

01

XPodify

Ascolta la curiosità!

Lorenzo Colombo - 1081134

Lambda FindPodcast

La lambda function ha il compito di selezionare vari podcast di Spotify correlati ai tags del **Talk** che l'utente sta guardando



```
module.exports.FindPodcast = (event, context, callback) => {
  context.callbackWaitsForEmptyEventLoop = false;
  console.log('Received event:', JSON.stringify(event, null, 2));

  let body = {};
  if (event.body) {
    body = JSON.parse(event.body);
  }

  // Set default values
  if (!body.tags || !Array.isArray(body.tags) || body.tags.length === 0) {
    callback(null, {
      statusCode: 400,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Tags are required and must be provided as a non-empty array.' + JSON.stringify(body.body)
    });
    return;
  }

  console.log(JSON.stringify(body))

  if (!body.doc_per_page) {
    body.doc_per_page = 10;
  }
  if (!body.page) {
    body.page = 1;
  }

  // Connect to MongoDB
  connect_to_db().then(() => {
    console.log('=> Searching for related talks');

    // Find talks related to the provided tags
    talk.find({ tags: { $in: body.tags } }, { _id: 0, id_related_list: 1, title_related_list: 1 })
      .then(talks => {
        if (!talks || talks.length === 0) {
          throw new Error('No related talks found');
        }

        // Extract unique tags from related talks
        const relatedTags = [...new Set(talks.flatMap(talk => talk.tags))];
```

Spotify API - 1

Prima API sfruttata per richiedere il token d'accesso e sfruttare le potenzialità di ciò che l'applicazione può offrire

```
// Funzione per ottenere un token di accesso da Spotify
function getSpotifyAccessToken() {
  return new Promise((resolve, reject) => {
    const tokenUrl = 'https://accounts.spotify.com/api/token';
    const authHeader = 'Basic ' + Buffer.from(SPOTIFY_CLIENT_ID + ':' + SPOTIFY_CLIENT_SECRET).toString('base64');

    const options = {
      url: tokenUrl,
      method: 'POST',
      headers: {
        'Content-Type': 'application/x-www-form-urlencoded',
        'Authorization': authHeader,
      },
      form: {
        grant_type: 'client_credentials',
      },
    };

    request(options, (error, response, body) => {
      if (error) {
        console.error('Errore ottenendo token di accesso:', error);
        reject(error);
      } else if (response.statusCode !== 200) {
        console.error('Errore risposta non valida:', body);
        reject(new Error('Errore ottenendo token di accesso'));
      } else {
        const data = JSON.parse(body);
        console.log('Access token ottenuto:', data.access_token);
        resolve(data.access_token);
      }
    });
  });
}
```

```
// Funzione per cercare i podcast usando i tag
function searchPodcastsByTags(accessToken, tags) {
  return new Promise((resolve, reject) => {
    const searchUrl = 'https://api.spotify.com/v1/search';
    const query = tags.join(' '); // Combina i tag per la ricerca

    const options = {
      url: searchUrl,
      method: 'GET',
      headers: {
        'Authorization': 'Bearer ' + accessToken,
      },
      qs: {
        q: query,
        type: 'show', // 'show' è usato per cercare podcast
        market: 'IT',
        limit: 10 // Limita i risultati a 10
      },
    };

    request(options, (error, response, body) => {
      if (error) {
        console.error('Errore cercando podcast:', error);
        reject(error);
      } else if (response.statusCode !== 200) {
        console.error('Errore risposta non valida:', body);
        reject(new Error('Errore cercando podcast'));
      } else {
        const data = JSON.parse(body);
        console.log('Risultati della ricerca di podcast:', data);
        if (data && data.shows && data.shows.items) {
          resolve(data.shows.items); // Restituisce i podcast trovati
        } else {
          resolve([]); // Nessun podcast trovato
        }
      }
    });
  });
}
```

Spotify API - 2

Seconda API utilizzata per ricavare effettivamente i dati dei Podcast correlati ai tag del video



Chiamata API -



GET



https://dsuvyfxzkk.execute-api.us-east-1.amazonaws.com/default/FindPodcast

Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON**

Beautify

```
1 [{"tags":["soccer", "money"]}]
```

Body Cookies Headers (7) Test Results



Status: 200 OK Time: 5.87 s Size: 5.12 KB

Save Response

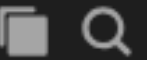
Pretty

Raw

Preview

Visualize

JSON



```
1 [
2   {
3     "id": "6wVReMOTg1aL9SbbVOZxff",
4     "name": "The Plus Money Podcast",
5     "publisher": "The Plus Money Podcast",
6     "description": "Welcome to the Plus Money Podcast, the best soccer betting podcast in 15 minutes or less. Twice a week we are
                        helping you bet on the Premier League, the Champions League, and everything in between. Every episode the guys give their four
                        best bets of the match day and a parlay. Hosted by Andrew Passaro and Ani Sridhar Follow us on social media @theplusmoneypod
                        @adrewpassaro @anisri23"
7   },
8   {
9     "id": "00MWVEYGs5GFF2bMoUGH1h",
10    "name": "Motherhood Talk Radio",
11    "publisher": "Sandra Beck",
```

Perchè Spotify API?



- 1 Applicazione leader nel mercato
- 2 Integrazione gratuita
- 3 Informazioni dettagliate
- 4 Enorme numero di podcast





Punti critici e Miglioramenti

1

**Caricamento
dell'applicazione
rallentato**

2

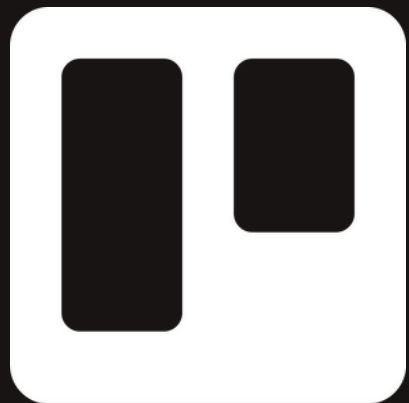
**Specificità delle
informazioni**

3

**Utilizzo di servizi
per gestire la
Cache**



Riferimenti



Trello



Github

