0x04. AirBnB clone - Web framework

Python | Back-end | Webserver | Flask

Concepts

For this project, we expect you to look at this concept:

- AirBnB clone

Resources

Read or watch:

- What is a Web Framework?
- A Minimal Application
- Routing (except "HTTP Methods")
- Rendering Templates
- Synopsis
- Variables
- Comments
- Whitespace Control
- List of Control Structures (read up to "Cycle")
- Filters
- Jinja

Recommended YouTube playlist to get you started

Learning Objectives

At the end of this project, you are expected to be able to **explain to anyone, without the help of Google:**

General

- What is a Web Framework
- How to build a web framework with Flask
- How to define routes in Flask
- How to handle URL
- How to handle variables in a route
- What is a template
- How to create a HTML response in Flask by using a template
- How to create a dynamic template (loops, conditions, ...)
- How to display in HTML data from a MySQL database

Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.

Requirements

Python Scripts

- Allowed editors: vi, vim, emacs
- All your files will be interpreted/compiled on Ubuntu 20.04 LTS using python3 (version 3.4.3)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/python3`
- A `README.md` file at the root of the folder of the project, is mandatory
- Your code should follow PEP 8 style (version 1.7)
- All your files must be executable
- The length of your files will be tested using `wc`
- All your modules should have documentation (`python3 -c "print(__import__('my_module').__doc__)"`)
- All your classes should have documentation (`python3 -c "print(__import__('my_module').MyClass.__doc__)"`)
- All your functions and methods should have documentation (`python3 -c "print(__import__('my_module').MyClass.my_function.__doc__)"`)
- A documentation is not a simple word, it is a real sentence explaining what's the purpose of the module, class or method (the length of it will be verified)

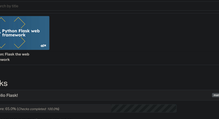
HTML/CSS Files

- Allowed editors: vi, vim, emacs
- All your files should end with a new line
- A `README.md` file at the root of the folder of the project is mandatory
- Your code should be W3C compliant and validate with W3C Validator (except for plug-in/ template)
- All your CSS files should be in the `styles/` folder
- All your HTML files should be in the `templates/` folder
- You are not allowed to use `!important` or `id` in the CSS file
- All tags must be in uppercase
- Current screenshots have been done on Google Chrome 95.0.2845.87
- No cross-browsers

More Info

Install Flask

```
$ pip3 install Flask
```



Manual QA Review

It is your responsibility to request a review for this project from a peer before the project's deadline. If no peers have been reviewed, you should request a review from a TA or staff member.

Video library - 2 read

Search by title

Python Flask web framework

Python Flask web framework

Tasks

0. Hello Flask!

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a script that starts a Flask web application:

- Your web application must be listening on `0.0.0.0`, port `5000`
- Routes:
  - `/`: display "Hello HBNB!"
- You must use the option `strict_slashes=False` in your route definition

```
guillaumebouchard@ubuntu$ vim 0-hello_route
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

In another tab:

```
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000 ; echo "" | cat -e
Hello HBNB!
guillaumebouchard@ubuntu$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- Directory: `web_flask`
- File: `0-hello_route.py`, `__init__.py`

Done! | Check your code | Ask a teaching assistant | QA Review

1. HBNB

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a script that starts a Flask web application:

- Your web application must be listening on `0.0.0.0`, port `5000`
- Routes:
  - `/`: display "Hello HBNB!"
  - `/hbnb`: display "HBNB"
- You must use the option `strict_slashes=False` in your route definition

```
guillaumebouchard@ubuntu$ vim 1-hbnb_route
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

In another tab:

```
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/hbnb ; echo "" | cat -e
HBNB
guillaumebouchard@ubuntu$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- Directory: `web_flask`
- File: `1-hbnb_route.py`

Done! | Check your code | Ask a teaching assistant | QA Review

2. C is fun!

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a script that starts a Flask web application:

- Your web application must be listening on `0.0.0.0`, port `5000`
- Routes:
  - `/`: display "Hello HBNB!"
  - `/hbnb`: display "HBNB"
  - `/c/<text>`: display "C", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
- You must use the option `strict_slashes=False` in your route definition

```
guillaumebouchard@ubuntu$ vim 2-c_route
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

In another tab:

```
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/c/is_fun ; echo "" | cat -e
C is fun
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/c/cisfun ; echo "" | cat -e
C cisfun
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/c
<!DOCTYPE HTML>
<!Latin1 Not Found!!!Line>
<!Char! Found!
<p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
guillaumebouchard@ubuntu$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- Directory: `web_flask`
- File: `2-c_route.py`

