Django Training - 7. Testing using Pytest

Materials

1. pytest
2. pytest-django
3. Testing in DRF
4. (optional) Factory Boy and Factory Boy with ORMs

Task

1. Under each app that we have, create a `tests` directory and don't forget `__init__.py`
2. If the app has endpoints, create a file `tests/test_endpoints.py` or `tests/test_views.py`
3. Create a global fixture `auth_client` that returns a function, if that function is passed a user instance, it'll return an instance of DRF's `APIClient` authenticated by that user instance, otherwise, it'll return an instance of `APIClient` authenticated by an arbitrary user instance. example:

```
def test_something(api_client):
    client = api_client(user)
    # or
    client = api_client()



    client.get(url)
```

3. For each endpoint, test the following:
    - If the view has permission classes, test making requests that will obey and disobey the permissions, For example, if a view has `IsAuthenticatedOrReadOnly` permission class, test that making a write and non-authenticated request will return `403 Forbidden` status code
    - If the view is expecting a certain set of required fields, test that making a request with one or more missing fields will return `400` status code and a proper error message
    - If a view is expected to return a set of fields, test that these fields are indeed returned, and that their values match what you expect. For example, if I make a request to `/users/1` I expect that the data returned will be that that of the user whose id is 1
4. Your grade will be affected if a test fails.
5. *What is the purpose of all of this?* Testing is one of the essential tools for building a high quality & solid software, mainly, we write tests for the 2 following reasons:
    - Making sure our software behaves as expected when we support a new feature for the first time
    - Making sure our software still behaves as expected when we modify an existing feature

Guidelines

1. Use this gitignore template
2. List your environment variables (if any) in a `.env.example` file
3. Don't forget to run `manage.py makemigrations` and `python manage.py migrate`
4. You are allowed to refer to any articles, documentation source, questions on StackOverFlow
5. You are not allowed to consult other people through any medium of communication
6. Your grade will be affected if your code isn't clean, you should maintain code cleanliness as much as you can
7. **I should be able to run all your tests by only running `pytest` in the command line.**