Homework 3 - Tecnologie Cloud & Mobile 2020

1040886 - Marco Adobati
1040394 - Valerio Sabato

Lambda functions implementate

Le lambda functions implementate sono state:

- Get_Watch_Next_by_Idx: per permettere di visualizzare i talk successivi (*url, id*) dato un id di un talk
- Get_taks_two_dates: per permettere di visualizzare tutti i talk presenti tra due date

Modifica Job aws glue

La prima lambda function implementata è stata Get_Watch_Next_by_Idx la quale ritorna *url* e *id* dei video successivi, tuttavia per poter farla funzionare al meglio e alla luce di alcune criticità(aka url duplicati) mostrate da alcuni compagni durante la scorse presentazioni abbiamo modificato il job di aws glue, in particolare una parte di una query pyspark.

```
watch_dataset_agg = watch_dataset.groupBy(col("idx").alias("idx_ref")) \
                    .agg(collect_set("watch_next_idx").alias("wnext"), \
                    collect_set("url").alias("wnurl"))
```

In particolare abbiamo utilizzato la funzione *collect_set* per evitare di inserire i duplicati all'interno del database.

LF Get_Watch_Next_by_Idx

Qui riportato c'è l'implementazione della query per mongodb nella prima LF:

```
connect_to_db().then(() => {
    console.log('=> get_all talks');
    talk.find({ "_id" : body.idx}).select('wnurl wnext')
        .skip((body.doc_per_page * body.page) - body.doc_per_page)
        .limit(body.doc_per_page)
        .then(talks => {
                callback(null, {
                    statusCode: 200,
                    body: JSON.stringify(talks)
                })
            }
        )
```

Abbiamo scelto di utilizzare il seguente metodo di implementazione perchè abbiamo convenuto che sia il più efficiente senza andare a sporcare troppo la query, andando così semplicemente a selezionare gli *url* e gli *id* dei video successivi dato l'*_id* del talk in input. Notare che *_id* viene ritornato automaticamente dalla query (vedasi slide successiva).

Test Get_Watch_Next_by_Idx

Per testare abbiamo utilizzato *curl,* ad esempio:

curl -d '{"idx":"29d3c061268bac83b4333ec691ad80f5", "doc_per_page": 10, "page" : 1}' -H "Content-Type: application/json" -X POST https://igb33htah2.execute-api.us-east-1.amazonaws.com/default/Get_Watch_Next_by_Idx > test.json

Il quale ha ritornato:


```
▼ 0:
  ▼ wnext:
      0:    "206c240bcadd34f1e16be95b6ef1cc37"
      1:    "2654f85bba4eb9e037874d929f06c6fb"
      2:    "209f537631d5c97e7086b71b487f513f"
      3:    "e606a04def48df8ee4d2c776a4674909"
      4:    "f6606433c84f2180ad2c6fd2ff1729d4"
      5:    "9f7b1654e792011b7e1c6f4288520226"
      6:    "21d95d0e8ba36acb317c71c858e42cb9"
  ▼ wnurl:
    ▼ 0:    "https://www.ted.com/talks/aimee_mullins_my_12_pairs_of_legs"
    ▼ 1:    "https://www.ted.com/talks/david_sengeh_the_sore_problem_of_prosthetic_limbs"
    ▼ 2:    "https://www.ted.com/session/new?context=ted.www%2Fwatch-later"
    ▼ 3:    "https://www.ted.com/talks/christoph_keplinger_the_artificial_muscles_that_will_power_robots_of_the_future"
    ▼ 4:    "https://www.ted.com/talks/murat_dalkilinc_the_benefits_of_good_posture"
    ▼ 5:    "https://www.ted.com/talks/hugh_herr_the_new_bionics_that_let_us_run_climb_and_dance"
    ▼ 6:    "https://www.ted.com/talks/jeffrey_siegel_what_makes_muscles_grow"
      id:   "29d3c061268bac83b4333ec691ad80f5"
```

LF Get_talk_two_dates

La seconda LF implementata è la Get_talk_two_dates.

Questa funzione consente di fornire tutti i video presenti tra due date (fornite dall'utente), ordinate in ordine crescente di data.

In questo modo l'utente potrà vedere i video pubblicati in un determinato periodo di tempo senza dover cercare manualmente.

Di seguito la query effettuata:

```
connect_to_db().then(() => {
    console.log('=> get talks between two dates');
    talk.find({ data : {$gt : body.data1, $lt : body.data2}}).select('main_speaker title details data').sort({'data': 1})
        .skip((body.doc_per_page * body.page) - body.doc_per_page)
        .limit(body.doc_per_page)
        .then(talks => {
                callback(null, {
                    statusCode: 200,
                    body: JSON.stringify(talks)
                })
            }
        )
```

Test: Get_talk_two_dates

Per testare abbiamo ancora utilizzato curl, ad esempio:

curl -d '{"data1":"2019-01-01", "data2" : "2019-04-01" ,"doc_per_page": 150, "page" : 1}' -H "Content-Type: application/json" -X POST https://rwrx3ngxa3.execute-api.us-east-1.amazonaws.com/default/Get_talk_two_dates > test.json

Il quale ha ritornato (per esigenze grafiche mostriamo solo i primi due elementi):


```
▼ 0:
    _id:            "766656b5121b60bddcd2767caaa52e3f"
    main_speaker:   "Aja Monet and phillip agnew"
    title:          "A love story about the power of art as organizing"
  ▶ details:        "In a lyrical talk full o…ion, anger and anxiety."
    data:           "2019-02-01T00:00:00.000Z"
▼ 1:
    _id:            "929341c2d1f35757a316e1b914557c58"
    main_speaker:   "Yannick Roudaut"
    title:          "How today's truths shape tomorrow's possibilities"
  ▶ details:        "For centuries, we believ…nd another renaissance."
    data:           "2019-02-01T00:00:00.000Z"
```

Criticità tecniche

- La criticità più “importante” è il corretto uso degli indici all’interno di mongodb, il quale se non correttamente effettuato può portare a malfunzionamenti delle LF
- Se la data non è nel formato corretto(ovvero ‘yyyy-mm-dd’) Get_talk_two_dates non funziona correttamente
- Maggiore dati richiesti in *doc_per_page* maggiore sarà il tempo di attesa per l’utente (come ovvio che sia)

Possibili evoluzioni

Una possibile evoluzione della Get_Watch_Next_by_Idx potrebbe essere:

- Aggiungere un numero massimo di video successivi suggeriti
- La possibilità di scegliere i tag dei video successivi, in questo modo si potrebbe fare una scrematura migliore

Mentre una possibile evoluzione della Get_talk_two_dates è:

- Aggiungere un controllo sulle date affinchè siano nel formato corretto
- Aggiungere altri formati accettati nel campo *data1* e *data2*, oltre a quelli già accettati

Api esposte

Get_Watch_Next_by_Idx

- https://igb33htah2.execute-api.us-east-1.amazonaws.com/default/Get_Watch_Next_by_Idx

Get_talk_two_dates

- https://rwrx3ngxa3.execute-api.us-east-1.amazonaws.com/default/Get_talk_two_dates