Agenda For the Week:

Build a CRUD flask API

GitHub Repo Link:

https://github.com/SarayuGautam/Flask-Crud-API

Related Readings:

- https://www.youtube.com/watch?v=PTZiDnuC86g

- https://realpython.com/flask-connexion-rest-api/
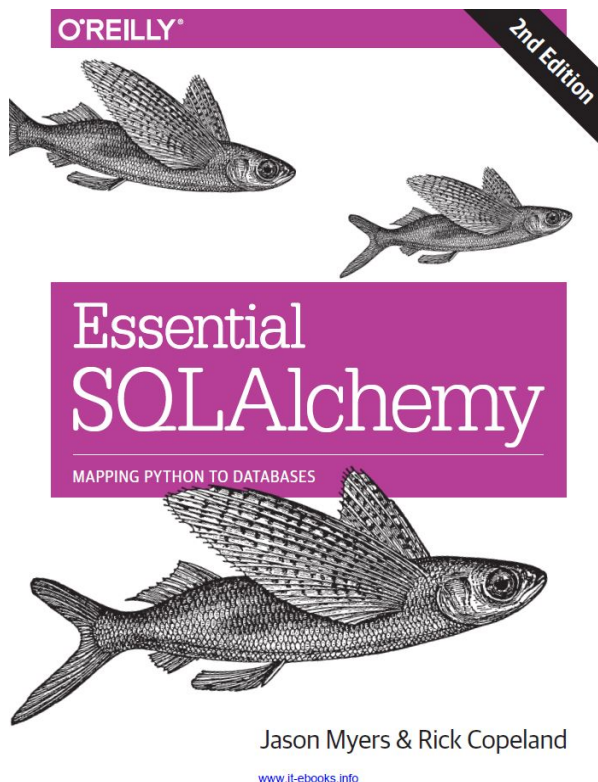
- https://www.amazon.com/Essential-SQLAlchemy-Rick-Copeland/dp/0596516142
  (Half-Finished)

Incorporated Suggestion from last week + Solved what I was having trouble with last week:

- Made a separate file for dealing with the database.

- Used BluePrint to modularise my API

- Resolved the error because of circular import while using a different file for a different purpose

- Used SQLAlchemy along with MySQL as major database

Main Learning:

1. **Object Relational Mapper:** It is an interface that allows the programming language to interact with the database without having to write queries.

2. Just like there is mongoose ORM for MongoDB, SQLAlchemy is only an abstraction layer that works with multiple databases. Here I have made it to work with MySQL.

3. [Marshmallow](#) is a python package that is used for object Serialization and Deserialization

- **[pipenv](#)** => very useful virtual environment package manager. Have used in this week's project
- **pipenv shell** => creates pipenv folder

4. To connect to a database, we need to create an SQLAlchemy engine. The SQLAlchemy engine creates a common interface to the database to execute SQL statements.

Related Screenshots:

FLASK CRUD API / **Get Book**

GET ⌄	http://127.0.0.1:5000/book/1

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings

Query Params

	KEY	VALUE
	Key	Value

Body   Cookies   Headers (4)   Test Results

Pretty   Raw   Preview   Visualize   JSON ⌄   ⇥

```json
1  {
2      "author": "Agatha Cristie",
3      "description": "A gripping thriller",
4      "id": 1,
5      "name": "And then there were none",
6      "price": 400.0
7  }
```

FLASK CRUD API / Get Books

GET ∨ http://127.0.0.1:5000/book |

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON ∨ ⮒

```json
1   [
2       {
3           "author": "Agatha Cristie",
4           "description": "A gripping thriller",
5           "id": 1,
6           "name": "And then there were none",
7           "price": 400.0
8       },
9       {
10          "author": "Jodi Piccoult",
11          "description": "A suspenseful and sad story of a daughter's quest to find her mother",
12          "id": 2,
13          "name": "Leaving Time",
14          "price": 800.0
15      }
16  ]
```

PUT ∨ http://127.0.0.1:5000/book/2

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings

● none   ● form-data   ● x-www-form-urlencoded   ● raw   ● binary   ● GraphQL   JSON ∨

```
1  {
2      "name":"Leaving Time",
3      "description":"A suspenseful and sad story of a daughter's quest to find her mother",
4      "price":"850.00",
5      "author":"Jodi Piccoult"
6  }
```

Body   Cookies   Headers (4)   Test Results

Pretty   Raw   Preview   Visualize   JSON ∨   ⇥

```
1  {
2      "author": "Jodi Piccoult",
3      "description": "A suspenseful and sad story of a daughter's quest to find her mother",
4      "id": 2,
5      "name": "Leaving Time",
6      "price": 850.0
7  }
```

DELETE   http://127.0.0.1:5000/book/1

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings

Query Params

KEY	VALUE
Key	Value

Body   Cookies   Headers (4)   Test Results

Pretty   Raw   Preview   Visualize

{ "author": "Agatha Cristie", "description": "A gripping thriller", "id": 1, "name": "And then there were none", "price": 400.0 }

Fig: Figure showing request and response in Postman