

Backend Coding Test

Download 3 csv files from <https://tatoeba.org/en/downloads>:

1. Sentences

Filename

`sentences.tar.bz2`

File description

Contains all the sentences in the selected language. Each sentence is associated with a unique id and an `ISO 639-3` language code.

Fields and structure

Sentence id `[tab]` Lang `[tab]` Text

2. Links Filename

`links.tar.bz2`

File description

Contains the links between the sentences. `1 [tab] 77` means that sentence #77 is the translation of sentence #1. The reciprocal link is also present, so the file will also contain a line that says `77 [tab] 1`.

Fields and structure

Sentence id `[tab]` Translation id

3. Sentences with audio Filename

`sentences_with_audio.tar.bz2`

File description

Contains the ids of the sentences, in all languages, for which audio is available. Other fields indicate who recorded the audio, its license and a URL to attribute the author. If the license field is empty, you may not reuse the audio outside the Tatoeba project.

Fields and structure

Sentence id [tab] **Username** [tab] **License** [tab] **Attribution URL**

Requirements

- Using 3 given csv files above to write a program (**Java or Python script**) to generate English-Vietnamese translation csv file :
 1. ID of English sentence (id)
 2. Text of English sentence (text)
 3. URL of English sentence's audio (audio_url)
 - format: <https://audio.tatoeba.org/sentences/<Lang>/<Sentence id>.mp3>
 - ex: <https://audio.tatoeba.org/sentences/eng/1319.mp3>
 4. ID of Vietnamese translation (translate_id)
 5. Text of Vietnamese translation (translate_text)
- Import translation csv file to PostgreSQL (<https://www.elephantsql.com/>) by another program (**Java or Python script**)
- Make a public **API service (Java Spring Boot)** to show translation data with paging of 10 records using database from above step:
 - Project should be well structured (with layers), not only to fulfill the requirement

How to submit

- Submit your code somewhere accessible from us (Github, Google Drive, ...) with folder structure as below

```
├── api-service
└── scripts
    ├── csv_parse_and_merge.py
    ├── import_db.py
    └── requirements.txt
```

- Host API service somewhere accessible from us (Heroku, Firebase, ...)
Expected API: <YOUR_HOST>/api/translations?page_number=2&page_size=10