SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1246

VIZUALIZACIJA PROMETA JAVNOG PRIJEVOZA U STVARNOM VREMENU UPORABOM FORMATA GTFS

Luka Miličević

Zagreb, lipanj, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1246

Pristupnik:	**Luka Miličević (0036543289)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentorica:	izv. prof. dr. sc. Ivana Bosnić

Zadatak: **Vizualizacija prometa javnog prijevoza u stvarnom vremenu uporabom formata GTFS**

Opis zadatka:

Proučiti problematiku pružanja informacija o javnom prijevozu te dostupne norme za razmjenu takvih podataka. Posebnu pažnju posvetiti formatima namijenjenima slanju informacija u stvarnom vremenu. Istražiti normu General Transit Feed Specification (GTFS), a posebno podskup GTFS Realtime (GTFS RT). Za odabrani grad koji pruža usluge javnog prijevoza i daje informacije u formatu GTFS RT potrebno je razviti web-aplikaciju za vizualizaciju trenutnog položaja vozila javnog prijevoza na karti. S obzirom na moguća ograničenja izvora podataka i njihovo slanje u "skoro stvarnom vremenu" (near real-time), potrebno je ostvariti jednostavno predviđanje i animaciju kretanja vozila do sljedeće stanice. Aplikacija treba omogućiti korisniku odabir pojedine prijevozne linije koju želi pratiti i pružiti mu dodatne informacije, poput vremena dolaska vozila na određenu postaju, kašnjenja i slično. S obzirom na način uporabe, aplikacija treba imati responzivni dizajn uz korisničko sučelje prilagođeno i uređajima manjih zaslona.

Rok za predaju rada: 14. lipnja 2024.

Hvala na kavi...

Sadržaj

1. Uvod

U velikim urbanim gradovima poput Zagreba, javni gradski prijevoz predstavlja neizostavan dio svakodnevnog života. On je temeljni stup mobilnosti, pružajući vitalnu infrastrukturu za povezivanje građana i omogućujući lakše i efikasnije kretanje unutar grada. Velika većina građana svakodnevno koristi javni prijevoz te s obzirom na važnost vremena u užurbanom ritmu gradskog života, dobra organizacija i planiranje putovanja su ključni, jer svi bi htjeli 15 minuta koje potroše na čekanju prijevoza iskoristiti na drukčiji način. Upravo s tim ciljem, ideja završnog rada je stvoriti web aplikaciju, koja pruža stvarno praćenje tramvaja u realnom vremenu, olakšavajući korisnicima efikasno planiranje svojih putovanja, osiguravajući im precizne informacije o dolasku tramvaja i informacijama o pojedinim gradskim linijama prijevoza. Kroz ovaj zadatak, istražit ćemo tehničke izazove s kojima se susrećemo u razvoju takve web aplikacije. Dublje ćemo istražiti ključne elemente koji oblikuju temelje naše aplikacije poput analize API-ja pruženog od strane Zagrebačkog Električnog Tramvaja, proučavanje otvorenih standarda GTFS i GTFS realtime, te detaljniju analizu strukture web-aplikacije, od PostgreSQL baze podataka, backend dijela ostvarenog s node.js i Express.js, do frontend dijela implementiranog s React.js i bibliotekama poput Leaflet.js

