



TEDxSPEAKERS

Improve your public speaking skills



```
module.exports.get_watch_next = (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)
  }
  if(!body.slug) {
    callback(null, {
      statusCode: 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the talks. Slug is null.'
    })
  }
  connect_to_db().then(() => {
    talk.find({"slug": body.slug})
      .then(talks => {
        var related = talks[0].related
        callback(null, {
          statusCode: 200,
          body: JSON.stringify(related)
        })
      })
  })
  .catch(err =>
    callback(null, {
      statusCode: err.statusCode || 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the talks.'
    })
  );
});
};
```

API Watch Next

Il codice della API che dato un
l'id di un video restituisce
l'elenco dei video correlati



[Vedi il codice
completo](#)

Test con PostMan

Il risultato del test eseguito con Postman. Il parametro in ingresso è lo slug del video che si sta visualizzando, e in risposta si ottengono tutti i video correlati da mostrare all'utente. Se non vengono trovati video correlati, viene tornato l'errore 500.

POST

https://5qvwet700a.execute-api.us-east-1.amazonaws.com/default/getWatchNext

```
[
  {
    "slug": "stephanie_honchell_smith_whatever_happened_to_the_hole_in_the_ozone_layer",
    "title": "Whatever happened to the hole in the ozone layer?",
    "duration": "293",
    "viewedCount": "552783",
    "presenterDisplayName": "Stephanie Honchell Smith"
  },
  {
    "slug": "amy_hrdina_and_jesse_kroll_what_s_in_the_air_you_breathe",
    "title": "What's in the air you breathe?",
    "duration": "306",
    "viewedCount": "387040",
    "presenterDisplayName": "Amy Hrdina and Jesse Kroll"
  }
]
```

Copia il link dell'API



Model Lesson

```
const mongoose = require('mongoose');

const lesson_schema = new mongoose.Schema({
  _id: Number,
  title: String,
  link: String,
  content: String
}, { collection: 'speaking_skills' });

module.exports = mongoose.model('lesson', lesson_schema);
```

Aggiungiamo il modello Lesson a cui mappiamo i campi title, link e content presenti nella collection di MongoDB

```
_id: 8
title: "Supporting Your Speech Ideas"
link: "https://pressbooks.bccampus.ca/speaking/chapter/chapter-8/"
content: "8.1 – Why Supporting Materials are NeededPreparing to give a
presentat..."
```

Vai al codice



```

const lesson = require('./Lesson');

module.exports.get_all_lessons = (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)
  }

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

  const sort = { _id: 1 }
  const fields = { content: 0 }

  connect_to_db().then(() => {
    lesson.find({}, fields)
      .sort(sort)
      .skip((body.doc_per_page * body.page) - body.doc_per_page)
      .limit(body.doc_per_page)
      .then(lessons => {
        callback(null, {
          statusCode: 200,
          body: JSON.stringify(lessons)
        })
      })
  })
  .catch(err => {
    callback(null, {
      statusCode: err.statusCode || 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch lessons.'
    })
  })
});
};

```

API getAllLessons

Questa API serve per ottenere tutte le lezioni disponibili caricate.

Utilizziamo la funzione sort per ordinare i risultati in base al numero del capitolo ed escludiamo il contenuto della lezione da i campi che vengono inviati in risposta perché non necessari.

Si potrebbe aggiungere un campo con un estratto della lezione.

Vai al codice



Copia il link dell'API



Test con PostMan

Come risultato ho un json contenente tutte le lezioni di tedxspeakers. Quest'API verrà utilizzata per far scegliere all'utente una lezione tra quelle disponibili.

```
{
  "_id": 1,
  "title": "The Basics of Public Speaking",
  "link": "https://pressbooks.bccampus.ca/speaking/chapter/chapter-1/"
},
{
  "_id": 2,
  "title": "Listening Skills",
  "link": "https://pressbooks.bccampus.ca/speaking/chapter/chapter-2/"
},
}
```



API getLessonContent

Questa API server per ottenere il contenuto di una singola lezione, che verrà poi mostrata all'utente.

Anche in questo caso andrebbero creati dei campi nel database, come ad esempio i concetti chiave, in modo da renderla più accattivante.

```
const connect_to_db = require('./db');

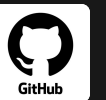
// GET WATCH NEXT

const lesson = require('./Lesson');

✓ module.exports.get_lesson_content = (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.capter) {
    callback(null, {
      statusCode: 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the lesson. Capther is null.'
    })
  }
  var capther = body.capter

  connect_to_db().then(() => {
    lesson.find({"_id":capther},{link : 0})
      .then(lessons => {
        callback(null, {
          statusCode: 200,
          body: JSON.stringify(lessons[0])
        })
      })
    )
    .catch(err =>
      callback(null, {
        statusCode: err.statusCode || 500,
        headers: { 'Content-Type': 'text/plain' },
        body: 'Could not fetch the selected lesson.'
      })
    );
  });
};
```

Vai al codice



Copia il link dell'API




Test con PostMan

Il risultato del test eseguito con Postman. Si nota che sarà necessario dividere il contenuto in capitoli in fase di import, per facilitare la visualizzazione.

```
{
  "_id": 8,
  "title": "Supporting Your Speech Ideas",
  "content": "8.1 - Why Supporting Materials are NeededPreparing to give a presentation
    is not a totally linear process. Even as you practice, you will make small changes
    to your basic outline, since the way something looks on paper and the way it
    sounds are usually different. For example, long sentences may look intelligent on
    paper, but they are hard to say in one breath and hard for the audience to
    understand.Using your supporting materials effectively is essential because
    audiences crave detail and specifics. Supporting material make your ideas,
    arguments, assertions, points, or concepts real and concrete. Sometimes supporting
    materials are thought of as supports for a bridge. Without these supports, while
    you might be able to see the beginning and the end, the whole structure would
```



Prossimi sviluppi



- All'interno di ogni lezione va creato un campo con i concetti chiavi della lezione e un riassunto della lezione
 - In fase di importazione il contenuto della lezione verrà diviso in capitoli, così da poterlo paginare su mobile.
 - Aggiunta del collegamento tra i capitoli e i video di TedX
- 