TILE 38

GEOSPATIAL GEOFENCING

INTRODUCTION

- Tile38 è un progetto open source dedicato al “realtime geospatial e geofencing”: è un server che memorizza informazioni geospaziali, come ad esempio delle coordinate, ed esegue operazioni in tempo reale con esse. E utilizzata per la localizzazione fisica di utenti o oggetti entro un certo perimetro, associato a un’area geografica del mondo reale. Nell’impiego comune potrebbe essere utilizzata come applicazione per smartphone dedicata per l’utilizzo nel settore marketing. Il funzionamento e interfacciamento principale avviene tramite console di comando e client. L’utilizzo di Tile38 avviene attraverso la creazione ed esecuzione di comandi; per usarli è necessario conoscere due importanti elementi: l’operazione che vogliamo eseguire e gli oggetti che vogliamo utilizzare. Vediamo prima gli oggetti, poi i comandi veri e propri. Tile 38 supporta una varietà di tipi di oggetti suddivisi in tre categorie principali: coordinate di singoli punti, aree di delimitazione e coordinate complesse.

OBJECT TYPES

- Lat/lon point: Punto singolo.
- Bounding box: Aree.
- Geohash (stringhe), GeoJSON (formato standard del settore), XYZ Tile, QuadKey: Coordinate complesse.

Per quanto riguarda i protocolli network, si utilizza una libreria client o Tile38 CLI, ma ci sono momenti in cui è disponibile solo HTTP o quando è necessario eseguire il test da un terminale remoto. In questi casi si forniscono opzioni HTTP e telnet.