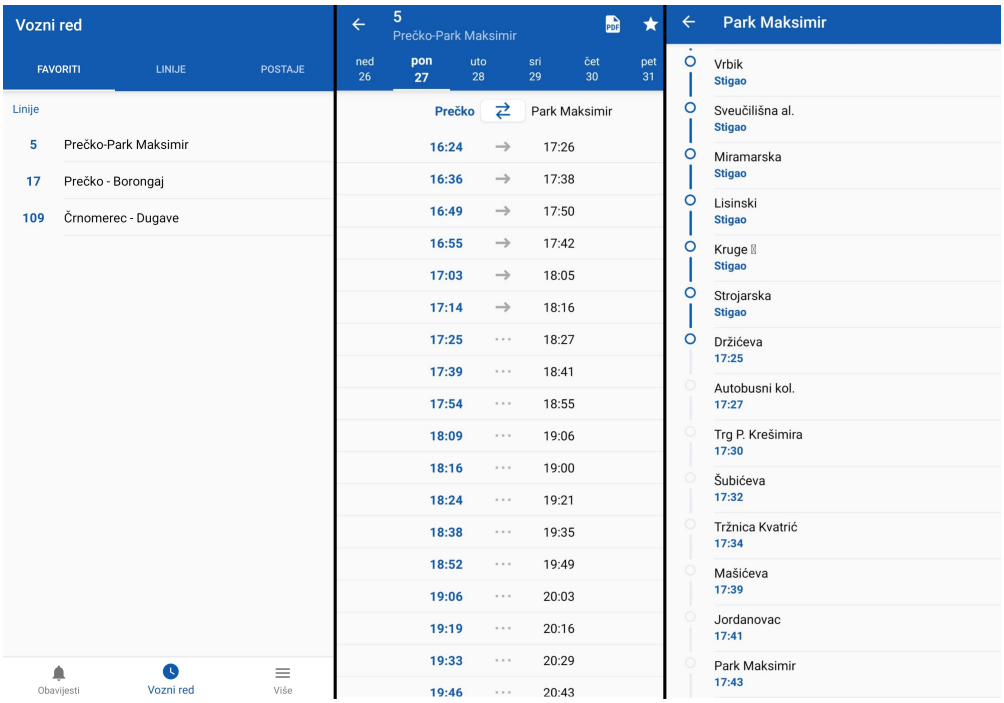
2. Glavni dio

2.1. Zagrebački Električni Tramvaj

Zagrebački električni tramvaj, poznatiji kao ZET, je trgovačko društvo koje je u direktnom vlasništvu Grada Zagreba, te ima ključnu ulogu u organizaciji javnog gradskog prijevoza u Zagrebu. Posluje od 1910. godine i svakodnevno pruža prijevozne usluge za gotovo milijun ljudi pomoću tramvaja, autobusa, uspinjače i specijalnih prijevoza [1]. Nedavno je ZET otvorio svoje podatke javnosti pod Otvorenom dozvolom Republike Hrvatske. Ovi podaci obuhvaćaju GTFS static i GTFS realtime podatke 2.2., koji su trenutno parcijalno implementirani te su dostupni na službenoj stranici ZET-a [2]. Ti podaci će biti temelj moje aplikacije i kroz njih ću detaljnije proći u narednim poglavljima 2.2.3.

Slične aplikacije - ZET info

Jedna popularna aplikacija koja koristi iste ZET-ove podatke je ZET info [3]. Koja primarno koristi GTFS static podatke za prikaz rasporeda tramvaja ZET-a. Ona nije službena ZET-ova aplikacija te dostupna je preko Apple App Store-a i Google play-a. Vrlo je zgodna za imati pri ruci, ali bih ja sa svojom aplikacijom htio imati i stvarne (realtime) podatke za tramvaje.