Done! | Check your code | Ask a teaching assistant | QA Review

3. Python is cool!

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a script that starts a Flask web application:

- Your web application must be listening on `0.0.0.0`, port `5000`
- Routes:
  - `/`: display "Hello HBNB!"
  - `/hbnb`: display "HBNB"
  - `/c/<text>`: display "C", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
  - `/python/(<text>)`: display "Python", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
    - The default value of `text` is "is cool"
- You must use the option `strict_slashes=False` in your route definition

```
guillaumebouchard@ubuntu$ vim 3-python_route
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

In another tab:

```
guillaumebouchard@ubuntu$ curl -lp 0.0.0.0:5000/python/is_major ; echo "" | cat -e
Python is major
guillaumebouchard@ubuntu$ curl -lp 0.0.0.0:5000/python ; echo "" | cat -e
Python is cool
guillaumebouchard@ubuntu$ curl -lp 0.0.0.0:5000/python/ ; echo "" | cat -e
Python is cool
guillaumebouchard@ubuntu$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- Directory: `web_flask`
- File: `3-python_route.py`

Done! | Check your code | Ask a teaching assistant | QA Review

4. Is it a number?

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a script that starts a Flask web application:

- Your web application must be listening on `0.0.0.0`, port `5000`
- Routes:
  - `/`: display "Hello HBNB!"
  - `/hbnb`: display "HBNB"
  - `/c/<text>`: display "C", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
  - `/python/(<text>)`: display "Python", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
    - The default value of `text` is "is cool"
  - `/number/<n>`: display "`n` is a number" only if `n` is an integer
- You must use the option `strict_slashes=False` in your route definition

```
guillaumebouchard@ubuntu$ vim 4-number_route
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

In another tab:

```
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number/98 ; echo "" | cat -e
98 is a number
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number/0.9
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0 Final//EN">
<!Latin1 Not Found!!!Line>
<!Char! Found!
<p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number/cisfun
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0 Final//EN">
<!Latin1 Not Found!!!Line>
<!Char! Found!
<p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
guillaumebouchard@ubuntu$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- Directory: `web_flask`
- File: `4-number_route.py`

Done! | Check your code | Ask a teaching assistant | QA Review

5. Number template

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a script that starts a Flask web application:

- Your web application must be listening on `0.0.0.0`, port `5000`
- Routes:
  - `/`: display "Hello HBNB!"
  - `/hbnb`: display "HBNB"
  - `/c/<text>`: display "C", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
  - `/python/(<text>)`: display "Python", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
    - The default value of `text` is "is cool"
  - `/number/<n>`: display "`n` is a number" only if `n` is an integer
  - `/number_template/<n>`: display a HTML page only if `n` is an integer:
    - H1 tag: "Number: `n`" inside the tag `body`
  - `/number_odd_or_even/<n>`: display a HTML page only if `n` is an integer:
    - H1 tag: "Number: `n` is `{odd | even}`" inside the tag `body`
- You must use the option `strict_slashes=False` in your route definition

```
guillaumebouchard@ubuntu$ vim 5-number_template
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

In another tab:

```
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number_template/89 ; echo ""
<!DOCTYPE html>
<HTML lang="en">
  <HEAD>
    <!-- UTF-8 Encoding -->
    <TITLE>
    </TITLE>
    <BODY>
      <H1>Number: 89</H1>
    </BODY>
  </HTML>
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number_template/8.7
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0 Final//EN">
<!Latin1 Not Found!!!Line>
<!Char! Found!
<p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number_template/cisfun
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0 Final//EN">
<!Latin1 Not Found!!!Line>
<!Char! Found!
<p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
guillaumebouchard@ubuntu$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- Directory: `web_flask`
- File: `templates/number_template.html`

Done! | Check your code | Ask a teaching assistant | QA Review

6. Odd or even?

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a script that starts a Flask web application:

- Your web application must be listening on `0.0.0.0`, port `5000`
- Routes:
  - `/`: display "Hello HBNB!"
  - `/hbnb`: display "HBNB"
  - `/c/<text>`: display "C", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
  - `/python/(<text>)`: display "Python", followed by the value of the `text` variable (replace underscore `_` symbols with a space `_`)
    - The default value of `text` is "is cool"
  - `/number/<n>`: display "`n` is a number" only if `n` is an integer
  - `/number_template/<n>`: display a HTML page only if `n` is an integer:
    - H1 tag: "Number: `n`" inside the tag `body`
  - `/number_odd_or_even/<n>`: display a HTML page only if `n` is an integer:
    - H1 tag: "Number: `n` is `{odd | even}`" inside the tag `body`
- You must use the option `strict_slashes=False` in your route definition

```
guillaumebouchard@ubuntu$ vim 6-number_odd_or_even
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

In another tab:

