

Esplora, Impara, con
TEDx e le lingue!

LinguaLearnX

Tomasoni Francesco - 1080980

Homework 3

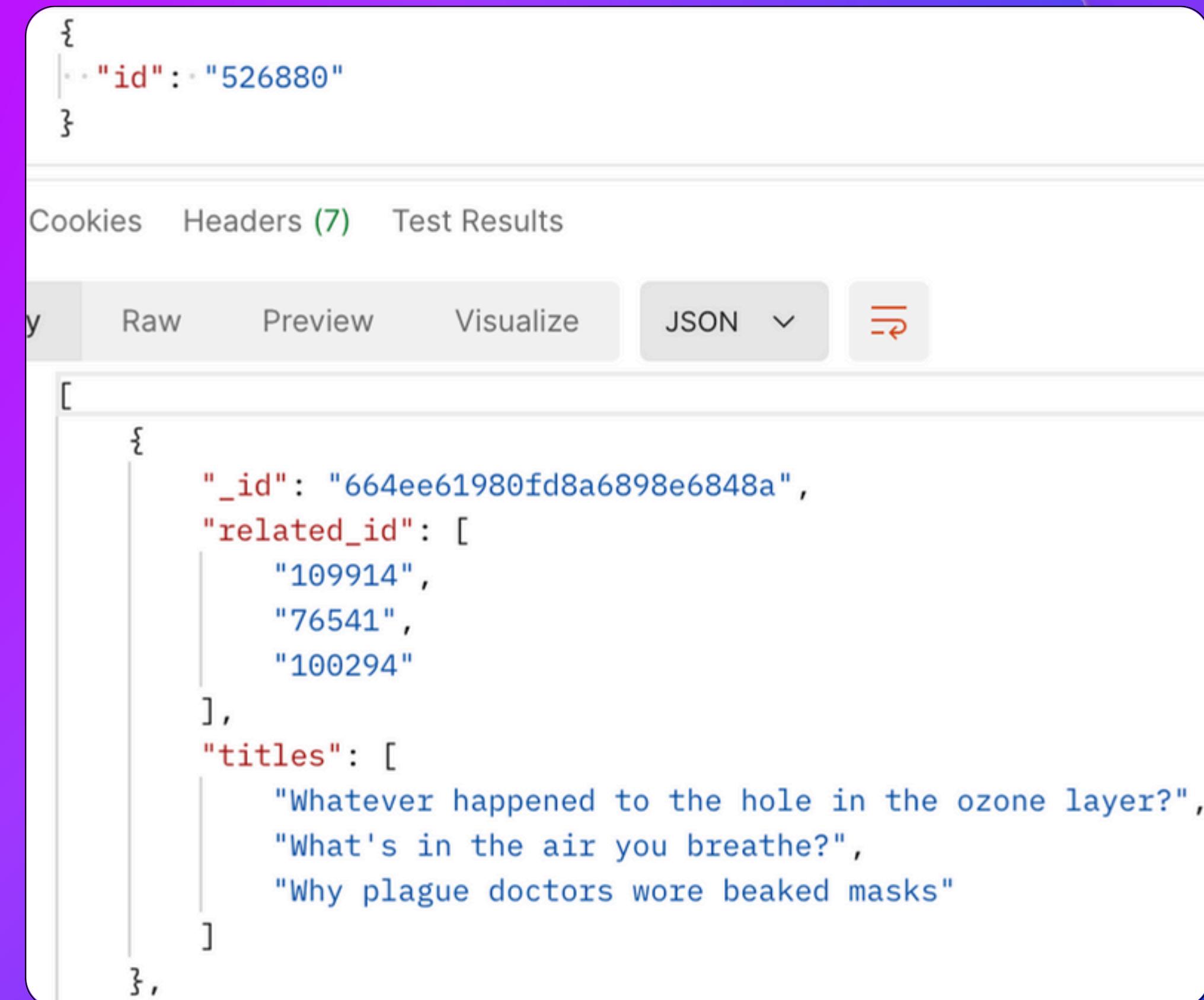
LF - Watch Next

```
connect_to_db().then(() => {
  console.log('=> get_all talks');
  talk.find({ id_ref: body.id }, { related_id : 1})
    .then(talks => {
      if (!talks) {
        throw new Error('nessun talk trovato');
      }
      callback(null, {
        statusCode: 200,
        body: JSON.stringify(talks)
      })
    })
    .catch(err =>
      callback(null, {
        statusCode: err.statusCode || 500,
        headers: { 'Content-Type': 'text/plain' },
        body: 'impossibile caricare i talk.'
      })
    );
});
```