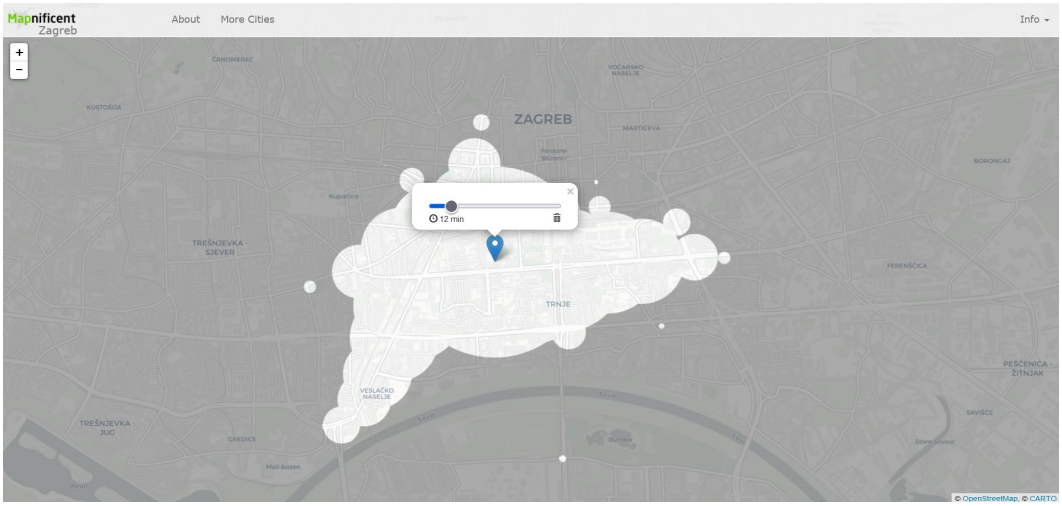
Slika 2.1. ZET info

2.2. General Transit Feed Specification

General Transit Feed Specification, kraće GTFS je otvoreni standardizirani format za rasporede javnog prijevoza i pripadajuće geografske informacije. Široko se koristi diljem svijeta te ga podržavaju više od 10 tisuća agencija javnog prijevoza u preko 100 zemalja. Neke od platformi koje koriste GTFS su Google Maps, Apple Maps, Moovit, OpenStreetMap i druge [4]. GTFS je započeo 2005. godine kao ideja za sporedni projekt Googleovog zaposlenika Chris Harrelsona te je danas De facto standard industrije. GTFS se sastoji od dva glavna dijela GTFS schedule (static) i GTFS realtime.

Činjenica da je GTFS otvoreni standard omogućuje da agencije za prijevoz informacije učine dostupnima pomoću bilo kojeg od mnogih alata koji već podržavaju GTFS i korisnicima da preuzmu i obrađuju te informacije sa proizvoljnom aplikacijom koja podržava GTFS ili ako žele razviju svoje programe za to. Otvoreni standardi dovode do stvaranja podataka koji se mogu lako dijeliti i potiču daljnji razvitak.

Jedan zanimljiv primjer male aplikacije koja koristi GTFS podatke da prikaže područja na karti do kojih možete doći javnim prijevozom u određenom vremenu je otvoreni projekt Mapnificent koji je javno dostupan na Github-u [5]. Na slici 2.2. je prikazano dokle je moguće doći javnim prijevozom od zgrade FER-a unutar 12 minuta.

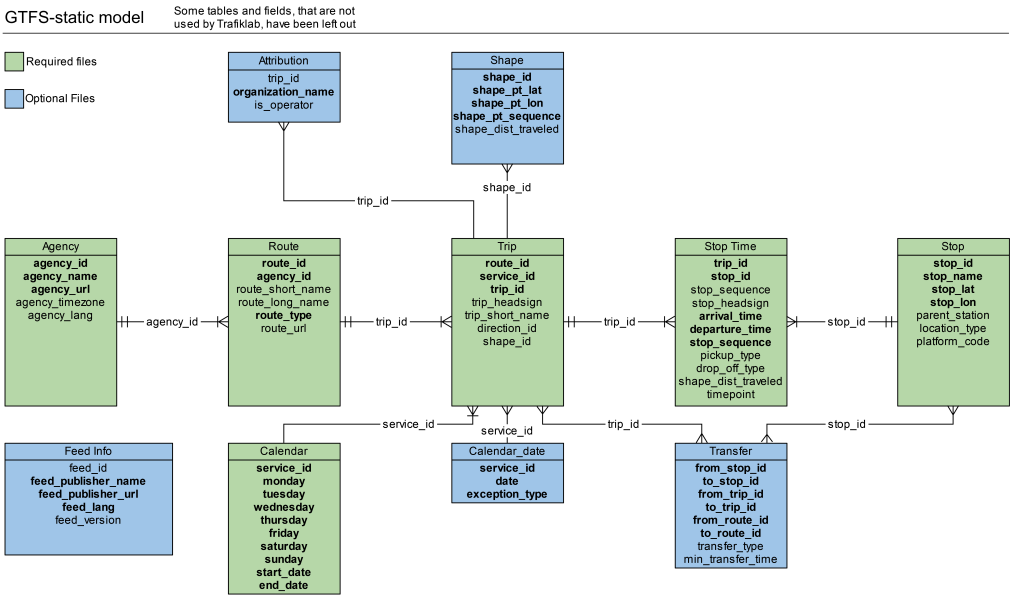


Slika 2.2. Mapnificent Zagreb

2.2.1. GTFS static

GTFS static ili GTFS schedule je kolekcija od barem 6 osnovnih, do 26 CSV (comma-separated values) datoteka s ekstenzijom .txt zapakiranih unutar jedne komprimirane .zip datoteke. Te datoteke sadrže informacije o rutama, rasporedima, stanicama, i ostalim raznim informacijama o prijevozu [6]. Sam format datoteka koji je CSV pruža otvorenost i jednostavnost rukovanja jer mnogo programskim alata i jezika ima metode za efektivnu obradu CSV što čini implementaciju strojnog čitanja lakoćom. Važno za napomenuti da je način kodiranja datoteka imperativan, obavezuje se korištenje UTF-8 kodiranja (podržan je i UTF-8 BOM).