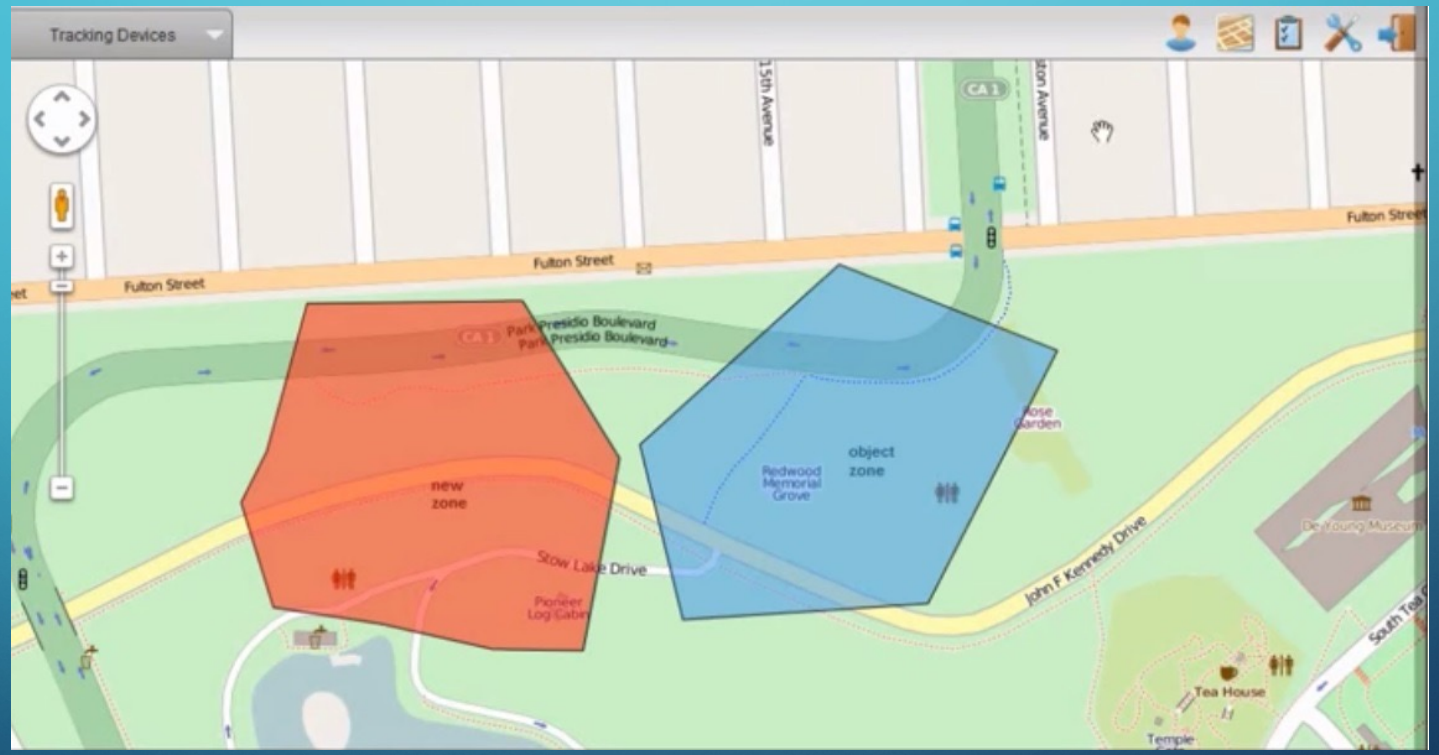
MAIN API

Tile38 possiede una moltitudine di comandi, noi vedremo solo quelli più importanti, ossia quelli di inserimento, cancellazione e ricerca. La sintassi e struttura dei comandi per Tile38 è mantenuta invariata fra le varie modalità di client, ad esempio fra la console di comando docker e il client java. In seguito è espressa in ordine la struttura da conoscere per creare un comando Tile38 per l'inserimento:

- 1: comando: keyword che determina l'operazione che vogliamo compiere.
- 2: collezione: questa variabile (opzionale) determina il nome della collezione di oggetti di cui farà parte l'elemento.
- 3: elemento: variabile che individua il nome del soggetto dell'operazione.
- 4: campo: variabile opzionale da inserire se si vuole specificare ulteriori campi, come ad esempio la velocità. Verrà descritta meglio successivamente.
- 4: oggetto: keyword che assegna un tipo di oggetto all'elemento inserito prima (visto precedentemente).
- 5: coordinata: variabili numeriche da inserire per specificare le coordinate e informazioni riguardanti l'elemento.

Esempio: SET fleet truck1 POINT 33.5123 -112.2693 245.0

GEOFENCING


Tracking Devices
15th Avenue
Fulton Street
Fulton Street
Presidio Boulevard
new
zone
object
zone
Redwood
Memorial
Grove
Rose
Garden
Stow Lake Drive
John F Kennedy Drive
Tea House
Temple

FIELDS AND SEARCHING

- Fields: I campi (o ”fields”) sono dati aggiuntivi che appartengono a un oggetto. Un campo `e sempre un punto mobile a doppia precisione. Non esiste un limite al numero di campi che un oggetto può avere. Per impostare un campo quando si imposta un oggetto:

```
set fleet truck1 field speed 90 point 33.5123 -112.2693
```

- Where: : Questa opzione consente di filtrare i risultati in base ai valori dei campi. Ad esempio:

```
nearby fleet where speed 70 +inf point 33.462 -112.268 6
```

REDIS SERVER

Tile38 comunica con il server Redis utilizzando un protocollo denominato RESP (REdis Serialization Protocol). Mentre il protocollo è stato progettato specificamente per Redis, può essere utilizzato per altri progetti software client-server. Redis è un DBMS NoSQL di tipo “key/value storage”. Esso si basa infatti su una struttura a dizionario: ogni valore immagazzinato è abbinato ad una chiave univoca che ne permette il recupero. Tile38 usa il protocollo RESP Redis in modo nativo. Pertanto, la maggior parte dei client che supportano i comandi Redis di base supporterà a sua volta Tile38.

TESTS

Alcune semplici operazioni di base: aggiungiamo 2 punti chiamati ”truck1” e ”truck2” ad una collezione chiamata ”fleet”, per poi eseguire una ricerca spaziale entro una certa area e avere come risultato uno dei due punti; infine eliminiamo tutti i punti.

```
#creazione punti
> set fleet truck1 point 33.5123 -11
> set fleet truck2 point 33.4626 -11

# ricerca
> scan fleet
> nearby fleet point 33.462 -112.268

# ritorno e cancellazione punti
> get fleet truck1
> del fleet truck2
> drop fleet
```

TESTS

```
SETCHAN warehouse NEARBY fleet FENCE POINT 33.462 -112.268 6(
SUBSCRIBE warehouse
```

```
SET fleet bus POINT 33.460 -112.260
```

```
"command": "set",
  "group": "5c5203ccf5ec4e4f349fd038",
  "detect": "inside",
  "hook": "warehouse",
  "key": "fleet",
  "time": "2019-01-30T13:06:36.769273-07:00",
  "id": "bus",
  "object":  "type": "Point", "coordinates": [-112.26, 33.46]
```

CONCLUSION

- Il progetto ha soddisfatto gli obiettivi posti inizialmente, sono stati toccati tutti i punti di interesse, dalle caratteristiche principali di Tile38, fino ai protocolli network e client alla base di vari funzionamenti. E' stata vista la versatilità di questo software, soprattutto per quanto riguarda l' interoperabilità dei vari client che il protocollo Redis supporta; nel nostro caso abbiamo visto i client Python e Java.