Viene sposta come API, dato un id come
parametro di input , viene restituito l'id, il titolo
e l'url dei watch next corrispondenti

Chiama API

Risposta



A screenshot of a JSON response in a browser developer tools interface. The response is displayed in a JSON viewer with tabs for 'Raw' and 'JSON'. The JSON structure is as follows:

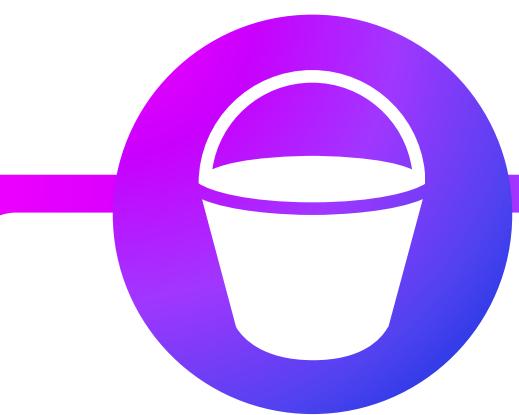
```
{  
  "id": "526880"  
}  
  
[  
  {  
    "_id": "664ee61980fd8a6898e6848a",  
    "related_id": [  
      "109914",  
      "76541",  
      "100294"  
    ],  
    "titles": [  
      "Whatever happened to the hole in the ozone layer?",  
      "What's in the air you breathe?",  
      "Why plague doctors wore beaked masks"  
    ]  
  },  
  ...  
]
```

Login API - Step



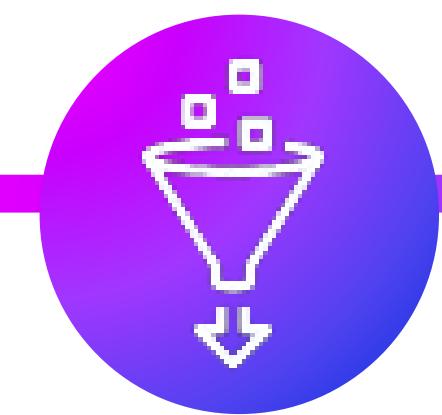
FILE CSV

Creazione del file “User.csv”, contenente le informazioni di accesso username e password, e ulteriori dati come nome, cognome e email



S3

Caricamento del file csv su un bucket S3 : “user-database-tedx”



JOB LOGIN

Job PySpark che legge i dati all'interno del csv e li inserisce nella collection “tedx_user”



API LOGIN

Inseriti username e password come richiesta, verranno cercate le informazioni in modo da garantire l'accesso all'applicazione

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

    await connectToDatabase();

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

    try {
        if (!body.username || !body.password) {
            return callback(null, {
                statusCode: 400,
                body: JSON.stringify({ message: 'Username and password are required' })
            });
        }

        const user = await User.findOne({ username: body.username });

        if (!user) {
            return callback(null, {
                statusCode: 404,
                body: JSON.stringify({ message: 'User not found' })
            });
        }

        const isPasswordValid = await user.comparePassword(body.password);

        if (!isPasswordValid) {
            return callback(null, {
                statusCode: 401,
                body: JSON.stringify({ message: 'Incorrect password' })
            });
        }

