Kenytt Avery, Computer Science Department, California State University, Fullerton

CPSC 449 - Web Back-End Engineering

Exercise 1 - Fall 2023

This exercise may be completed individually, or in a pair with another student. If you work with another student, only one of you needs to submit via Canvas.

Exercise

By the end of this exercise you will have written Python programs to obtain information from REST and GraphQL APIs for an online music store.

Setup

Complete the following steps to set up REST and GraphQL APIs for an online music store. The instructions given are for Tuffix 2022.

Obtain the Chinook sample database

Download and extract the chinook SQLite sample database from sqlitetutorial.net.

```
wget https://www.sqlitetutorial.net/wp-content/uploads/2018/03/chinook.zip
unzip chinook.zip
```

Install the sqlite3 command line tool, then verify that `chinook.db` has been extracted correctly by dumping the contents of the database.

```
sudo apt update
sudo apt install --yes sqlite3
sqlite3 chinook.db .dump
```

Configure a Python virtual environment

Read Relieving your Python packaging pain to understand the installation instructions given in this section, then install the pip and venv packages.

```
sudo apt install --yes python3-pip python3-venv
```

Create a virtual environment where packages will be installed, activate the environment, then configure the Bash shell to automatically activate the environment.

```
python3.10 -m venv $HOME/.venv
echo 'source $HOME/.venv/bin/activate' | tee -a $HOME/.bashrc
. $HOME/.venv/bin/activate
```

Install REST and GraphQL API servers

~~Install the sandman2 server to “automagically” generate REST APIs for an existing database.~~

```
~~python -m pip install sandman2~~
~~python -m pip install Flask-SQLAlchemy==2.5.1~~
```

Use n-install to install the n version manager for Node.js, install the latest LTS version of Node.js, reload the Bash shell configuration, then update to the latest version of the NPM package manager.

```
curl -s -L http://git.io/n-install | bash -s -- -y
. $HOME/.bashrc
npm update --global
```

Install the soul server to expose a REST API for an existing database.

```
npm install --global soul-cli
```

Install the tuql server to expose a GraphQL API for an existing database.

```
npm install --global tuql
```

Start the API servers

In separate terminal windows, start API servers for the chinook database

```
soul --database chinook.db --studio
tuql --db chinook.db --graphiql
```

Querying data with REST and GraphQL

Access the admin interfaces

Check that you can access the ~~admin interface for sandman2~~ Soul Studio GUI, the Soul API documentation, and the GraphiQL IDE for tuql by browsing to the following URLs:

- ~~http://localhost:5000/admin/~~
- http://localhost:8000/studio/
- http://localhost:8000/api/docs
- http://locahost:4000/graphql

Retrieve a record

Visit the following URL to retrieve information about the first artist in the database using the REST API:

```
~~http://localhost:5000/artists/1~~
http://localhost:8000/api/tables/artists/rows/1
```

Note: in Firefox, you can use the “Raw Data” tab and the “Pretty Print” button to view the output in JSON.

In GraphiQL, execute the following query to retrieve the same information from the GraphQL API:

```
query {
  artist(where: {artistId: 1}) {
    artistId,
    name
  }
}
```

For details on filtering GraphQL query results using the where clause, see the documentation for the Where in the Sequelize ORM documentation.

Write more complicated queries

Using the database diagram for reference, write queries using both APIs to retrieve the following:

1. Albums by the artist “Red Hot Chili Peppers.”
2. Genres associated with the artist “U2.”
3. Names of tracks on the playlist “Grunge” and their associated artists and albums.

Note: while the results for each query can be retrieved in a single GraphQL API call, you will likely need to make several REST API calls for each query. You may wish to save these calls with a desktop GUI tool such as Insomnia or Postman.

Making API calls from Python

Following the instructions in the Quickstart for Requests, install the library and write Python programs to print out the answers to each of the queries listed above, using both the REST and GraphQL APIs. For more details, see Python’s Requests Library (Guide).

Documenting your results

In order to complete the exercise you will need to write a short report in PDF format documenting your results for each step.

At the beginning of your report, include:

- Your name
- Your partner's name (if you worked with someone else)
- The semester, course number, and section number
- The exercise number

For each item above:

- Clearly identify which step is being performed.
- Document performing the step itself, including any commands that were run and any code that was written or modified.
- Document the results of the step. Include diagrams, written descriptions of the results, or screenshots clearly demonstrating its success or failure.
- Analyze and comment on the results. Include references that you consulted, variations that you attempted, issues you encountered, and troubleshooting steps that you needed to perform.

Submit your work as a `.pdf` file through Canvas by the due date. If you work with another student, only one submission is required as long as it includes the names of both students.