

TECNOLOGIE CLOUD E MOBILE

SWEETEDx Dreams

ALESSIA LAZZARI 1078863



TRELLO, GITHUB,

GET_WATCH_NEXT_BY_IDX - FUNCTION 1

```
const talk = require('./Talk');

module.exports.get_by_idx = (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
  if(!body.idx) {
    callback(null, {
      statusCode: 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the talks. Tag is null.'
    })
  }

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

  connect_to_db().then(() => {
    console.log('=> get_all talks');
    talk.find({ _id: body.idx }, { WatchNext_id: 1, WatchNext_title: 1 })
      .then(talks => {
        if (!talks) {
          throw new Error('Talk not found');
        }
        callback(null, {
          statusCode: 200,
          body: JSON.stringify(talks)
        })
      })
  })
}
```

La lambda function implementata permette, al termine della riproduzione di un talk scelto dall'utente, di consigliare altri video correlati a quello appena visto.

Ambiente di test per la lambda function 1 , attraverso API:

```
2  {
3    "_id": "523970",
4    "WatchNext_id": [
5      "3686",
6      "93273",
7      "1810",
8      "17923",
9      "117739",
10     "53523"
11   ],
12   "WatchNext_title": [
13     "The brain benefits of deep sleep -- and how to get more of it",
14     "The brain science (and benefits) of ASMR",
15     "Why do we sleep?",
16     "What causes insomnia?",
17     "Can you change your sleep schedule?",
18     "Is marijuana bad for your brain?"
19   ]
}
```

FUNZIONALITÀ DEL SERVIZIO

Per lo svolgimento della seconda richiesta, è necessario ricordare le funzionalità dichiarate in precedenza:

1. **Raccomandazioni Personalizzate di Contenuti**
2. **Modalità Notturna**
3. **Timer di Spegnimento Automatico**
4. **Raccolta di Feedback e Valutazioni (implica login utente)**
5. **Creazione di Playlist Personalizzate**
6. **Integrazione con Dispositivi e App di Monitoraggio del Sonno**
7. **Consigli e Risorse per il Benessere Notturno**

Funzionalità scelta

04

Login utente

Funzionalità scelta

03

Timer di
Spegnimento
Automatico

LOGINSO STEP BY STEP

UTILIZZO DI LOGUTENTE.CSV

Ho creato un file .csv dove ho inserito i dati necessari per l'autenticazione di un utente: mail, username e password.

UTILIZZO DI BUCKET S3

Ho caricato logutente.csv nel bucket S3 di riferimento: alessialazzari-data.

JOB LOGIN

Ho creato un job che legge il file .csv caricato e crea una nuova collection nel db: tedx-data-login.

LAMBDA FUNCTION 2

Ho creato la lambda function LOGINSO che si occupa della gestione dei dati di autenticazione e ne ho verificato il funzionamento attraverso API.

LOGINSERVICE - FUNCTION 2

```
const talk = require('./Talk');

module.exports.loginutente = (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
  if(!body.mail) {
    callback(null, {
      statusCode: 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Mail is null.'
    })
  }
  if(!body.password) {
    callback(null, {
      statusCode: 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Password is null.'
    })
  }
}
```

Questa lambda function gestisce la fase di login dell'utente per l'utilizzo del servizio.

Il codice verifica la presenza e la validità dell'indirizzo email e della password forniti.

***Evoluzione futura:* possibilità di commentare, consigliare e valutare i talk visionati.**

TIMER AUTOSPEGNIMENTO - FUNCTION 3

```
// Definizione delle opzioni di timeout predefinite in minuti
const timeoutOptions = {
  '10_minutes': 600000, // 10 minuti
  '20_minutes': 1200000, // 20 minuti
  '30_minutes': 1800000, // 30 minuti
  '40_minutes': 2400000 // 40 minuti
};

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

  // Verifica se l'utente ha specificato una delle opzioni di timeout predefinite
  const timeoutKey = event.timeoutOption;
  let timeoutDuration = timeoutOptions[timeoutKey];

  // Verifica se la chiave del timeout è valida, altrimenti utilizza un valore predefinito
  if (!timeoutDuration) {
    console.log('Invalid timeout option, using default.');
```

timeoutDuration = timeoutOptions['30_minutes']; // Imposta il timeout predefinito a 10 minuti

```
  }

  connectToDb().then(() => {
    console.log('=> shutdown inactive talks');

    const currentTime = new Date();

    VideoTalk.find({ isActive: true })
      .then(videoTalks => {
        const videosToTurnOff = videoTalks.filter(video => {
          return (currentTime - video.startTime) >= timeoutDuration;
        });

        return Promise.all(videosToTurnOff.map(video => {
          return VideoTalk.findByIdAndUpdate(video._id, { $set: { isActive: false } });
        }));
      });
  });
};
```

La terza lambda function implementata si occupa del timer di spegnimento automatico: l'utente seleziona un timer entro il quale prevede di addormentarsi. Al termine di questo timer, la riproduzione di talk viene interrotta. Se l'utente non sceglie un intervallo di tempo, la riproduzione dei talk termina automaticamente dopo 30 minuti.

Ambiente di test per la lambda function 2 , attraverso API:

rawJSON

```
1  {
2    ...."mail": "sophia.stars@example.com",
3    ...."password": "sleepytime2024"
4  }
```

Body

200 OK2.93 s417 B

PrettyRawPreviewVisualizeJSON

```
1  {
2    "Mail": "sophia.stars@example.com",
3    "Username": "SophiaStars",
4    "Password": "sleepytime2024"
5  }
```

Ambiente di test per la lambda function 3 con AWS:

Test Event Name

TestTimer

Response

```
{
  "statusCode": 200,
  "body": "Inactive talks successfully shutdown."
}
```

CRITICITÀ ED EVOLUZIONI

SICUREZZA DATI

La gestione dei dati sensibili potrebbe non essere sicura.
È consigliabile valutare l'uso di metodi di hash o crittografia per proteggere le password.

UTILIZZO AMAZON COGNITO

Utilizzo di Amazon Cognito per gestire la registrazione e l'autenticazione degli utenti.

FUNZIONAMENTO TIMER DI SPEGNIMENTO

Il timer di spegnimento è una funzionalità utile per il servizio, ma non garantisce che entro il termine del timer, l'utente si sia addormentato

TEST E VALUTAZIONI

Solo con l'effettivo utilizzo del servizio e del timer di spegnimento automatico, è possibile valutarne l'efficacia e apportare miglioramenti.