Na dijagramu 2.3. prikazan je jedan primjer važeće strukture GTFS static modela kojeg prikladno prikazuju datoteke u ER (Entity–relationship) modelu. Zelenom bojom su označene neophodne datoteke, a plavom par dodatnih opcionalnih datoteka te za svaku od njih struktura i međusobne relacije.



Slika 2.3. GTFS-static model [7]

Neophodne datoteke

Moraju biti definirane da bi zadovoljili GTFS standard. One zajedno spojene pomoću pojedinih identifikatora čine cjelovitu sliku mreže javnog prijevoza

- **Agency.txt** - Opće informacije o prijevoznoj agenciji, uključujući njezino ime, web stranicu, i vremensku zonu.
- **Routes.txt** - Definira različite linije javnog prijevoza. Svaka ruta obuhvaća informacije o identifikatorima ruta, tipovima vozila i drugim detaljima.
- **Trips.txt** - Informacije o specifičnim putovanjima na određenim rutama, uključujući vremena polaska i druge povezane podatke.
- **Stop_times.txt** - Informacije o vremenima dolaska i odlaska vozila na svakoj stanici tijekom svakog putovanja.
- **Stops.txt** - Informacije o stanicama, uključujući identifikatore, nazive i geografske koordinate.
- **Calendar.txt** - Informacije o tjednom rasporedu putovanja.

Opcionalne datoteke

Pored neophodnih datoteka korisno je imati i dodatne opcionalne datoteke poput:

- **Shapes.txt** - Informacije o geografskim koordinatama rute i omogućuje lijepo iscrtavanje rute na karti.
- **Transfers.txt** - Informacije o povezivanju više ruta kako bi se olakšala mogućnost presjedanja i kombiniranja prijevoznih linija.
- **Calendar_dates.txt** - Omogućuje definiranje iznimaka u redovnom rasporedu kada usluga radi ili ne radi na pojedinoj liniji zbog smetnji ili radova, ili pak posebnih prijevoza.

2.2.2. GTFS realtime

GTFS realtime ili GTFS-rt je ekstenzija GTFS-a koja je nastala 2011. godine te je također de facto industrijski standard za dijeljenje stvarnih podataka o prijevozu. Jer je GTFS otvoreni standard, realtime se razvio s ciljem maksimalne interoperabilnosti i lakoće korištenja da bi olakšali programerima razvoj aplikacija i lakoću dijeljenja informacija o prijevozu između agencija. GTFS-rt sadrži informacije u stvarnom vremenu o lokacijama vozila, predviđenim vremenima dolaska te obavijestima o promjenama ruta i otkazivanjima putem web poslužitelja (putem nekog API) koji koristi protocol buffers (protobuf, poglavlje 2.3.). Podaci o stvarnoj lokaciji stvaraju se neprekidno od strane agencije sustavom za automatsko praćenje vozila dok se vremena dolaska na odredište najčešće izračunavaju pomoću strojnog učenja koji analiziraju povijesne podatke o položaju i voznom redu te daje očekivana vremena. Upravo jer se podaci neprestano kreiraju koriste se protocol buffers koji jako brzo i efektivno rade binarnu serijalizaciju podataka u male pakete koji se lako šalju [8].