```
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number_odd_or_even/98 ; echo ""
<!DOCTYPE html>
<HTML lang="en">
  <HEAD>
    <!-- UTF-8 Encoding -->
    <TITLE>
    </TITLE>
    <BODY>
      <H1>Number: 98 is Even</H1>
    </BODY>
  </HTML>
guillaumebouchard@ubuntu$ curl 0.0.0.0:5000/number_odd_or_even/89 ; echo ""
<!DOCTYPE html>
<HTML lang="en">
  <HEAD>
    <!-- UTF-8 Encoding -->
    <TITLE>
    </TITLE>
    <BODY>
      <H1>Number: 89 is Odd</H1>
    </BODY>
  </HTML>
guillaumebouchard@ubuntu$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- Directory: `web_flask`
- File: `templates/number_odd_or_even.html`

Done! | Check your code | Ask a teaching assistant | QA Review

```
pulicanhobbies:~$ curl 2.0.0.5:8000/number_add_in_store/43 ; echo **
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>404: Not Found</title>
  </head>
  <body>
    <h1>Not Found</h1>
    <ul>
      <li>URL requested: /43 is not</li>
    </ul>
  </body>
</html>
pulicanhobbies:~$ curl 2.0.0.5:8000/number_add_in_store/12 ; echo **
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>404: Not Found</title>
  </head>
  <body>
    <h1>Not Found</h1>
    <ul>
      <li>URL requested: 12 is a negative</li>
    </ul>
  </body>
</html>
pulicanhobbies:~$ curl 2.0.0.5:8000/number_add_in_store/python
{"detail":"Not Found"}pulicanhobbies:~$
pulicanhobbies:~$ find / -name "*number_add_in_store.py"
find: '/run/user/1000': Permission denied
find: '/run/snapd.ns': Permission denied
find: '/run/lvm': Permission denied
/home/pulicanhobbies/.local/share/Trash/files/number_add_in_store.bkup
/home/pulicanhobbies/.local/share/Trash/files/number_add_in_store.bkup
pulicanhobbies:~$
```

Repo:

- GitHub repository: ALEXNE_STAR/JS
- Directory: JS/Node
- File: Project_JS/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup

Clone | Check your code | Tell a random | Ask Review

7. Improve engines

Handmade

Score: 80.01% (Checks uncovered: 100.0%)

Before using Replay to record HDM data, you will need to update some part of our engine:

Update `Project_JS/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup`

- Add a public method `def listen(self):` call `recv()` method on the private session attribute `self._session` to receive the data `recv_data` like:

Update `Project_JS/_js_on_aws/_templates/_js_on_aws.bkup` if it's not already present

- If your engine uses `Node` to run your JS/Node.js server, add a public method `def node(self, *):` return the list of `str` objects from `sys.argv` linked to the current `Node`:

```
pulicanhobbies:~$ ls -ltrh /home/ubuntu/.local/share/Trash/files/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup
-rwxrwxrwx 1 pulicanhobbies pulicanhobbies 1.1K Jan 1 00:00 /home/ubuntu/.local/share/Trash/files/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup
pulicanhobbies:~$ find / -name "*_js_on_aws.bkup"
find: '/run/user/1000': Permission denied
find: '/run/snapd.ns': Permission denied
find: '/run/lvm': Permission denied
/home/pulicanhobbies/.local/share/Trash/files/_js_on_aws.bkup
/home/pulicanhobbies/.local/share/Trash/files/_js_on_aws.bkup
pulicanhobbies:~$ cp /home/pulicanhobbies/.local/share/Trash/files/_js_on_aws.bkup /home/pulicanhobbies/Project_JS/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup
pulicanhobbies:~$ ls -ltrh /home/pulicanhobbies/Project_JS/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup
-rwxrwxrwx 1 pulicanhobbies pulicanhobbies 1.1K Jan 1 00:00 /home/pulicanhobbies/Project_JS/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup
pulicanhobbies:~$
```

