Repost FastAPI

This is the FastAPI implementation of the Repost API.

Installation

[Python 3 (https://www.python.org/)](https://www.python.org/) must be installed and accessible through the use of a terminal and the
keyword python or python3. Below are the steps for a proper setup using VENV (Python Virtual Environment).

1. Clone the repository

```
git clone https://github.com/pckv/repost-fastapi.git
```

2. Navigate to the repost-fastapi directory and create a new VENV

```
cd repost-fastapi
python -m venv venv
```

3 (**Linux**). Activate the venv (alternatively: run all commands after this step prefixed with venv/bin/)

```
source venv/bin/activate
```

3 (**Windows**). Activate the venv (alternatively: run all commands after this step prefixed with venv\Scripts\)

```
venv\Scripts\activate
```

4. Install the required packages

```
pip install -r requirements.txt
```

Configurations

Configurations are set by environment variables. Follow the instructions below to run
the server once and a file config.env will be created in the root directory. Otherwise,
the following settings can also be set using exported environment variables.

- **REPOST_CLIENT_ID** - The OAuth2 client_id. Default is repost
- **REPOST_JWT_SECRET** - The secret key used for [JSON Web Tokens (https://jwt.io/)](https://jwt.io/)
- **REPOST_JWT_ALGORIGHTM** - The algorithm used for the key above
- **REPOST_DATABASE_URL** - An SQLAlchemy database url. See
  [Engine Configuration (https://docs.sqlalchemy.org/en/13/core/engines.html)](https://docs.sqlalchemy.org/en/13/core/engines.html)
- **REPOST_ORIGINS** - A list of
  [CORS (https://en.wikipedia.org/wiki/Cross-origin_resource_sharing)](https://en.wikipedia.org/wiki/Cross-origin_resource_sharing) URLs separated by ;

Running the API with uvicorn

[Uvicorn (https://www.uvicorn.org/)](https://www.uvicorn.org/) is a single-threaded ASGI server designed around uvloop to run fast. It is included in the requirements and should be used to run the API.

```
uvicorn repost:app
```

The default host and port is `localhost` and `8000`. They can be changed with the `--host` and `--port` arguments. To run the server publically, set the host to `0.0.0.0` like so.

```
uvicorn repost:app --host 0.0.0.0
```

Running the API with gunicorn

[Gunicorn (https://gunicorn.org/)](https://gunicorn.org/) is a WSGI server that can manage multiple workers. Uvicorn [has a worker for Gunicorn (https://www.uvicorn.org/#running-with-gunicorn)](https://www.uvicorn.org/#running-with-gunicorn) that can be used to run multiple Uvicorn workers. Since Repost is a stateless API, this works perfectly and will allow utilizing more processing power.

```
gunicorn repost:app -w 17 -k uvicorn.workers.UvicornWorker
```

The example above uses 17 workers for a system with 8 CPUs (16 threads + 1 workers). This value can be tweaked to your setup. You can also set the host and port in gunicorn with the `-b` argument, which includes both host and port in the same argument.

```
gunicorn repost:app -b 0.0.0.0:8000 -w 17 -k uvicorn.workers.UvicornWorker
```

Documentation

Documentation for the API is available after deployment at the `/api/swagger` and `/api/docs` endpoints.