



TalkWithTed

Mehmood Zakria - Technologie Cloud & Mobile 2024

Homework 3

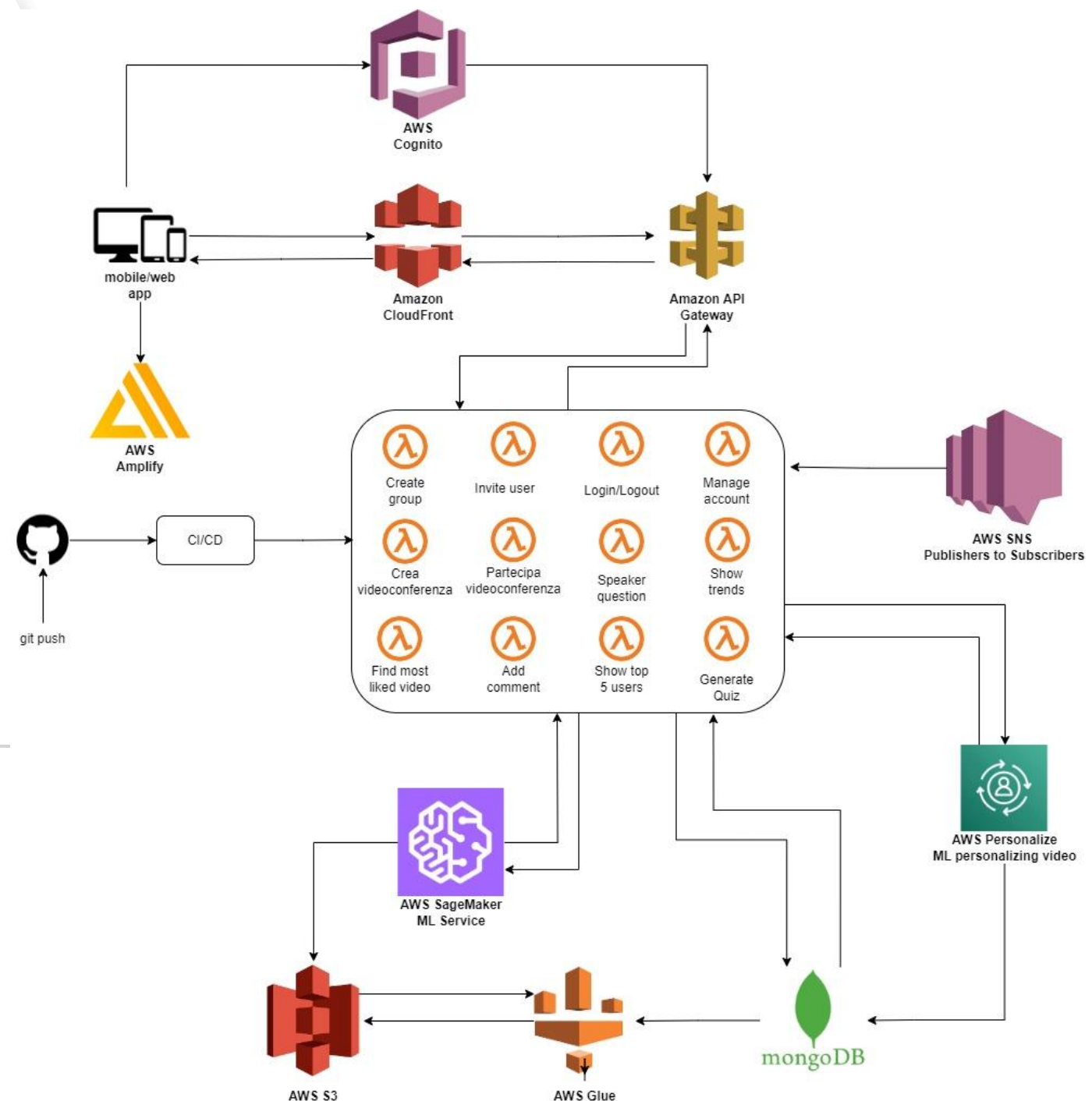
Architettura

Prima:

- ✓ Caricamento dati su S3 + mongoDB Atlas
- ✓ Implementazione Job PySpark mediante AWS Glue

Ora:

- Implementazione lambda function
- Esposizione API mediante Postman



Handler.js

WatchNext

Operazioni:

- Controllo se _id esiste
- Ricerca video relativi per id
- Formattazione Json con relativi campi per output

```
connect_to_db().then(() => {
  console.log('=> get_all talks');
  talk.find({_id: body.id})
    .skip((body.doc_per_page * body.page) - body.doc_per_page)
    .limit(body.doc_per_page)
    .then(talks => {
      talk.find({_id: talks[0].watch_next})
        .then(t => {
          callback(null, {
            statusCode: 200,
            body: JSON.stringify({id: t[0]._id, title: t[0].title, details: t[0].details, url: t[0].url, watch_next: t[0].watch_next, speaker: t[0].speaker})
          })
        })
      })
    })
  .catch(err =>
    callback(null, {
      statusCode: err.statusCode || 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the talks.'
    })
  );
});
```

Esempio visualizzazione dati:

```
{"_id": "526880"}
```

```
_id: "526880"  
slug: "george_zaidan_how_do_gas_masks_actually_work"  
speakers: "George Zaidan"  
title: "How do gas masks actually work?"  
url: "https://www.ted.com/talks/george_zaidan_how_do_gas_masks_actually_work"  
▼ collect_list(struct(related_id AS watch_next, presenterDisplayName AS speaker)): Array (3)  
  ▼ 0: Object  
    watch_next: "109914"  
    speaker: "Stephanie Honchell Smith"  
  ▶ 1: Object  
  ▶ 2: Object
```