Glavni "objekt" GTFS realtime feed-a (stream) je FeedMessage koji sadrži FeedHeader i FeedEntity. FeedHeader sadrži osnovne informacije (metapodatke) o podacima poput verzije GTFS-a, "incrementiality" koji opisuje šalje li se cijeli skup podataka (dataset) ili samo razlika od posljednje verzije te vremenska oznaka (timestamp) kada su podaci generirani. Feed entity sadrži jedan ili više entiteta koji sadrže najnovije informacije o stanju puta, pozicijama vozila, promjenama na rutama i upozorenjima. Neka od njih su:

- **tripUpdate** - predstavlja informacije jednog te sadrži objekte trip (osnovne informacije o ruti/putu), stopTimeUpdate, vehicle (informacije o položaju vozila npr. geografske koordinate)
- **stopTimeUpdate** - nalazi se unutar tripUpdate i on sadrži informacije o dolascima/odlascima sa stanica te kašnjenja
- **alert** - sadrži upozorenja od nesreća, radova na cesti i njihovi obilazaka, vremenskih upozorenja i drugih raznih iznimaka

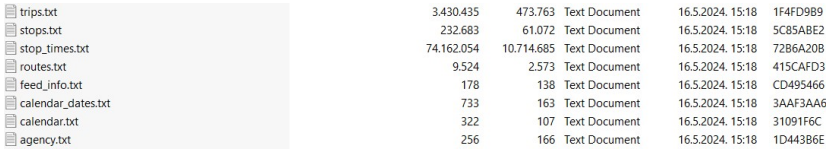
Primjer GTFS-rt podataka se može vidjeti u sljedećem poglavlju na slikama 2.10. i 2.11.

2.2.3. ZET GTFS podaci

Kao što je već rečeno ZET-ovi podaci su dostupni na službenoj stranici svima pod Otvorenom dozvolom Republike Hrvatske [2]. Na slici 2.4. je prikazana ta stranica te na njoj možemo pronaći linkove za GTFS static i GTFS realtime podatke.

Prvo ćemo proći kroz ZET-ove GTFS static podatke, a potom i GTFS realtime podatke. Kada sta stranice skinemo static arhivu i otvorimo ju s odgovarajućim alatom za pregled komprimiranih arhiva dobivamo datoteke sa slike 2.5.

Slika 2.4. ZET stranica za GTFS podatke



trips.txt	3.430.435	473.763	Text Document	16.5.2024. 15:18	1F4FD9B9
stops.txt	232.683	61.072	Text Document	16.5.2024. 15:18	5C85ABE2
stop_times.txt	74.162.054	10.714.685	Text Document	16.5.2024. 15:18	72B6A20B
routes.txt	9.524	2.573	Text Document	16.5.2024. 15:18	415CAFD3
feed_info.txt	178	138	Text Document	16.5.2024. 15:18	CD495466
calendar_dates.txt	733	163	Text Document	16.5.2024. 15:18	3AAF3AA6
calendar.txt	322	107	Text Document	16.5.2024. 15:18	31091F6C
agency.txt	256	166	Text Document	16.5.2024. 15:18	1D443B6E

Slika 2.5. ZET GTFS static podaci

Kao što vidimo tu su sve neophodne datoteke (trips.txt, stops.txt, stop_times.txt, routes.txt, calendar.txt i agency.txt) te par dodatnih datoteka feed_info.txt i calendar_dates.txt koje nisu pretežito korisne za moju implementaciju aplikacije. Od svih tih jedino nisam spomenuo što sadrži datoteka feed_info.txt - Ona sadržava opće informacije o GTFS feed-u, odnosno informacije poput naziva agencije koja je stvorila feed, vremena kada je feed generira, verzije GTFS standarda koja se koristi, i slično.

Pobliže ćemo pogledati datoteke trips.txt, stops.txt, stop_times.txt i routes.txt jer sam njih primarno koristio.

trips.txt

```
route_id,service_id,trip_id,trip_headsign,trip_short_name,direction_id,block_id,shape_id
268,"0_21","0_21_26820_268_10001","V. Gorica",,0,26820,
268,"0_21","0_21_26820_268_10002","Gl.kolodvor",,1,26820,
268,"0_21","0_21_26820_268_10007","V. Gorica",,0,26820,
268,"0_21","0_21_26820_268_10008","Gl.kolodvor",,1,26820,
268,"0_21","0_21_26820_268_10013","V. Gorica",,0,26820,
268,"0_21","0_21_26820_268_10014","Gl.kolodvor",,1,26820,
212,"0_21","0_21_21220_212_10003","Sesvete",,0,21220,
```

