AvenueCode

AC - Java Coding test

Story Phrase

As a passionate of Star Wars movies, Anthony Collins (AC) would like to receive your help to process some information from his movies. Friendly as you are, you should help him creating an application to consume and process movie details returning it to him.

Movie Script Test

Given a movie script in the format described below, the application will have to extract the movie settings, the characters and the dialogues, identifying the characters that appeared in each setting, and counting the dialogue words for each character.

Movie settings

Scene headings have one line, all upper-case chars, no indentation, and begin with "EXT. ", "INT. " or "INT./EXT. ". Setting names in scene headings are separated by " - ". Only consider the first setting name and ignore the prefix. Settings with the same name are considered the same setting.

Ex1:

```
INT./EXT. LUKE'S SPEEDER - DESERT WASTELAND - TRAVELING - DAY
```

Results into "LUKE'S SPEEDER".

Ex2:

```
INT. LUKE'S X-WING FIGHTER - COCKPIT - TRAVELING
```

Results into "LUKE'S X-WING FIGHTER".

Movie characters and dialogues

Dialogue headings have one line, all upper-case chars, indented by 22 space chars, with the character name. Character names may contain space chars in the middle. Characters with the same name are considered the same character. Dialogues are indented by 10 space chars. Dialogue descriptions are indented by 15 space chars.
Words for the character dialogues should be counted, case insensitive, ignoring punctuation marks and dialogue descriptions.

Ex1:

```
                      THREEPIO
          I see, sir.

                      LUKE
          Uh, you can call me Luke.

                      THREEPIO
          I see, sir Luke.

                      LUKE
               (laughing)
          Just Luke.
```

Results:
character name: "THREEPIO"
counted words: "i" (2 times), "see" (2 times), "sir" (2 times), "luke" (1 time).
character name: "LUKE"
counted words: "luke" (2 times), "uh" (1 time), "you" (1 time), "can" (1 time), "call" (1 time), "me" (1 time), "just" (1 time).

Ex2:

```
                      THREEPIO
          Artoo-Detoo! It's you! It's you!
```

Results:
character name: "THREEPIO"
counted words: "it’s" (2 times), "you" (2 times), "artoo" (1 time), "detoo" (1 time).

Dialogue word counts should be summed for all the dialogues of each character in all the settings.

Requisites

- Maven must be used to build, run tests and start the application.
- The tests must be started with the `mvn test` command.
- The application must start with a Maven command: `mvn exec:java`, `mvn jetty:run`, `mvn spring-boot:run`, etc.
- The application must have a stateless API and use a database to store data.
- An embedded in-memory database should be used: either H2, HSQL, SQLite or Derby.
- The database and tables creation should be done by Maven or by the application.

REST API

The application will have to expose a RESTful API to receive the movie script and return the collected information.

When an error occurs, the message should not be a generic "Unexpected error", but instead describe the specific error. Errors should also be logged by the application.

POST /script

Accepts

- text/plain; charset=utf-8

Description

Receives the movie script in the text format described above. If a movie script was already received before, it should respond with an error.

Responses

- 200: Success message

  ```
  { "message": "Movie script successfully received" }
  ```

- 403: Forbidden message

  ```
  { "message": "Movie script already received" }
  ```

- default: Unexpected error

```
{ "message": "Unexpected error" }
```

GET /settings

Description

Returns information about all the movie settings. The response includes the name of the settings, the list of characters that appeared in each setting, and the ten top dialogue word counts for each character, considering dialogues in all settings, in descending order by count.

Responses

- 200: An array of movie settings

```
[
  {
    "id": 1,
    "name": "REBEL BLOCKADE RUNNER",
    "characters": [
      {
        "id": 1,
        "name": "THREEPIO",
        "wordCounts": [
          { "word": "you", "count": 50 },
          { "word": "i", "count": 41 },
          { "word": "the", "count": 37 },
          { "word": "sir", "count": 34 },
          { "word": "to", "count": 32 },
          { "word": "a", "count": 28 },
          { "word": "this", "count": 21 },
          { "word": "i'm", "count": 21 },
          { "word": "my", "count": 20 },
          { "word": "don't", "count": 18 }
        ]
      }
    ]
  }
]
```

- default: Unexpected error

```
{ "message": "Unexpected error" }
```

GET /settings/{id}

Description

Returns information about the movie setting with the given id, if one exists. The response includes the name, the list of characters that appeared in the setting, and the ten top dialogue word counts for each character, considering dialogues in all settings, in descending order by count.

Parameters

- id: Movie setting id, required, integer (int64)

Responses

- 200: A movie setting

```
{
  "id": 1,
  "name": "REBEL BLOCKADE RUNNER",
  "characters": [
    {
      "id": 1,
      "name": "THREEPIO",
      "wordCounts": [
        { "word": "you", "count": 50 },
        { "word": "i", "count": 41 },
        { "word": "the", "count": 37 },
        { "word": "sir", "count": 34 },
        { "word": "to", "count": 32 },
        { "word": "a", "count": 28 },
        { "word": "this", "count": 21 },
        { "word": "i'm", "count": 21 },
        { "word": "my", "count": 20 },
        { "word": "don't", "count": 18 }
      ]
    }
  ]
}
```

- 404: Not found

```
{ "message": "Movie setting with id 100 not found" }
```

- default: Unexpected error

```
{ "message": "Unexpected error" }
```

GET /characters

Description

Returns information about all the movie characters. The response includes the name of the characters and the ten top dialogue word counts for each character, in descending order by count.

Responses

- 200: An array of movie characters

```
[
  {
    "id": 1,
    "name": "THREEPIO",
    "wordCounts": [
      { "word": "you", "count": 50 },
      { "word": "i", "count": 41 },
      { "word": "the", "count": 37 },
      { "word": "sir", "count": 34 },
      { "word": "to", "count": 32 },
      { "word": "a", "count": 28 },
      { "word": "this", "count": 21 },
      { "word": "i'm", "count": 21 },
      { "word": "my", "count": 20 },
      { "word": "don't", "count": 18 }
    ]
  }
]
```

- default: Unexpected error

```
{ "message": "Unexpected error" }
```

GET /characters/{id}

Description

Returns information about the movie character with the given id, if one exists. The response includes the name and the ten top dialogue word counts for the character.

Parameters

- id: Movie character id, required, integer (int64)

Responses

- 200: A movie character

```
{
  "id": 1,
  "name": "THREEPIO",
  "wordCounts": [
    { "word": "you", "count": 50 },
    { "word": "i", "count": 41 },
    { "word": "the", "count": 37 },
    { "word": "sir", "count": 34 },
    { "word": "to", "count": 32 },
    { "word": "a", "count": 28 },
    { "word": "this", "count": 21 },
    { "word": "i'm", "count": 21 },
    { "word": "my", "count": 20 },
    { "word": "don't", "count": 18 }
  ]
}
```

- 404: Not found

```
{ "message": "Movie character with id 100 not found" }
```

- default: Unexpected "error

```
{ "message": "Unexpected error" }
```

Delivery

1. Follow the instructions on the email body.

Obs: Any doubt, send an email to **ac.brazil.recruitment@avenuecode.com**