

README

New York is one of the world's largest cities and its own set of public transportation challenges. The subway system of the city is of great significance to the city and it creates numerous problems if it slows down or worse breaks down. This assignment will use MTA's real time API to print alerts about train delays.

The MTA Real-Time Data feed provides information in the General Transit Feed Specification. More details can be found at <https://developers.google.com/transit/gtfs-realtime/>. Every time we poll this API we receive a message consisting of a header and an entity. We are concerned with entities. Each entity shall consist of various sub entities like "trip_update", "vehicle" and "alerts".

Alerts are generated when trains are delayed. Each alerts will contain an "informed_entity" field. This field is the list of one or more "trips" that may be delayed. Each "trip" consists of a "trip_id" and a "route_id". This is what shall be displayed on the terminal and the web page after minimal filtering. The following is an example of an alert in an entity:

```
informed_entity {  
  trip {  
    trip_id: "094200_1..N02R"  
    route_id: "1"  
  }  
}  
header_text {  
  translation {  
    text: "Train delayed"  
  }  
}
```

As you can see, the header text says "Train Delayed", alerts such as the above one are only generated if there is an actual delayed train i.e. it corresponds to the real life event when we see a 'Train Delayed' sign on a billboard at a subway station in NYC.

The subway transit system of New York is quite efficient and hence train delay alerts occur only once in 10-15 minutes and the alert is published by the API for 20-50 seconds. In our assignment we poll the API every two seconds, hence once an alert is published by the API the application will print the alert along with the time at which it was polled. As such we expect to encounter 10-15 alerts (usually about the same trip) every 10-15 minutes. This is conditional though, if on a particular day there are several trains that are delayed we can expect an even larger volume of alerts that will be received.

To run the application please make sure you have all the modules of requirement.txt, use:

"python app.py" from command line

Then type <http://localhost:5000/> in your Chrome browser to establish connection between client page and server.

Credits: Miguel Grinberg's tutorial: <http://blog.miguelgrinberg.com/post/easy-websockets-with-flask-and-gevent>