

INFO 2602 Assignment 1

Description

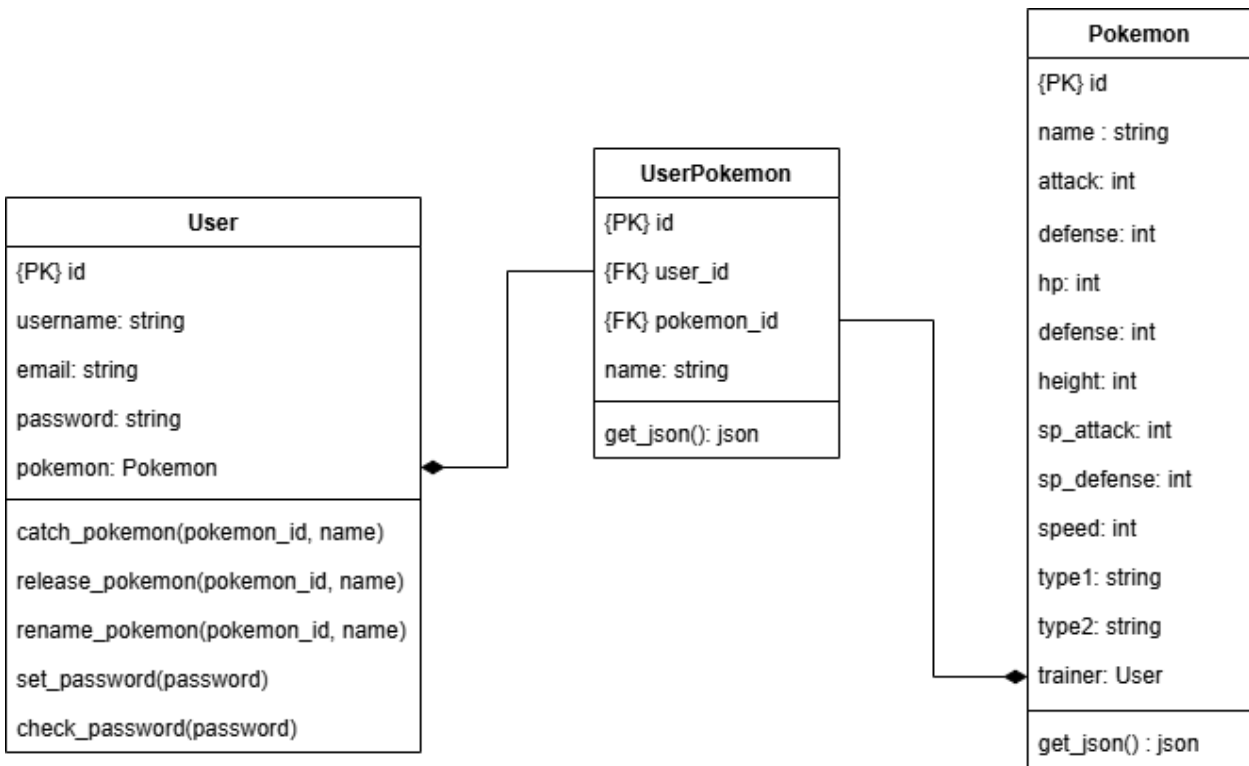
In this assignment, you are to implement a RESTful API using Flask and SQLAlchemy with Token Based authentication.

[WORKSPACE Repo](#)

Under no circumstance you are to fork this repo or commit your assignment code to a public repository. You must submit your code files directly to the submission box.

1. Models

You are to implement the models; User, UserPokemon and Pokemon in models.py.



2. Initialize the Database

Update the `initialize_db()` function in `app.py` to parse the `pokemon.csv` file, create `pokemon` objects and save them to the database. (Note not all pokemon have a second type so when a en empty value is given in that column you should set `type2` to `None`)

You can execute `init` from the command line by opening a terminal (`ctrl + ``) and executing the following.

```
flask init
```

Or by implementing and making a request to the `/init` route.

Note: any changes to the model would require the database to be reinitialized

3. REST API

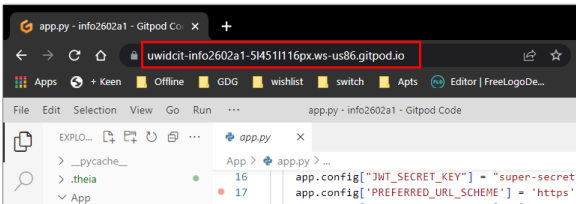
You are required to implement a REST API that adheres **exactly** to the specification of [this postman collection](#). Ensure JWT cookie based authentication is implemented as was done in [lab 3](#).

Testing

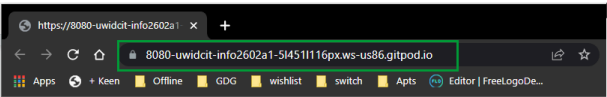
Via Postman

Import the postman collection then create an environment variable called “host” with the url of your gitpod’s application server url (not the url of the editor). Refer to lab3 for further postman instructions.