Slika 2.6. Izgled trips.txt

Bitne informacije u trips.txt su:

- route_id - identifikator rute, odnosno broj linije (npr. Kod 17 Borongaj, to je 17)
- trip_id - identifikator putovanja na pojedinoj ruti (svaka ruta ima više tramvaja koji više puta dnevno putuju na toj ruti)
- trip_short_name i direction_id - ime smjera tramvaja i binarni broj koji određuje smjer (npr. Borongaj)

Također možemo primijetiti da datoteka sadrži i shape_id koji se koristi kao ključ za relaciju trips.txt i shapes.txt koja kao što je rečeno sadrži detaljne koordinate putanje pojedine rute, s kojom bi se na karti s lakoćom moglo iscrtati put glatkom neisprekidanom krivuljom. Ali u ZET-ovim podacima se ne može pronaći shapes.txt te sam kao zaobi-

lazak tom problemu "laički" uzeo koordinate svih stanica na pojedinoj ruti i spojio ih ravnom linijom. Izgled te linije nije najljepši, ali je približno rješenje. Još jedan način koji sam probao je koristeći routing komponente, ali to rješenje također ima nedostataka, ali o tom više u poglavlju o frontend implementaciji 2.4.4.

stops.txt

```
stop_id,stop_code,stop_name,stop_desc,stop_lat,stop_lon,zone_id,stop_url,location_type,parent_station
"98_1","1","Črnomerec",,45.815002,15.934932,,,0,98
"98_2","2","Črnomerec",,45.815398,15.933924,,,0,98
"98_11","11","Črnomerec",,45.815127,15.934536,,,0,98
"98",,"Črnomerec",,45.815175,15.934464,,,1,
"99_23","23","Črnomerec",,45.815318,15.933231,,,0,99
"99_26","26","Črnomerec",,45.814933,15.933877,,,0,99
"99_31","31","Črnomerec",,45.816087,15.933219,,,0,99
```

Slika 2.7. Izgled stops.txt

Bitne informacije u stops.txt su:

- stop_id - identifikator pojedine stanice
- stop_name - naziv stanice
- stop_lat i stop_lon - koordinate stanice

Vrijedi napomenuti da nedostaju neki podaci, ali nama to nije bitno za implementaciju makar bi bilo korisno imati i stop_description za dodatne informacije.

stop_times.txt

```
trip_id,arrival_time,departure_time,stop_id,stop_sequence,stop_headsign,pickup_type,drop_off_type,shape_dist_traveled
"0_21_26820_268_10001","00:30:00","00:30:00","110_82",1,"V. Gorica",,,
"0_21_26820_268_10001","00:32:30","00:32:30","238_24",2,"V. Gorica",,,
"0_21_26820_268_10001","00:33:40","00:33:40","450_24",3,"V. Gorica",,,
"0_21_26820_268_10001","00:35:20","00:35:20","1085_24",4,"V. Gorica",,,
"0_21_26820_268_10001","00:36:20","00:36:20","277_24",5,"V. Gorica",,,
"0_21_26820_268_10001","00:37:20","00:37:20","1233_21",6,"V. Gorica",,,
"0_21_26820_268_10001","00:38:30","00:38:30","565_24",7,"V. Gorica",,,
```

Slika 2.8. Izgled stop_times.txt

Bitne informacije u stop_times.txt su:

- trip_id - identifikator putovanja na pojedinoj ruti
- arrival_time i departure_time - vrijeme dolaska i odlaska
- stop_id - identifikator pojedine stanice
- stop_sequence - redni broj stanice na putu

Ovo je najveća datoteka (oko 70MB, ~1000000 linija) koja sadrži za svako pojedino putovanje (trip_id) informacije o svim stanicama na putu, očekivanom vremenu dolaska i odlaska sa stanice i redoslijedu. Ona će biti ključna za iscrtavanje rute i predviđanju lokacije tramvaja.

routes.txt

```
route_id,agency_id,route_short_name,route_long_name,route_desc,route_type,route_url,route_color,route_text_color
1,0,"1","Zap.kol. - Borongaj",,0,,"ffffff","000000"
2,0,"2","Črnomerec - Savišće",,0,,"ffffff","000000"
3,0,"3","Ljubljanica -Savišće",,0,,"ffffff","000000"
4,0,"4","Savski most - Dubec",,0,,"ffffff","000000"
5,0,"5","Prečko-Park Maksimir",,0,,"ffffff","000000"
6,0,"6","Črnomerec - Sopot",,0,,"ffffff","000000"
```

