Rapport TLC

Mahmoud JAMMAL

Mouhyi MOURAJJI

M2 ILA

Introduction:

Le but de ce projet est de découvrir la solution (PaaS) et comprendre ses bénéfices dans la programmation.

Nous avons choisi les technologies Google AppEngine il est accessible à ce lien

https://confident-inn-224415.appspot.com/api/run et Google DataStore pour la persistance des données

Nous avons créé une application BackEnd pour enregistrer des informations liées à la course de l'utilisateur et pour cela nous avons utilisé NodeJS pour pouvoir profiter de ses libraries.

Pour interagir avec notre application, nous avons créé un service HTTP REST pour :

- Ajouter un ou plusieurs Record avec une seule requête.
- Supprimer un ou plusieurs Record par leurs Ids
- Rechercher les records par : id, username, date(timestamp), position(l'altitude, longitude) ou par une combinaison des combinaisons suivantes :
  - Nom, Position, Date
  - Nom, Position
  - Nom, Date
  - Id, Date

Bulk Add:

Pour éviter la perte des données chaque record créées est enregistrée directement dans l DataStore, même si on envoie plusieurs records à la fois.

Exemple :

Type de requete : POST
Header : Content-Type application/json
Body :

```
[
    {
        "id":145,
        "lat":47.8601,
        "long":2.8507,
        "user":"john",
        "timestamp":154379267
    },
    {
        "id":145,
        "lat":47.8631,
        "long":2.8517,
        "user":"john",
        "timestamp":154379278
    },
    {
        "id":145,
        "lat":47.8721,
        "long":2.8637,
        "user":"john",
        "timestamp":154379967
    }
]
```

une fois les données sont enregistrées on reçoit en réponse : `{"add": "done"}`.

Search :

Les records enregistrés sont en accès publics les recherches sont fait par une combinaison comme c’est expliqué dans l’introduction ou par un seul element.

Exemple :

Type de requete: GET
Header : Content-Type application/json
Requete : https://confident-inn-224415.appspot.com/api/run?user=lea&timestamp=1543775726,1543775729

Bulk Delete:

Un utilisateur peut supprimer un ou plusieurs records à la fois.

Exemple :

Type de requete: DELETE
Header : Content-Type application/json
Requete : https://confident-inn-224415.appspot.com/api/run/13,14

où 13 et 14 sont des Ids, une fois la requete est terminée nous aurons la réponse : {"delete": "done"}.

Analyses:

Tests effectués :

Pour tester les requêtes nous avons utilisé le script fournis test.sh.

Pour tester la sécabilité de notre Backend sur AppEngine nous avons utilisé l’outil Hey pour le benchmark.

Requêtes envoyées et leurs résultats :

Requete avec 200 fois POST:

Type de requete: POST
Header : Content-Type application/json
Body :

```
[
    {
        "id":13,
        "lat":33.8601,
        "long":"2.8507",
        "user":"alice",
        "timestamp":1543778727
    }
]
```

Requete : hey -m POST -H "Content-Type: application/json" -D ./TLC/TLC/lab1/test.json https://confident-inn-224415.appspot.com/api/run
Remarque : l’outil Hey envoie 200 requetes par défaut et c’est le cas dans ce test.

Reponse :

Summary:

Total: 3.6441 secs

Slowest: 3.5667 secs

Fastest: 0.0215 secs

Average: 0.2491 secs

Requests/sec: 54.8832

Total data: 2800 bytes

Size/request: 14 bytes

Response time histogram:

0.022 [1] |
0.376 [196]|■■
0.731 [0] |
1.085 [0] |
1.440 [0] |
1.794 [0] |
2.149 [0] |
2.503 [0] |
2.858 [0] |
3.212 [0] |
3.567 [3] |■

Latency distribution:

10% in 0.1221 secs
25% in 0.1669 secs
50% in 0.2059 secs
75% in 0.2332 secs
90% in 0.2983 secs
95% in 0.3095 secs
99% in 3.3572 secs

Details (average, fastest, slowest):

DNS+dialup: 0.0225 secs, 0.0215 secs, 3.5667 secs
DNS-lookup: 0.0021 secs, 0.0000 secs, 0.0109 secs
req write: 0.0002 secs, 0.0000 secs, 0.0038 secs
resp wait: 0.2262 secs, 0.0213 secs, 3.4733 secs
resp read: 0.0002 secs, 0.0001 secs, 0.0005 secs

Status code distribution:

[200] 200 responses

Requete avec 2000 fois POST:

Type de requete: POST

Header : Content-Type application/json

Body :

```
[
  {
    "id":13,
    "lat":33.8601,
    "long":"2.8507",
    "user":"alice",
    "timestamp":1543778727
  }
]
```


Requete : hey -n 2000 -m POST -H "Content-Type: application/json" -D ./TLC/TLC/lab1/test.json https://confident-inn-224415.appspot.com/api/run

Reponse :

Summary:

```
  Total:    11.9488 secs
  Slowest:  4.5197 secs
  Fastest:  0.0219 secs
  Average:  0.2347 secs
  Requests/sec: 167.3808
  Total data:   28000 bytes
  Size/request: 14 bytes
```

Response time histogram:

```
  0.022 [1] |
  0.472 [1962]  |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.921 [28]    |■
  1.371 [0] |
  1.821 [0] |
  2.271 [0] |
  2.721 [0] |
  3.170 [0] |
  3.620 [1] |
  4.070 [4] |
  4.520 [4] |
```

Latency distribution:

```
  10% in 0.0895 secs
  25% in 0.1207 secs
  50% in 0.1990 secs
  75% in 0.3034 secs
  90% in 0.3752 secs
  95% in 0.4013 secs
  99% in 0.5716 secs
```

Details (average, fastest, slowest):

```
  DNS+dialup:   0.0029 secs, 0.0219 secs, 4.5197 secs
  DNS-lookup:   0.0004 secs, 0.0000 secs, 0.0184 secs
  req write:    0.0001 secs, 0.0000 secs, 0.0014 secs
  resp wait:    0.2315 secs, 0.0215 secs, 4.5194 secs
  resp read:    0.0002 secs, 0.0000 secs, 0.0016 secs
```

Status code distribution:

```
  [200] 2000 responses
```

Problèmes rencontrés :

Dans la partie recherchée nous avons eu des difficultés car nous ne pouvons pas faire des requêtes où on peut définir les conditions de recherche par ex :

SELECT FROM Record WHERE username = ‘alice’ and id = 13

Alors nous avons dû mettre à jour le fichier index.yaml pour mettre les combinaisons de recherche souhaitée.