Output WatchNext

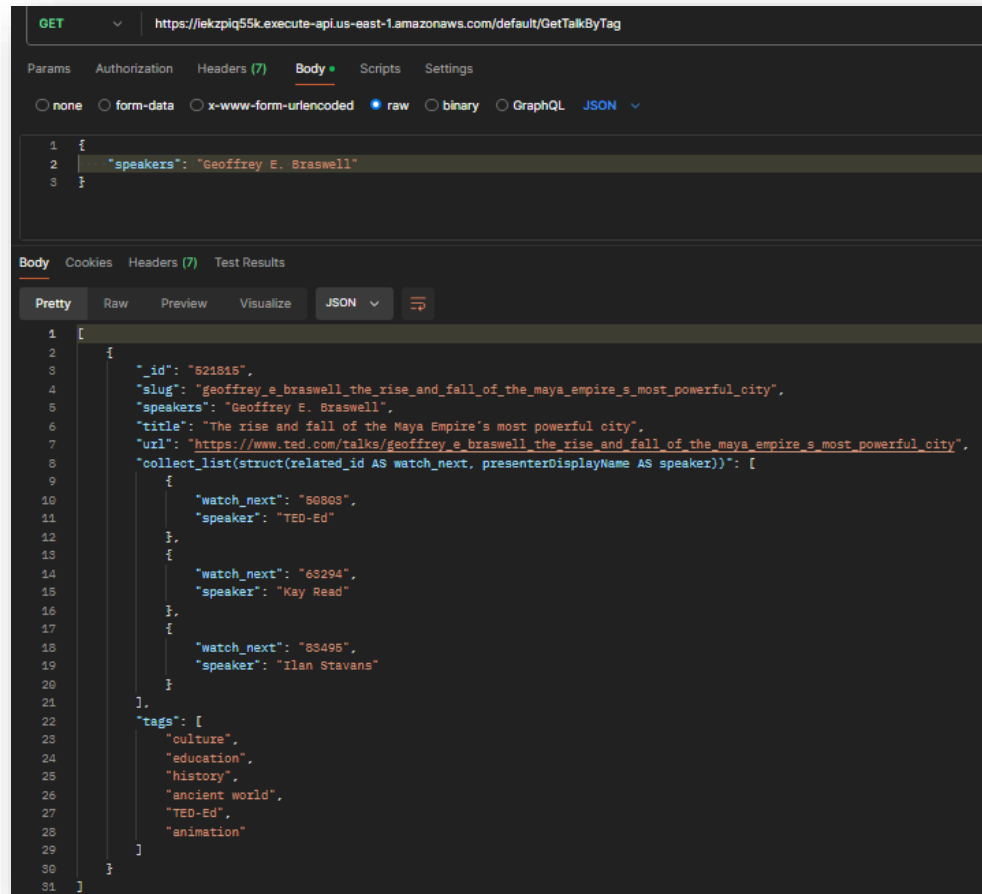
TedxSpeaker

- Connessione a mongodbAtlas
- Controllo se speaker esiste
- Ricerca talk per speaker
- Formattazione output in JSON

```
const talk_schema = new mongoose.Schema({
  _id: String,
  title: String,
  url: String,
  description: String,
  speakers: String,
  watch_next: String
}, { collection: 'tedx_data' });
```

```
connect_to_db().then(() => {
  console.log('=> get_all talks');
  talk.find({speakers: body.speakers})
    .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)
      })
    })
  .catch(err =>
    callback(null, {
      statusCode: err.statusCode || 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the talks.'
    })
  );
});
```