At this moment, in another tab:

```
pulicanhobbies:~$ ls -ltrh /home/ubuntu/.local/share/Trash/files/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup
-rwxrwxrwx 1 pulicanhobbies pulicanhobbies 1.1K Jan 1 00:00 /home/ubuntu/.local/share/Trash/files/_js_on_aws/_templates/Project_JS/_js_on_aws.bkup
pulicanhobbies:~$ find / -name "*_js_on_aws.bkup"
find: '/run/user/1000': Permission denied
find: '/run/snapd.ns': Permission denied
find: '/run/lvm': Permission denied
/home/ubuntu/.local/share/Trash/files/_js_on_aws.bkup
/home/ubuntu/.local/share/Trash/files/_js_on_aws.bkup
pulicanhobbies:~$
```

And let's go back the Python console:

```
>>> # Time to restart the node!
>>> lambinsage.kill(sta=1)

>>> # Finally the SQLAlchemy JSON'd
>>> # So first the JS must remove the current engine to create a new one:
>>> engine=None
>>> lambinsage.kill(sta=1)
>>> exit()
pulicanhobbies:~$
```

And for the game `Hilbert` in the `Static` model:

```
pulicanhobbies:~/ch1ho8_v3 ls -ltrh
drwxrwxrwx 1 pulicanhobbies pulicanhobbies 1.1K Jan 1 00:00 Project_JS
pulicanhobbies:~/ch1ho8_v3 cat main.py
#!/usr/bin/python3
"""
Input video screen from a state
"""
from models import storage
from models.state import State
from models.engine import City

"""
\Description
"""
state_1 = State(name="California")
state_1.save()
state_2 = State(name="Arizona")
print("New state {}".format(state_2))
state_2.save()

city_1 = City(name_invista_1.id, name="Napa")
print("New city {} in the state {}".format(city_1, city_1.state_id))
city_1.save()
city_2 = City(name_invista_1.id, name="Fresno")
print("New city {} in the state {}".format(city_2, city_2.state_id))
city_2.save()
city_3 = City(name_invista_1.id, name="Fresno")
print("New city {} in the state {}".format(city_3, city_3.state_id))
city_3.save()

"""
\Description
"""
print("Number of states:{}".format(
for state_in_state in all_states.items():
    for city in state.cities:
        print("New city {} in the state {}".format(city, state))

pulicanhobbies:~/ch1ho8_v3
pulicanhobbies:~/ch1ho8_v3 ls -ltrh
drwxrwxrwx 1 pulicanhobbies pulicanhobbies 1.1K Jan 1 00:00 Project_JS
drwxrwxrwx 1 pulicanhobbies pulicanhobbies 1.1K Jan 1 00:00 main.py
pulicanhobbies:~/ch1ho8_v3 ./main.py
New state {'id': '7b87c8b0-e41a-451c-ad20-ad1d7b3f7b4b', 'name': 'Napa', 'updated_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000), 'created_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000)}
New state {'id': '7b87c8b0-e41a-451c-ad20-ad1d7b3f7b4b', 'name': 'Napa', 'updated_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000), 'created_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000)}
New city {'id': '7b87c8b0-e41a-451c-ad20-ad1d7b3f7b4b', 'name': 'Napa', 'updated_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000), 'created_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000)} in the state {'7b87c8b0-e41a-451c-ad20-ad1d7b3f7b4b': {'id': '7b87c8b0-e41a-451c-ad20-ad1d7b3f7b4b', 'name': 'California', 'updated_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000), 'created_at': datetime.datetime(2017, 12, 1, 23, 27, 52, 000000)}
```

- `/<slug>` : display a HTML page (inside the tag `body`)
  - Use `{% block content %}`
  - tag with the list of all `object` objects present in `objects.all()` **sorted by** `name` (A->Z) tip
    - tag: **key** descriptor of one `object` in `objects.all()`
  - `/<slug>/<key>` : display a HTML page (inside the tag `body`)
    - Use `{% block content %}`
    - tag: `Name`
    - tag: `Citizen`
    - tag with the list of `child` objects linked to the `child` **sorted by** `name` (A->Z)
    - tag: **key** descriptor of one `child` in `child.all()`
  - **IMPORTANT**
    - tag: "Not found"
- You must use the option `{{ request.path }}` in your route definition
- Import the `{% load static %}` in your templates

IMPORTANT

- Make sure you have a running and valid `group_<your_email>` in your `<LDAP_CONN_ID>` repository (Tool)
- Make sure all tables are created when you run `./<app_name>.py -c <DB_NAME_CONFIGURATION_NAME_IN_DWH_MYSQL_PROFILES_for_you>

- File: 1000_F_1_0_0_0