

[Appendices 6](#_Toc50539227)

## Introduction

### Mashup Purpose & description

The American Covid-19 tracker app is a node based web app that shows you different covid-related information for different states in America. The initial page displays a Choropleth map of the total number of positive covid-19 cases each state has had with a darker green state showing more cases than a lighter green state. If you then hover over the state in the map you are shown more covid-19 related counts such as death tally and number of hospitalisations. It also lets the user know whether any restrictions are applied to the state at this point in time such as gathering limits, travel restrictions, restaurant limitions, face masks etc. If you click on the state in the map it will then take you to a new page that will show the user two bar graphs. One for the number of new positive cases over time since the first covid-19 case and one that shows you the google trend ranking for that state for the term coviid-19.

### Services used

#### Leaflet (v.1.1)

Returns the final map shown on the first page

Docs: https://leafletjs.com/examples/choropleth/

#### Covid Tracking API

There are two different endpoints of this app used to return first of the cumulative statistics shown on the choropleth map and the second to show the daily stats for each state.

API Endpoint 1: <https://api.covidtracking.com/v1/states/current.json> (updated daily)

API Endpoint2: <https://api.covidtracking.com/v1/states/>

*More here…*

#### Local Covid Data API

This API was used to get the current restrictions of each state in America.

API Endpoint: https://localcoviddata.com/covid19/v1/high-level-policy?country=USA

#### Google Trends API (V 4.9.0)

This API was used to track the interest in a keywork over time. It was accessed using the google-trend-npm package.

API Endpoint: <https://www.npmjs.com/package/google-trends-api>

#### Adhoc American State Geocode JSON

In order to get the geo-code information for each American state, the following .json was used from github and then joined with the covid tracking API data for each state.

API Endpoint: <https://raw.githubusercontent.com/PublicaMundi/MappingAPI/master/data/geojson/us-states.json>

## Mashup Use Cases and Services

*Your User stories should go here. The basic structure is provided, and you should fill in the role and then the action and the good result that follows. Underneath the formal statement of the user story, you can then tell us how you have implemented this service – basically you would tell us how you get input from the user and then use this to get information from an API and then use those results in another one. This is at a semi-technical level – introduce it at a high level and then give more detail, but stop well short of code excerpts. You should then use screenshots to illustrate the process*

*To illustrate some of what we want, I will make up an example that might not make much sense in 2020:*

#### Restaurant Search in a Foreign City

|  |  |
| --- | --- |
| As an | American citizen |
| I want | Understand how my state has fared in Covid-19 vs other states |
| So that | I can determine whether I might want to move |

When the user goes to the initial page of the covid-19 node app they are met with an choropleth map developed using leaflet for the UI, geojson from a github file and the covid tracking API. In order to mash all the APIs a few steps are taken:

1. Add state code to the geojson taking from the github API file. This is done using a static file list saved in public/javascripts/data.js that has key value pairs of state and state code.
2. The geojson file contains a field called density which is then replaced with the ‘positive’ field from the covid tracking API

This file is then routed to the client side and this data is used with the leaflet API to create a choropleth map of total confirmed positive covid-19 cases for US states.

A close up of a map

Description automatically generated

#### US 2

|  |  |
| --- | --- |
| As a | American Citizen |
| I want | Understand what restrictions are in in place in an American |
| So that | I can make sure I follow the rules in the states I am in |

In order to get extra information about a state including extra covid counts & current restrictions the user will hover over each state. When the user hovers over the state the state code is sent from the client in summary.js to the server side via route ‘/summary/:code’. This router then uses the local Covid-19 API with the state code from the client side to get the restrictions for the state. The same route Is the used and sent back to the client to be displayed on the info component in the top right corner of the leaflet map.

A close up of a map

Description automatically generated

#### US 3

|  |  |
| --- | --- |
| As a | American Citizen |
| I want | View a time series plot of new covid-19 cases over time |
| So that | I can understand if we are doing better than previous periods |

#### US 4

|  |  |
| --- | --- |
| As a | Data scientist with an interest in human behavior |
| I want | Understand the relationship between google trends of covid-19 and covid-19 cases |
| So that | Get a better understand of human behaviors in a pandemic |

In order to see the tracking of COVID-19 cases over time for my state, I click on the state of interest. The client will then use the code of the state that was clicked on to route the user to the new page of the address localhost:3000/states/:code. The statesRoutes in states.js then uses the code to get the daily state information over time for that state. At the same time this router also gets the google trends data for that state with the keyword covid and gets the earliest date in the daily state information to pass as the start date into the google trends api. The outputs of these two APIs are then used to render info.pug. Info.pug uses chart.js library to render two bar charts of trend and positive cases over time.