Slika 2.9. Izgled routes.txt

Ovdje nam je jedina nova bitna informacija route_long_name koja nam daje puno ime rute (početno mjesto i krajnje mjesto)
To bi bile najbitnije GTFS static informacije koje sam ja upotrijebio za izradu aplikacije koji su bili jako bitni za izradu baze podataka koja sprema te odvojene datoteke kao tablice i spaja ih s odgovarajućim ključevima i time omogućuje lako i brzo pretraživanje te formuliranje odgovora 2.4.1.

ZET realtime podaci

Na ZET-ovoj stranici se nalazi link za GTFS realtime podatke 2.4. koji kada se skinu su binarna datoteka u .proto formatu. Protobuf je opisan u narednom poglavlju 2.3. Ovdje ćemo samo brzo proći kroz konačne podatke kada se obavi deserijalizacija.
Na slici 2.10. se nalazi jedan mali isječak od inače jako velike poruke (obično sadrže oko 500 - 600 entiteta), ali radi jednostavnosti je prikazan samo jedan.
Na početku svake poruke se nalazi header i možemo primijetiti da se koristi starija verzija GTFS 1.0, sada je aktualna 2.0 verzija koja ima bolje definiranu schemu s više informacija i pruža veću fleksibilnost. Također piše da se svaki put šalju svi podaci "FULL_DATASET" i na kraju je vremenska oznaka generiranja poruke koja je u UNIX formatu.
Zatim polje entity koje sadrži entitet (objekt) koji reprezentira informacije o jednom vozilu. Svaki od njih ima jedinstveni ID i tripUpdate. TripUpdate se sastoji od osnovnih informacija o putu na kojem je vozilo, ID tog puta, kada je započeo put, je li put prema rasporedu, broj rute kojoj pripada taj put, vremenska oznaka generiranja tih podataka i stopTimeUpdate koji sadrži informacije o zadnjoj posjećenoj stanici njen ID, redni broj i vremenska oznaka.
Jedna stvar koju je vrijedno napomenuti da su ZET-ovi podaci prilično osnovni tjst. sadrže najosnovnije informacije, vjerojatni razlog toga je što su ZET-ovi podaci u fazi razvoja i na stranici je navedeno da je ovo prototip. Nadam se da će malo više i detaljnije upotpuniti podatke, na primjer jedna jako velika stvar koja bi bila jako korisna jesu vehicle podaci (slika 2.11.) koji sadrže točne geografske lokacije vozila. Trenutačno je aplikacija napravljena tako da na karti prikaže zadnje potvrđene pozicije tramvaja koje su zapravo podaci iz stopTimeUpdate odnosno lokacije zadnje posjećenih stanica i markeri koji su predikcije položaja tramvaja temeljene na vremenima iz GTFS static podataka. Implementacija tih vehicle podataka bi izbacilo potrebu za predikcijom i puno olakšala direktno prikazivanje točnih lokacija tramvaja.

Slika 2.10. Dio ZET-ovog GTFS-rt feed-a

Slika 2.11. Primjer vehicle podataka

2.3. Protocol buffers

Protocol Buffers su jezično i platformski neutralni, proširivi mehanizmi za serijalizaciju strukturiranih podataka. Razvijeni od strane Googlea, prvotno su bili namijenjeni internoj uporabi, ali su kasnije postali dostupni pod otvorenom licencom [9]. Glavni cilj protocol buffera je pružiti jednostavnost i visoke performanse, te su posebno dizajnirani kako bi bili manji i brži od XML i JSON formata.

Protobuf koristi .proto datoteke za definiranje strukture podataka koji koriste posebnu sintaksu. Trenutno su podržane verzije proto2 i proto3 u kojima se s ključnom riječi message definira struktura podataka poruke. Na slici 2.12. je mali primjer jedne definicije poruke Person koja sadrži tri polja za podatke id, name i email. Svaka od njih mora imati tip podatka i vrijednost koja služi za identifikaciju polja kod binarne serijalizacije. Neki tipovi podataka koji su podržani su int32, int64, float, double, bool, string, enum i ugrađene poruke.

