

TEDBooks4You

books worth spreading

Powered by:

TED^xunibg

1081058 - Bellosi Jacopo

1078817 - Poloni Luca

Lambda - 1 (getID)

Nella prima funzione Lambda abbiamo modificato il file handler.js per ottenere tutti gli ID correlati ad un determinato video.

```
connect_to_db().then(() => {
  console.log('=> get_all talks');
  talk.find({ id_related: body.id }, { id_related_list: 1, title_related_list: 1 })
    .then(talks => {
      if (!talks) {
        throw new Error('Talk not found');
      }
      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.'
      })
    );
});
```

Chiamata API



```
1  {
2    ... "id": "525470"
3  }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ↕

```
3    "id_related_list": [
4      "127367",
5      "119204",
6      "125757",
7      "622",
8      "1412",
9      "192"
10   ],
11   "title_related_list": [
12     "Let your garden grow wild",
13     "A cleaner world could start in a rice field",
14     "How poop turns into forests",
15     "One seed at a time, protecting the future of food",
16     "The other inconvenient truth",
17     "A critical look at geoengineering against climate change"
```


Nuovo DB

Abbiamo creato una nuova collezione con tutti i tag trovati da una “cronologia”.
Li abbiamo contati e incluso
L’id di ogni video trovato.

```
  _id: ObjectId('664473f96edd176315934c90')
  ▼ all_tags_info : Array (356)
    ► 0: Object
    ► 1: Object
    ► 2: Object
    ► 3: Object
    ► 4: Object
    ► 5: Object
    ► 6: Object
    ► 7: Object
    ▼ 8: Object
      tag : "wildlife"
      tag_count : 2
      ▼ videos : Array (2)
        ▼ 0: Object
          video_id : "524079"
        ► 1: Object
    ► 9: Object
    ► 10: Object
```

Lambda - 2 (getBooks)

La seconda lambda function che abbiamo fatto ha il compito di consigliare i libri appartenenti al dataset di Google Books. L'associazione è resa possibile dal confronto con i tag dei video della cronologia dell'utente.



```
User.findOne({ _id: body.id })
  .then(user => {
    if (!user) {
      throw new Error('User not found');
    }
    const allTagsInfo = user.all_tags_info;




    const tagsWithCounts = allTagsInfo.map(tagInfo => ({
      tag: tagInfo.tag,
      count: tagInfo.tag_count
    }));

    const sortedTags = tagsWithCounts.sort((a, b) => b.count - a.count);
    const topTags = sortedTags.slice(0, Math.min(10, sortedTags.length)).map(tagInfo => tagInfo.tag);

    console.log('Calling Google Books API for tags:', topTags);

    const booksPromises = topTags.map(tag => {
      const url = `https://www.googleapis.com/books/v1/volumes?q=${tag}&maxResults=1`;
      return new Promise((resolve, reject) => {
        request(url, (error, response, body) => {
          if (error) {
            reject(error);
          } else if (response.statusCode !== 200) {
            reject(new Error(`Google Books API returned status code ${response.statusCode}`));
          } else {
            const data = JSON.parse(body);
            if (data && data.items && data.items.length > 0) {
              resolve({
                title: data.items[0].volumeInfo.title,
                authors: data.items[0].volumeInfo.authors,
                description: data.items[0].volumeInfo.description
              });
            } else {
              resolve(null);
            }
          }
        });
      });
    });
  })
```


Chiamata API



GET ▼ | https://u9kzhlg3ee.execute-api.us-east-1.amazonaws.com/default/getBooksByIdutente_cronologia Send ▼

Params Authorization Headers (8) **Body** ● Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL Text ▼

```
1 {
2   ... "id": "664473f96edd176315934c90"
3 }
```

Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 3.89 s Size: 13.79 KB 💾 Save as example ⋮

Pretty Raw Preview Visualize JSON ▼ ⇌ 📄 🔍

```
1 {
2   "books": [
3     {
4       "title": "The Science Book",
5       "authors": [
6         "DK"
7       ],
8       "description": "Now in Paperback! Take science to a whole new level. Created in partnership with Prentice Hall
9     },
10    {
11      "title": "Technology and Society",
12      "authors": [
13        "Deborah G. Johnson",
14        "Jameson M. Wetmore"
15      ],
16      "description": "An anthology of writings by thinkers ranging from Freeman Dyson to Bruno Latour that focuses
17    },
18    {
19      "title": "The First 20 Hours",
20      "authors": [
21        "Josh Kaufman"
22      ],
23      "description": "Forget the 10,000 hour rule— what if it's possible to learn the basics of any new skill in 20
```

Perchè Google Books API?

1. Vasta raccolta di libri
2. Informazioni dettagliate sui libri
3. Integrazione semplice e gratuita
4. **Risorse aggiuntive:** Possibilità di acquistare direttamente i libri o averne ulteriori consigliati.



Criticità e possibili evoluzioni

Tempo di caricamento della
funzione più lento del previsto



Utilizzare servizi come
ElastiCache che consentono di
memorizzare i dati utilizzati più
frequentemente (es. Utente
nell'app o i libri più consigliati)

TED Books4You