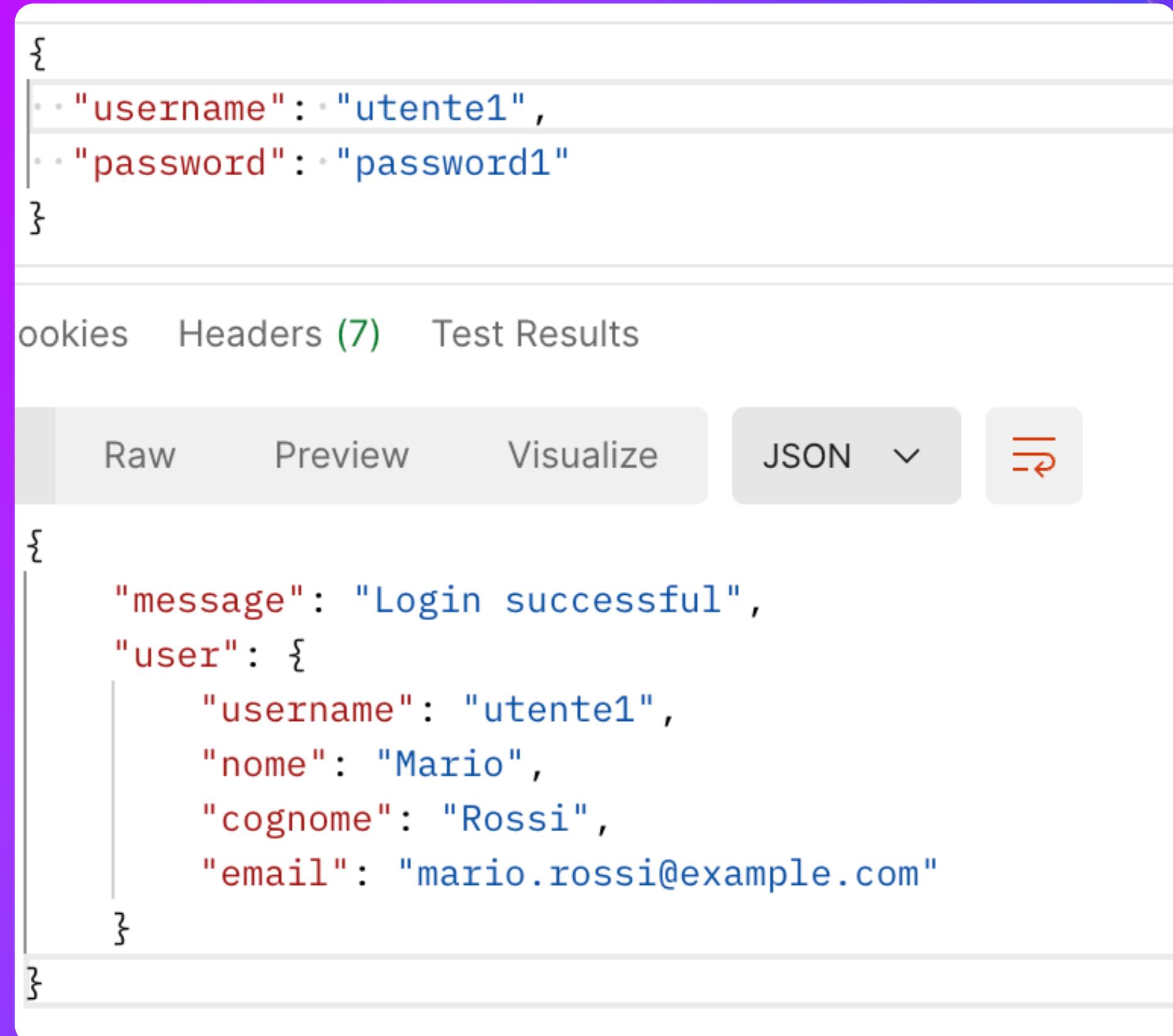
        // Se il login è riuscito, restituisci i dati dell'utente
        return callback(null, {
            statusCode: 200,
            body: JSON.stringify({
                message: 'Login successful',
                user: {
                    username: user.username,
                    nome: user.nome,
                    cognome: user.cognome,
                    email: user.email
                    // Aggiungi altre proprietà se necessario
                }
            })
        });
    } catch (error) {
        console.error('Error:', error);
        return callback(null, {
            statusCode: 500,
            body: JSON.stringify({ message: 'Internal server error' })
        });
    }
};
```

LF - Login

Cerca un documento nella collezione degli utenti corrispondete all'username e alla password fornite nel corpo della richiesta, se trova corripondeza restituisce i dettagli dell'utente

Chiamata API

Risposta



The screenshot shows a JSON response in a browser developer tools interface. The request body (top half) contains:

```
{  
  "username": "utente1",  
  "password": "password1"  
}
```

The response body (bottom half) is displayed in JSON format:

```
{  
  "message": "Login successful",  
  "user": {  
    "username": "utente1",  
    "nome": "Mario",  
    "cognome": "Rossi",  
    "email": "mario.rossi@example.com"  
  }  
}
```

Below the JSON fields, there are tabs for Cookies, Headers (7), and Test Results. A dropdown menu shows "JSON" selected, and a copy icon is visible.

LF - Generate_Quiz

Eposta tramite API gateway.

Riceve in input l'id del talk e controlla che il talk abbia una trascrizione e almeno un tag.

Accede al dataset mongo e recupera il talk tramite id.

Con la libreria “Axios” importata e messa come layer della funzione, viene effettuata una richiesta all’API di [PrepAI](#), passando: il client e il

- token necessario,
- il contenuto, ovvero la trascrizione
- il topic, ovvero i tag
- tipo di domande (categoria 1 sono le crocette), ma ce ne sono altre
- numero di domande massime da generare

Il quiz viene generato e salvato nella collection “tedx_quiz”.

```
const transcript = talk.transcript;
const tags = talk.tags;
const id_ref = talk.id_ref;

// Prepare payload for the API request
const payload = {
  topic: tags,
  content: transcript,
  quesType: '1',
  quesCount: '20'
};

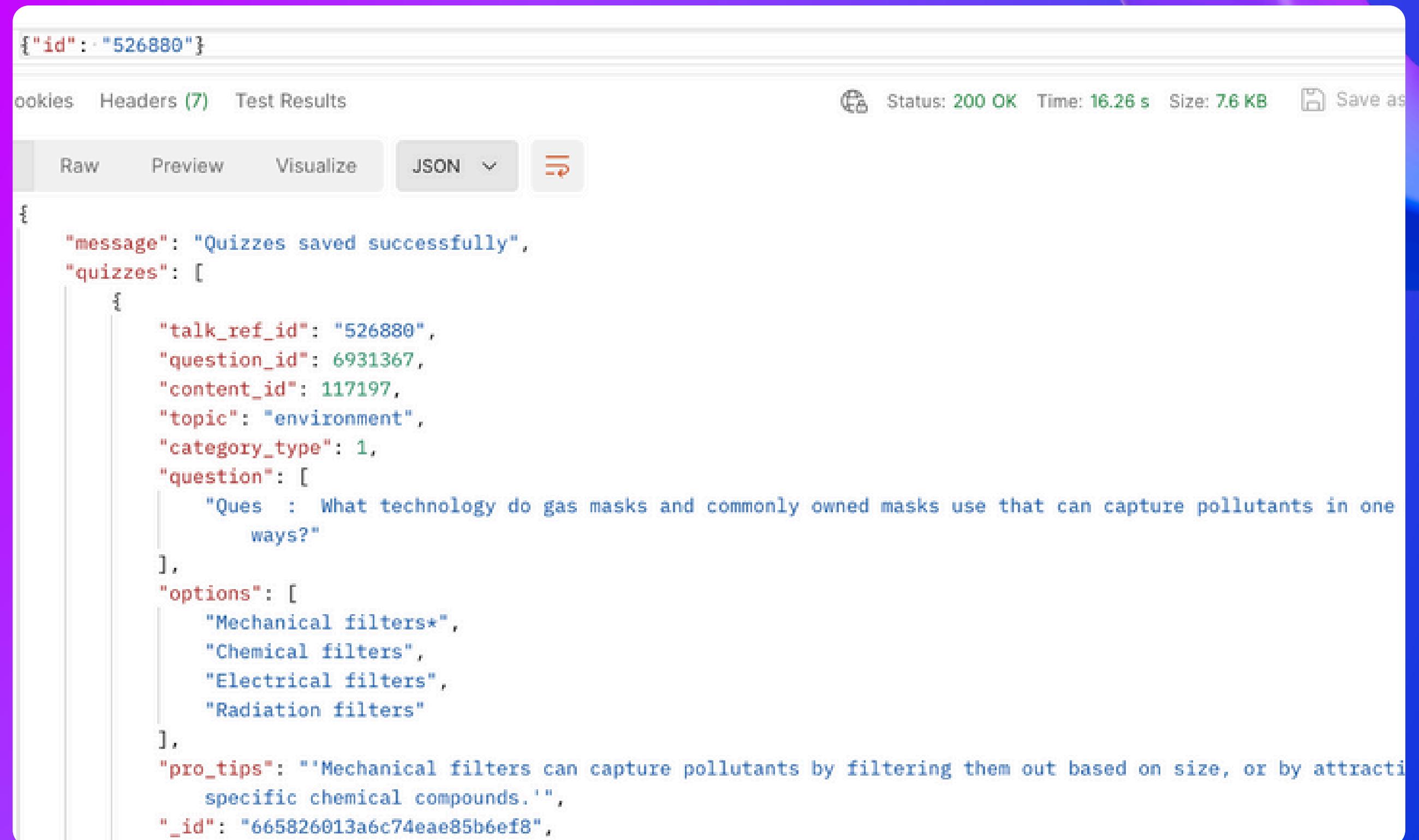
// Prepare headers for the API request
const headers = {
  clientId: '741195bb0324c1327e31a05c71a252d522c3038',
  clientSecret: 'ZjKidlQqHWQuGiqYhxBm4rCI0ybkuhLS_jg8Hwxd'
};

