

UNITEDx

Trova il percorso di studi più adatto
ai tuoi interessi.



Marco Abbadini (1048650)
Lorenzo Conti (1046163)
Fabio Sangregorio (1046566)



Lambda Functions

Lambda Function: Get_Watch_Next_By_Id

```
1 connect_to_db().then(() => {
2   talk.find({ "_id": body.idx })
3     .skip((body.doc_per_page * body.page) - body.doc_per_page)
4     .limit(body.doc_per_page)
5     .then(talks => {
6       const watch_next_ids = talks[0].watch_next;
7       talk.find({ _id: { $in: watch_next_ids }})
8         .then(watchnext => {
9           callback(null, {
10             statusCode: 200,
11             body: JSON.stringify(watchnext)
12           })
13         })
14     }.catch(err =>
15       callback(null, {
16         statusCode: err.statusCode || 500,
17         headers: { 'Content-Type': 'text/plain' },
18         body: err
19       })
20     );
21 })
22 .catch(err =>
23   callback(null, {
24     statusCode: err.statusCode || 500,
25     headers: { 'Content-Type': 'text/plain' },
26     body: err
27   })
28 );
29 });
30 };
```

Lambda Function: Find_similar_talks

Restituisce i talk consigliati in base ad un testo, ad esempio la descrizione di un corso. I talk vengono selezionati in base alla similarità tra query e descrizione, titolo e autore

```
def lambda_handler(event, context):

    s3 = boto3.client('s3')

    text_query = event["query"]

    # load model
    nn = s3.download_fileobj(Fileobj=fp, Bucket="<bucket-name>", Key="model.pkl")

    # load vectorizer
    count_vectorizer = s3.download_fileobj(Fileobj=fp, Bucket="<bucket-name>", Key="vectorizer.pkl")

    # load id_map
    id_map = s3.download_fileobj(Fileobj=fp, Bucket="<bucket-name>", Key="id_map.json")

    # get predictions
    vectorized = count_vectorizer.transform([text_query])
    recommendations = nn.kneighbors(vectorized, return_distance=False).reshape(-1)

    return [id_map[r] for r in recommendations]
```

Lambda Functions: Python packages

Creazione file .zip contenente i moduli necessari

Esecuzione docker container

```
docker run --rm -it -v ${PWD}:/var/task  
lambci/lambda:build-python3.8 bash
```

Template della cartella da creare

```
- python  
- lib  
  - python3.6  
    - site-packages  
      - <python packages here>
```

Installazione pacchetti

```
pip install -r requirements.txt --no-deps -t  
python/lib/python3.8/site-packages/
```

Compressione .zip

Il contenuto decompresso del file .zip non deve superare i 250MB

Caricamento del file .zip in un bucket S3

Creazione di un layer caricando il file .zip dal bucket

Assegnazione del layer alla lambda function

API Gateway

Chiamata attraverso l'**API** per ottenere i watch next dei talk

Ciascun talk è descritto dai campi che lo caratterizzano con l'aggiunta dell'array di id dei suoi watch next video.



```
1 [
2   {
3     "watch_next": [
4       "44b84f6420cfcf3036d9594c05b74477",
5       "5bd34fcc55d9e1267f605fa0c060d54e",
6       "bf369621049995050d9564a56dd49693",
7       "aebf2037f780baf51230a59db2940a0e",
8       "588319aec061eaaaf3d4a1d0c8ee74d1e",
9       "9f7b1654e792011b7e1c6f4288520226",
10      "85d8ff6f7f4dcc797e848350db273542"
11    ],
12    "_id": "078766d6cc461cf71d45dc268b66db95",
13    "details": " ... ",
14    "main_speaker": "Will Hurd",
15    "posted": "Posted Sep 2019",
16    "tags": [
17      "TED", "talks", "immigration", "government", "United States",
18      "refugees", "society", "social change", "politics", "community"
19    ],
20    "title": "A wall won't solve America's border problems",
21    "url": " ..."
22  },
23  ...
24 ]
```

Get_Watch_Next_By_Id API Result

Criticità

Dipendenze esterne

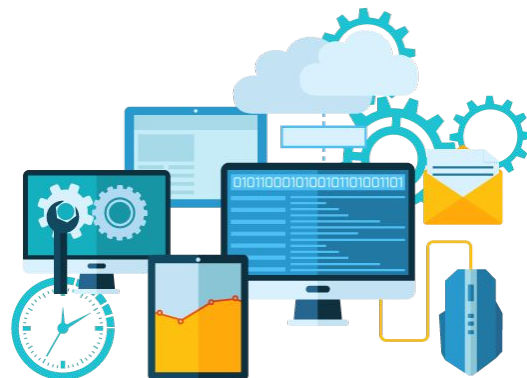
L'inclusione di pacchetti esterni non si è dimostrata sempre facile da effettuare (es. Max size per importare le dipendenze).

Lingua delle descrizioni

Alcuni talk presentano descrizioni in lingua diversa da quella inglese.

Recommender System

Non sempre i risultati sono del tutto pertinenti alle descrizioni dei corsi inseriti. Possono trattare gli stessi argomenti ma in ambiti diversi.



Possibili evoluzioni

Ottimizzazione dell'algoritmo di Machine Learning per la raccomandazione dei talk.



Implementazione dell'algoritmo di raccomandazione tramite AWS SageMaker.

Raccolta dei dati sui talk preferiti da ciascun utente.



Implementazione un modello che consigli dei corsi di studio sulla base dei talk preferiti.



**Amazon
SageMaker**

UNITEDx



[Board](#)

[GitHub](#)

