



AWS Lambda

Paolo Olivieri

Xhorxho Papallazi

Simone Ronzoni

Lambda: get watch next

Ritorna i watch next di un determinato talk il cui id è passato come parametro.

Di ciascun watch next sono ritornate tutte le informazioni e non solo l'url, vista la necessità di tali informazioni nell'implementazione.

Parametri:

- `id: String` identificatore univoco talk



Lambda: get watch next

Esempio richiesta:

```
1  {  
2    "id": "7e52bbc6379463aa0a6776d117b5fffd"  
3  }
```

Esempio risposta:

```
1  [  
2    {  
3      "link": "https://www.ted.com/talks/matt\_walker\_6\_tips\_for\_better\_sleep",  
4      "watch_next_idx": "3d09fa5a4a64a82554252244c9420355",  
5      "main_speaker": "Matt Walker",  
6      "title": "6 tips for better sleep",  
7      "posted": "Posted Sep 2020"  
8    },
```



Lambda: post talk questions

Permette ad un utente di inserire delle domande relative ad un talk.

Ci sono 3 tipi di domande accettate:

- «open» sono quelle a risposta aperta
- «v/f» vero o falso
- «choice» a scelta chiusa

Ogni tipo di domanda presenta un formato specifico come nell'esempio di richiesta sulla destra.

```
1  {
2    "id": "08c4c58647c0e91169d52970ef8c198e",
3    "questions":
4    [
5      {
6        "question": "Come si chiama la Pimpa?",
7        "type": "open",
8        "answer": "Pimpa"
9      },
10     {
11       "question": "La Pimpa sa ballare?",
12       "type": "v/f",
13       "answer": "False"
14     },
15     {
16       "question": "La Pimpa ha",
17       "type": "choice",
18       "choices": [
19         {
20           "choice": "4 anni"
21         },
22         {
23           "choice": "6 anni"
24         },
25         {
26           "choice": "4 nasi"
27         },
28         {
29           "choice": "5 orecchie"
30         }
31       ],
32       "answer": "1"
33     }
34   ]
35 }
```



Lambda: get questions by talk id

Ritorna le domande di un determinato talk il cui id è passato come parametro. È inoltre possibile passare come parametro il tipo di domande che si vogliono ottenere (aperte, vero/falso, chiuse).

Parametri:

- `id: String` identificatore univoco talk
- `type: "open|v/f|choice"` tipo di domanda (opzionale)



Lambda: get questions by talk id

Esempio richiesta:

```
{
  "id": "308a45a7c8521824aa5b72ee339504f6",
  "type": "choice"
}
```

Esempio risposta:

```
{
  "_id": "308a45a7c8521824aa5b72ee339504f6",
  "questions": [
    {
      "question": "La Pimpa ha",
      "type": "choice",
      "choices": [
        {
          "choice": "4 anni"
        },
        {
          "choice": "6 anni"
        },
        {
          "choice": "4 nasi"
        },
        {
          "choice": "5 orecchie"
        }
      ]
    }
  ],
  "answer": "1"
}
```

Criticità e evoluzioni

- L'attuale struttura del database è la seguente.
- Le domande sono associate ad ogni video anziché essere associate all'utente che le ha create e memorizzate in un'altra collection.
- A questo stadio dello sviluppo le domande vengono aggiunte dagli utenti, ma in futuro si potranno integrare dei tool per la generazione automatica in base ai testi e alle vecchie domande degli utenti.

```
_id: "017ee110210ef7de5c9e6eed4bb3aa57"
main_speaker: "Emma Bridgewater"
title: "How an entrepreneurial potter helped restore beauty to a city"
details: "At 23-years-old, Emma Bridgewater boldly launched her new ceramic comp..."
posted: "Posted Sep 2015"
url: "https://www.ted.com/talks/emma_bridgewater_how_an_entrepreneurial_pott..."
num_views: "0"
duration: "14:56"
> tags: Array
> subtitles: Array
> watch_next: Array
~ questions: Array
  ~ 0: Object
    question: "Come si chiama la Pimpa?"
    type: "open"
    answer: "Pimpa"
  ~ 1: Object
    question: "La Pimpa sa ballare?"
    type: "v/f"
    answer: "False"
  ~ 2: Object
    question: "La Pimpa ha"
    type: "choice"
    ~ choices: Array
      ~ 0: Object
        choice: "4 anni"
      > 1: Object
      > 2: Object
      > 3: Object
    answer: "1"
  > 3: Object
  > 4: Object
  > 5: Object
```

Criticità e evoluzioni

- Prima evoluzione del sistema è la creazione della parte relativa agli utenti (video preferiti, clan, risposte, ...)
- Essendo il sistema in uno stato embrionale non è presente alcuna autenticazione e per ora le api sono aperte. Andrebbe integrato un sistema di autenticazione su più ruoli.
- Futuro sviluppo riguarda un sistema di log e segnalazione per notificare eventuali problemi al sistema.
- Per come sono realizzate ora le api, l'utente può inserire una qualsiasi domanda relativa a un video, senza alcuna garanzia che questa sia corretta e che non contenga determinate parole inopportune.

Riferimenti

- Tutti i codici sono disponibili alla pagina GitHub:
<https://github.com/maggiorante/Mobile-Cloud-Technologies-2021/tree/master/Homework-3>