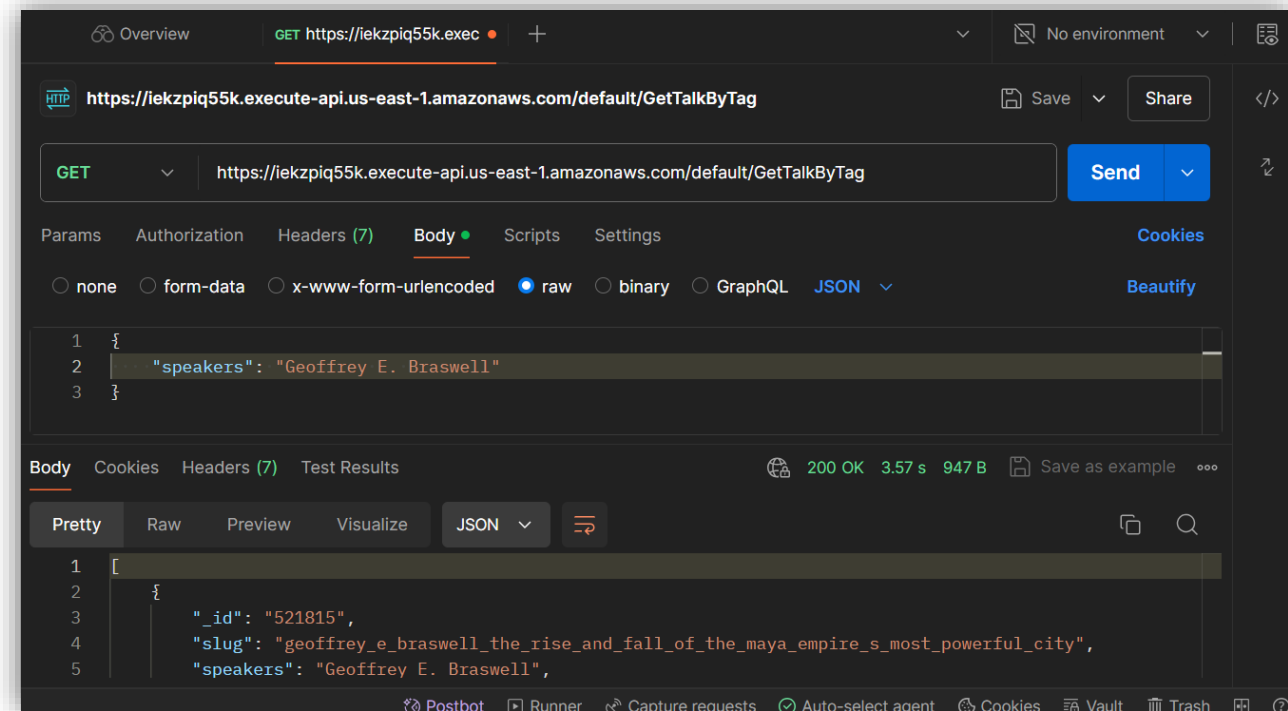
Output Speaker



A screenshot of a REST client interface. The top bar shows a GET request to `https://iekzpiq55k.execute-api.us-east-1.amazonaws.com/default/GetTalkByTag`. The 'Body' tab is selected, showing a JSON input: `{ "speakers": "Geoffrey E. Braswell" }`. The bottom section shows the response in 'Pretty' JSON format, displaying a list of talks with details like ID, slug, speakers, title, URL, and a list of related talks.

```
1 {
2   "speakers": "Geoffrey E. Braswell"
3 }
```

```
1 [
2   {
3     "_id": "521815",
4     "slug": "geoffrey_e_braswell_the_rise_and_fall_of_the_maya_empire_s_most_powerful_city",
5     "speakers": "Geoffrey E. Braswell",
6     "title": "The rise and fall of the Maya Empire's most powerful city",
7     "url": "https://www.ted.com/talks/geoffrey_e_braswell_the_rise_and_fall_of_the_maya_empire_s_most_powerful_city",
8     "collect_list(struct(related_id AS watch_next, presenterDisplayName AS speaker))": [
9       {
10        "watch_next": "50803",
11        "speaker": "TED-Ed"
12      },
13      {
14        "watch_next": "63294",
15        "speaker": "Kay Read"
16      },
17      {
18        "watch_next": "63496",
19        "speaker": "Ilan Stavans"
20      }
21    ],
22     "tags": [
23       "culture",
24       "education",
25       "history",
26       "ancient world",
27       "TED-Ed",
28       "animation"
29     ]
30   }
31 ]
```



A screenshot of a REST client interface. The top bar shows a GET request to `https://iekzpiq55k.execute-api.us-east-1.amazonaws.com/default/GetTalkByTag`. The 'Body' tab is selected, showing a JSON input: `{ "speakers": "Geoffrey E. Braswell" }`. The bottom section shows the response in 'Pretty' JSON format, displaying a partial JSON object with fields like ID, slug, and speakers.

```
1 {
2   "speakers": "Geoffrey E. Braswell"
3 }
```

```
1 [
2   {
3     "_id": "521815",
4     "slug": "geoffrey_e_braswell_the_rise_and_fall_of_the_maya_empire_s_most_powerful_city",
5     "speakers": "Geoffrey E. Braswell",
```




Criticità

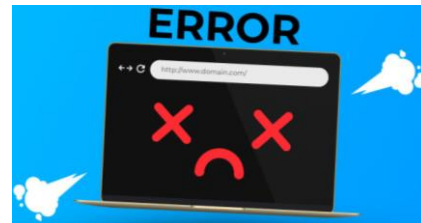
- Debugging



- Poca esperienza con linguaggio node js



- Molti problemi di connessione a database, causati anche da piccoli errori di codice



Evoluzioni



- Lambda function per l'ordinamento speaker per numero di talk

Follow us on



- Lambda function per la ricerca dei contatti per speaker



- Implementazione lambda function per generazione quiz



Quiz



Grazie

Mehmood Zakria

https://github.com/zakria-mehmood/unibg_cloud_e_mobile_2024

z.mehmood@unibg.it

<https://trello.com/invite/b/PhIMClv5/ATTI535o6od83afb7092dd5d9ce5fage7331B166BE6E/talkwithted>