Slika 2.12. Primjer .proto definicije

Kada smo definirali strukturu svoje poruke potrebno je iskoristiti proto compiler (protoc) da bi generirali kod koji se koristi za serijalizaciju i deserializaciju podataka u odabranom programskom jeziku (C++, Java, Python, Go, JavaScript i mnogo drugih). Za primjer sa slike 2.12. i za programski jezik Python, protoc compiler generira modul person_pb2 koji sadrži potrebne metode za kreiranje objekata te strukture, serijalizaciju i deserijalizaciju tih podataka.

```python
import person_pb2

person = person_pb2.Person()
person.id = 123
person.name = "John Doe"
person.email = "johndoe@example.com"

# Serijalizacija
serialized_data = person.SerializeToString()

# Deserijalizacija
new_person = person_pb2.Person()
new_person.ParseFromString(serialized_data)
```

Slika 2.13. Python kod s generiranim proto modulom

Neke prednosti korištenja protocol buffer-a za razliku od tradicionalnijih opcija poput XML-a i JSON-a su:

- Protobuf koristi binarni format za serijalizaciju, koji je mnogo kompaktniji i brži za parsiranje.
- Dizajniran s backward i forwards compatibility na umu
- Cross-Language compatibility - međujezična kompatibilnost
- Imaju schema validation koji striktno čuvaju strukturu podataka

2.4. Arhitektura sustava

Sustav će se temeljiti na modelu Klijent-Poslužitelj. Korisnik će koristiti svoje računalo za pristup stranici putem web-preglednika, koji će potom komunicirati s mojom aplikacijom koja će se nalaziti na nekom poslužiteljskom računalu. Frontend dio aplikacije će biti kreiran pomoću React radnog okvira te će koristiti razne biblioteke koje olakšavaju programiranje različitih funkcija. Taj dio će korisnik moći vidjeti i koristit će za upravljanje našom aplikacijom. Backend dio će biti kreiran u JavaScriptu pomoću node.js i raznih paketa poput express.js i drugih. Također u backend dijelu aplikacije će biti PostgreSQL baza podataka koja će se koristiti za pretraživanje informacija o rutama koje preuzmemo s ZET-ove stranice.

etc...

2.4.1. Baza podataka

2.4.2. Pomoćne skripte

2.4.3. Backend - node.js

2.4.4. Frontend - REACT

2.5. Izgled aplikacije

3. Rezultati i rasprava

4. Zaključak

Literatura

[1] Zagrebački Električni Tramvaj, “ZET - O nama”, https://www.zet.hr/o-nama/259, [Stranica posjećena: 16.05.2024.].

[2] ——, “ZET - GTFS podaci”, https://www.zet.hr/odredbe/datoteke-u-gtfs-formatu/669, [Stranica posjećena: 27.05.2024.].

[3] ZET info, https://zet-info.com/.

[4] General Transit Feed Specification, “GTFS - About”, https://gtfs.org/.

[5] Mapnificent, https://github.com/mapnificent/mapnificent.

[6] General Transit Feed Specification, “GTFS - Schedule/Static”, https://gtfs.org/schedule/, https://gtfs.org/schedule/reference/#dataset-files. https://developers.google.com/transit/gtfs/reference.

[7] Trafiklab, “Informacije o GTFS schedule modelu i slika modela”, https://www.trafiklab.se/docs/using-trafiklab-data/using-gtfs-files/static-gtfs-files/.

[8] General Transit Feed Specification, “GTFS - Realtime”, https://gtfs.org/realtime/,https://gtfs.org/realtime/reference/.

[9] P. buffers.

Sažetak

Vizualizacija prometa javnog prijevoza u stvarnom vremenu uporabom formata GTFS

Luka Miličević

Unesite sažetak na hrvatskom.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Ključne riječi: prva ključna riječ; druga ključna riječ; treća ključna riječ

Abstract

Visualization of public transport traffic in real time using the format GTFS

Luka Miličević

Enter the abstract in English.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Keywords: the first keyword; the second keyword; the third keyword

Privitak A: The Code