A screenshot of text

Description automatically generated

#### US 5

|  |  |
| --- | --- |
| As a | Someone with an interest in human behavior |
| I want | Search for a google keyword to assess the impact over time |
| So that | To see whether this keyword appears to correlate with the rise and rall of covid-19 cases |

TO DO IF I HAVE TIME TO ADD THIS USE CASE

## Technical breakdown

*This is a deeper discussion of the architecture, the technology used on the mashup, any issues encountered, and overall, how you implemented the project.*

### Architecture and Data Flow

*Explain how your system operates, making it clear how data flows around the system through requests and responses. You are describing the overall architecture of your application at a source code level. The description above tells us something of the application’s use. Now we want to see how that maps to the code organization – show us how your code is organized and tell us how you have split the responsibilities. We should get some sense of how the application works and how the data and control flows around. You may also find it helpful to show us screen grabs of code if that makes your points clearer. Tell us anything you think we need to know about how you have structured the application and made it work, but there also a section below to describe problems.*

*In Assignment 2 we provide you with a number of example diagrams that you may use to show how the application is structured. Most of these are optional even in assignment 2, but you might find the process flow diagram – especially as shown on the left hand side below – helpful in telling us how your application works. Here you may use a similar structure to show the transition between screens and service usage.*

|  |  |
| --- | --- |
|  |  |

### Deployment and the Use of Docker

The docker file uses the most recent node.js image taken (node:10) from <https://github.com/nodejs/docker-node>. The file then copies the files from the app directory directory of the docker image. It exposes port 3000 defined in the express app. It then uses npm install to install the app.

To build the docker image the user run:

docker build -t covid-19

To run the app the user will run:

docker run –p 80:3000 covid-19

### Test plan

*Manual testing is fine and our expectations are in line with the example grid below. You can show the results through a screen shot and point us to these from the table.*

*Your tests should include*

* *Positive outcome cases*
* *Negative outcome cases (error scenarios)*
* *Edge cases*
* *Non-functional cases (ideally, but not required this time).*

*Note that the grid below is unrelated to this application.*



Difficulties / Exclusions / unresolved & persistent errors /

*In this section, you should explain anything that caused you problems and how you overcame those problems. Tell us if there was any issue that prevented you completing the assignment to specification. Tell us about any assumptions or compromises that you have made. Those who worked with an API like Spotify, which presented particular concerns, should discuss the compromises here, and this is also where you can tell us about problems with API keys and responses.*

*More generally, you might consider:*

* *Your major roadblocks and how you resolved them.*
* *Any functionality you didn’t or couldn’t finish*
* *Are there any differences between your brief and what you delivered? If so, explain why.*
* *Are there any outstanding bugs?*

## Extensions (Optional)

*In this section, you can tell us if you wish to how you might extend your app and make it better. This is an opportunity to tell us about good ideas that you had that you didn’t have time to tell us about.*

## User guide

*Tell us how to use your application. You may re-use some of the screenshots from the use case descriptions, but this is more about how to use the app. As long as we can find what we need to do to use your application, this need not be all that long.*

*But either way, screenshots are your friend.*

## Statement on Assignment Demo

*In this section, please tell us explicitly whether you intend to demo your assignment face to face or via video. If you are demonstrating F2F, this tells us that we don’t need to look for a video and we can add you to our list of students for demo scheduling.*

*If you are demonstrating via video, the instructions will require you to submit a video as part of the zip archive for documentary reasons, but you may also like to make available a link to a higher resolution file. If you have made the video available via a web link – Dropbox or private YouTube link or whatever – please place the link here as this will assist the marker to move straight to your video and mark the assignment more quickly.*

## References

*Use a standard approach to referencing – see the guidance at* [*https://www.citewrite.qut.edu.au/cite/*](https://www.citewrite.qut.edu.au/cite/)*.*

## Appendices

*Stuff you want to include, but is too long or too complex to include in the main report text. The full Docker file, some longer excerpt from API docs. Whatever helps.*