

summer-break-app

summer-break-app is a web service app, it tracks revenue and expenses.

Features

The web service has the following API endpoints:

`POST /transactions`

Take as input the CSV formatted data as below, parse, and store the data. For example:

```
2020-07-01, Expense, 18.77, Gas
2020-07-04, Income, 40.00, 347 Woodrow
2020-07-06, Income, 35.00, 219 Pleasant
2020-07-12, Expense, 49.50, Repairs
```

`GET /report`

Return a JSON document with the tally of gross revenue, expenses, and net revenue (gross - expenses) as follows:

```
{
  "gross-revenue": <amount>,
  "expenses": <amount>,
  "net-revenue": <amount>
}
```

Requirements

The code is tested on CentOS Linux. We assume the following are available to run the app: - python3 - curl - virtualenv

Installation

- Download and extract the summer-break-app
- Define the project root directory

```
export ROOT_PRJ_DIR=/path/to/summer-break-app
```

- Create and activate a Python environment for your project. The virtual env must be inside the summer-break-app

```
cd ${ROOT_PRJ_DIR}
python3 -m virtualenv .venv
source .venv/bin/activate
```

- Install app dependencies

```
pip3 install -r build/requirements.txt
```

- You can run unittests as follows

```
python3 build/xrun.py -u
```

- You can start flask server as follows

```
python3 build/xrun.py -s
```

or

```
FLASK_APP=${ROOT_PRJ_DIR}/components/impl/server/app.py flask run
```

- After having started the flask server, you can run simple curl tests for the API endpoints as follows

```
python3 build/xrun.py -c
```

or

```
curl -X POST http://127.0.0.1:5000/transactions -F "data=@build/data.csv"  
curl http://127.0.0.1:5000/report
```

Drawbacks

- The csv headers ('Date, Type, Amount(\$), Memo') has been hardcoded in the app. It is difficult for the end user to adapt it to accommodate more headers.
- There is no script-based technique for the end user to insert business logic to handle desired computations.
- The business logic and the csv parser belong to the same class.

Future Improvements

- Instead of storing the csv file, read it as a stream.
- Allow app customizations such as file format, etc. through property files.
- Increase comment and test coverage.
- Use the Sphinx framework to improve app documentation.
- To assure CI, host the app on gitlab.
- Run the application in a Docker container as a service.