Editor URL



Application URL

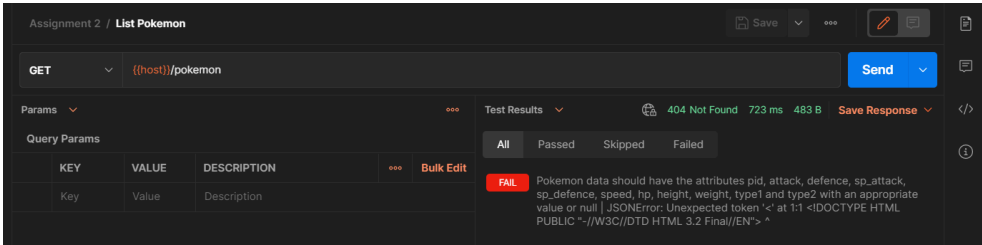
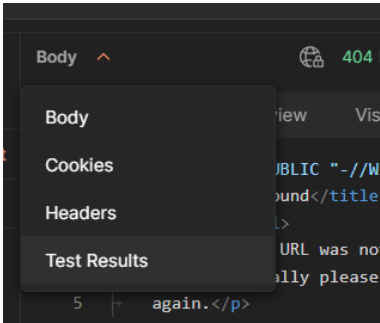


Poke API

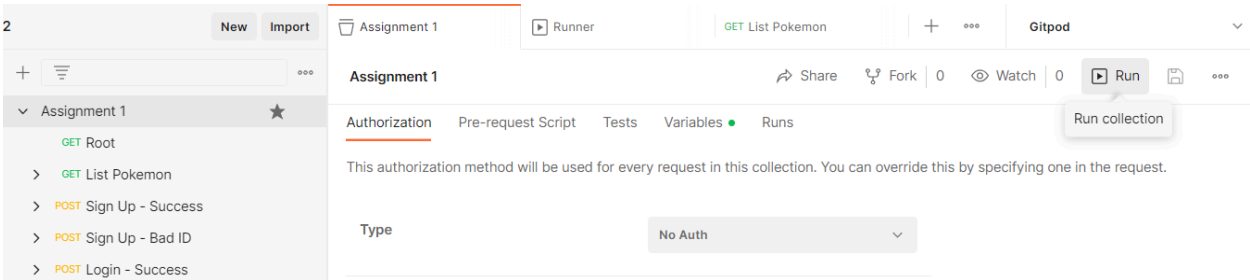
Gitpod		Fork 0		Save	Share	...
	VARIABLE	TYPE	INITIAL VALUE	CURRENT VALUE	Persist All	Reset All
<input checked="" type="checkbox"/>	host	default	https://8080-uwidcit-info2602a1-5f4511116px.ws-us86.gitpod.io	https://8080-uwidcit-info2602a1-5f4...		

Host should be Application url with no backslash at the end

You can then run a request in the collection and view the test result by selecting “Test Results” from the response dropdown.



You can also run the collection to test all of the requests in one run from the collection menu.



Via CLI

You can use the postman cli to test by running the following command in your workspace (must have nodejs installed and first ran “npm install” prior).

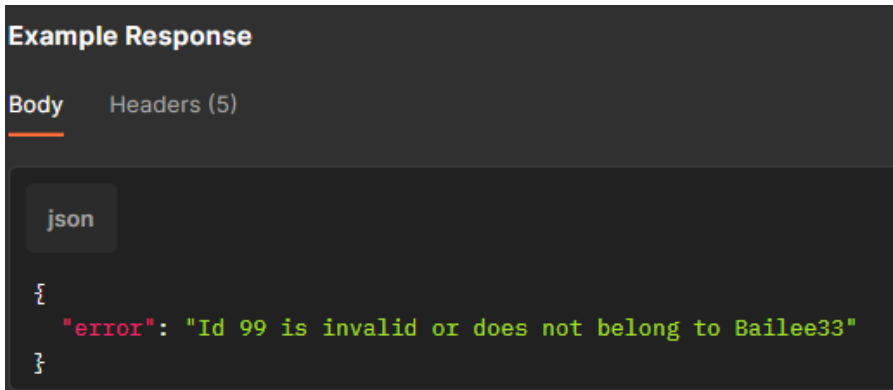
```
npm test
```

The collection will then run in the terminal window



IMPORTANT:

Pay close attention to the spelling, spacing, case and punctuation of the example responses. Your API must match the message structure exactly



This includes your error messages. The testcases are looking for the exact text

```
pm.test('Should return 'id 99 invalid or does not belong to ${username}`', function () {
  var {error} = pm.response.json();
  pm.expect(error).to.eql(`Id 99 is invalid or does not belong to ${username}`);
});
```

Submission

Download the following files

- 1. app.py
- 2. models.py

And drag and drop them directly into the upload box on my elearning.

Do not rename the files, do not include any other files or folders. Your submission must work in the workspace provided.

Your grade is determined by the number of passing test cases.