// Make the API request
const querystring = require('querystring');
const formData = querystring.stringify(payload);

axios.post('https://api.prepai.io/generateQuestionsApi', formData, { headers })
  .then(response => {
    const questions = response.data.response;
```

Chiamata API

Risposta



The screenshot shows a REST API response in JSON format. The status is 200 OK, and the time taken is 16.26 s. The response body contains a message and a quiz object.

```
{"id": "526880"}

cookies Headers (7) Test Results



Status: 200 OK Time: 16.26 s Size: 7.6 KB Save as



Raw Preview Visualize JSON



```
{ "message": "Quizzes saved successfully", "quizzes": [{ "talk_ref_id": "526880", "question_id": 6931367, "content_id": 117197, "topic": "environment", "category_type": 1, "question": [{ "Ques": "What technology do gas masks and commonly owned masks use that can capture pollutants in one way?" }, { "options": ["Mechanical filters*", "Chemical filters", "Electrical filters", "Radiation filters"] }, { "pro_tips": "Mechanical filters can capture pollutants by filtering them out based on size, or by attracting specific chemical compounds.", "_id": "665826013a6c74eae85b6ef8" }] }] }
```


```

Generate_Quiz - Salvataggio su Mongo

```
_id: ObjectId('6658234af0fd526b4342bef6')
talk_ref_id : "526880"
question_id : 6931306
content_id : 117195
topic : "environment"
category_type : 1
▼ question : Array (1)
  0: "Ques : What is the minimum size of the majority of
▼ options : Array (4)
  0: "1 micron"
  1: "2.5 microns"
  2: "5 microns"
  3: "10 microns"
pro_tips : "'The majority of wildfire smoke particles are sm
__v : 0
```

I quiz vengono salvati domanda per domanda (identificata tramite id) con l'id del talk a cui appartengono.

Criticità

Tempistiche

La generazione dei quiz potrebbe richiedere abbastanza tempo per testi lunghi

API di PrepAI

L'API di PrepAI offre una prova gratuita di 15 giorni, raggiungibile creando nuovi account, ma non all'infinito.

Questa API genera quiz con domande e risposte, ma al posto di fornire la risposta corretta viene restituito il campo "pro tips" che sarebbe il pezzo di testo da cui è stata estrapolata la domanda , richiedendo quindi una verifica umana.

Sicurezza

La gestione delle password potrebbe non essere del tutto sicura, è consigliabile l'utilizzo di metodi sicuri per la gestione delle password, come l'hashing e il salt

Possibili evoluzioni

Tempistiche

Si potrebbe pensare di separare la generazione dei quiz con il salvataggio su mongo, in modo da poter gestire le cose in modo asincrono.

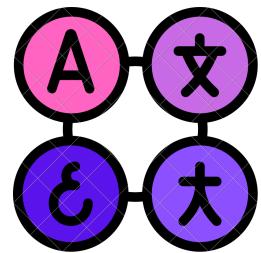
API di PrepAI

Per quanto riguarda la verifica, potrebbe essere automatizzata, tramite l'utilizzo di AI come ChatGPT, che fornisce delle API, che renderebbe poi il controllo umano più veloce.

Oppure trovare un'altri API che generi i quiz e fornisca la domanda corretta.

Amazon Cognito

Utilizzo e integrazione completa con Amazon Cognito per la gestione delle password e l'autenticazione



Esplora, Impara, con
TEDx e le lingue!

LinguaLearnX

