COMP2140_A1

Planning Documents

Emily Ng Shao Yih

s4676654

trip updates .json API callback
- id
- trip Update: {
                    trip: {
                            tripId
                            routeId
                            start Time
                            start Date
                            scheduleRelationship
                    }
                    vehicle: {
                                id
                    }
                    stopTimeUpdate: [
                                    {
                                        stop sequence
                                        stop Id
                                        arrival: {
                                                    delay
                                                    time
                                                    uncertainty
                                        }
                                        departure: {
                                                    delay
                                                    time
                                                    uncertainty
                                        }
                                        scheduleRelationship
                                    } ...
                    ]
                    timestamp

vehicle_position.json API callback
- id
- vehicle: {
      trip: {
            tripId ←
            routeId ←
      }
      vehicle: {
            id
            label
      }
      position: {
            latitude
            longitude
      }
      stopId
      currentStatus
      timeStamp

~~**##**~~ make a new arr object

    [{ route_short_name: 1
        route_long_name: 2
        service_id: 3
        ~~test~~trip_headsign: 4
        arrival_time: 5
        live_arrival_time: 6
        live_position: }]]

~~tripstimes~~
- ~~use trip .txt data~~

~~route stuff remap it~~

- use stop timesArr
- then trip ~~##~~ Arr
- then route_id

Cosmic unicorns take you to sweet dreams

Flowchart diagram:

- welcome
- get user input
- ×3 / not right
- validate
- searchin API — *output in table form
- if go again / not right
- validate
- eval message

★ one function
★★ recursive

~~class~~ ~~U9~~ ~~translator~~ .

main — print welcome message
— run_parse() function
— print exit message

run-parser() — get_user_input() fun.
— get_translink () fun.

> Create lists of games that
residents can rent

Happy Birthday mother
yay

<u>trips</u>
route_id
service_id
trip_id
trip_headsign.
direction_id
block_id
shape_id

Calendar

service_id
mon
tue
wed
thur
fri
sat
sun
start_date
end_date

routes

route_id
route_short_name
route_long_name
route_desc
route_type
route_url
route_color
route_text_color

calendar_data

service_id
date
exception_type

Stop_times

trip_id
arrival_time
departure_time
stop_id
stop_sequence
pickup_type
drop-off_type

stops

stop_id
stop-code
stop-name
stop-desc
stop_lat
stop-lon
zone_id

stop_url
location_type
parent_station
platform_code

* Writ your own validate-date cuz inbuilt date format is weird

validate_date ("YYYY-MM-DD")
   ① Split into arr [YYYY, MM, DD]
   ② Check that All inp is int
   ③ Check valid MM
   ④ Check valid DD

can I geet the date out onto a diff sh format??

validate_time (" HH:MM")
   ① split into arr [HH, MM]
   ② check that it is all int
   ③ Valid Hour & Minute

* read txt to csv.

string

user input time = "HH:MM"

com output = + 10 mins

txt : "HH:MM:SS"