

리크루팅 사이트 백엔드 실행법

작성: @정대용

to: @안민규 / 학생 / 전기·정보공학부 @박준영

cc. @Minkyu Lee

아무래도 눈으로 흐름 파악하기에는 도커보다는 직접 돌리는게 좋으니까 도커이미지 없이 설명드리겠습니다

순서: 파이썬 설치 (numpy 때때 3.9 이하여야 함) → requirements 설치 → Redis 설치 (celery worker의 메시지 큐 역할) → Redis 켜기 → 서버 켜기 → celery worker 켜기 → 프론트 켜기

1. 파이썬 3.8 이하 버전으로 (3.9는 불확실, 3.10은 numpy 충돌남) 설치

2. 라이브러리들 설치

- `waffle_recruit_server` 에서

```
$ pip3 install -r requirements.txt
```

디폴트가 3.8이 아니라면, 3.8 설치하고 여러 버전 중 3.8을 명시해주기 위해 아래처럼 실행

```
$ python3.8 -m pip install -r requirements.txt
```

3. 레디스 설치 후 켜기

- 위치는 상관 없이 본인 원하는 곳에

```
$ wget http://download.redis.io/redis-stable.tar.gz
$ tar xvfz redis-stable.tar.gz
$ cd redis-stable
$ make
```

- 위 make한 경로에서 (== `redis-stable` 에서)

```
$ ./src/redis-server
```

- 아래처럼 나오면 성공 (&로 백그라운드 실행하거나, 그냥 켜놓고 다른 터미널에서 작업)

```
=37298, just started
7298:C 14 Jul 2022 23:10:56.634 # Warning: no config file specified, using the default config.
n order to specify a config file use ./redis-server /path/to/redis.conf
7298:M 14 Jul 2022 23:10:56.635 * Increased maximum number of open files to 10032 (it was origi
ally set to 1024).
7298:M 14 Jul 2022 23:10:56.635 * monotonic clock: POSIX clock_gettime

Redis 7.0.3 (2c0b7b81/1) 64 bit

Running in standalone mode
Port: 6379
PID: 37298

https://redis.io

7298:M 14 Jul 2022 23:10:56.636 # Server initialized
7298:M 14 Jul 2022 23:10:56.636 # WARNING overcommit_memory is set to 0! Background save may fa
l under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.c
nf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
7298:M 14 Jul 2022 23:10:56.636 * Loading RDB produced by version 7.0.3
7298:M 14 Jul 2022 23:10:56.636 * RDB age 107830 seconds
7298:M 14 Jul 2022 23:10:56.636 * RDB memory usage when created 1.63 Mb
7298:M 14 Jul 2022 23:10:56.636 * Done loading RDB, keys loaded: 5, keys expired: 0.
7298:M 14 Jul 2022 23:10:56.636 * DB loaded from disk: 0.000 seconds
7298:M 14 Jul 2022 23:10:56.636 * Ready to accept connections
```

(선택) 시간 제한 해제

- `waffle-recruit-web/waffle_recruit_server/check/view.py` 의 139~140 line (자원이 마감되어 제출이 불가합니다) 부분 주석처리해주기
- 아래 사진 원본 클릭하면 자세히 나옵니다

4. 서버 켜기

- `waffle_recruit_server` 에서

```
$ python3 manage.py runserver 혹은
$ python3.8 manage.py runserver
```

- 아래처럼 나오면 성공

```
daeyong@daeyong-VivoBook-ASUSLaptop-X509JA-P1510CJA:~/Desktop/waffle/waffle-recruit-web/waffle_recruit_server$ python3.8 manage.py runserver
Performing system checks...
System check identified no issues (0 silenced).
July 14, 2022 - 14:29:54
Django version 3.0.14, using settings 'waffle_recruit_server.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

5. celery 켜기

- `waffle_recruit_server` 에서

```
$ python3 -m celery -A waffle_recruit_server worker -l info -c 1
혹은
$ python3.8 -m celery -A waffle_recruit_server worker -l info -c 1
```

- 아래처럼 나오면 성공

```
daeyong@daeyong-VivoBook-ASUSLaptop-X509JA-P1510C3A:~/Desktop/waffle/waffle-recruit-web/waffle_recruit_server$ python3.8 -m celery -A waffle_recruit_server worker -l info -c 1
----- celery@daeyong-VivoBook-ASUSLaptop-X509JA-P1510C3A v5.1.2 (sun-harmonics) -----
----- ***** Linux-5.15.0-41-generic-x86_64-with-glibc2.35 2022-07-14 14:22:20 -----
*** **
[config]
** -----> app: waffle_recruit_server:0x7f3c9e314df0
** -----> transport: redis://localhost:6379/0
** -----> results: redis://localhost:6379/0
** -----> concurrency: 1 (prefork)
-- *****> task events: OFF (enable -E to monitor tasks in this worker)
[queues]
-----> celery exchange=celery(direct) key=celery

[tasks]
. solver

[2022-07-14 14:22:20.834: INFO/MainProcess] Connected to redis://localhost:6379/0
[2022-07-14 14:22:20.836: INFO/MainProcess] mingle: searching for neighbors
[2022-07-14 14:22:21.845: INFO/MainProcess] mingle: all alone
[2022-07-14 14:22:21.865: INFO/MainProcess] celery@daeyong-VivoBook-ASUSLaptop-X509JA-P1510C3A ready.
[2022-07-14 14:22:21.868: INFO/MainProcess] Task solver[2f579784-adea-42d7-9a13-b874d89ba5b1] received
```

6. 프론트 켜기

- 아래 사진처럼 `/src/apis/requester.ts` 의 `const isProduction = false;` 로 해서 로컬로 연결 바꿔줘야 합니다

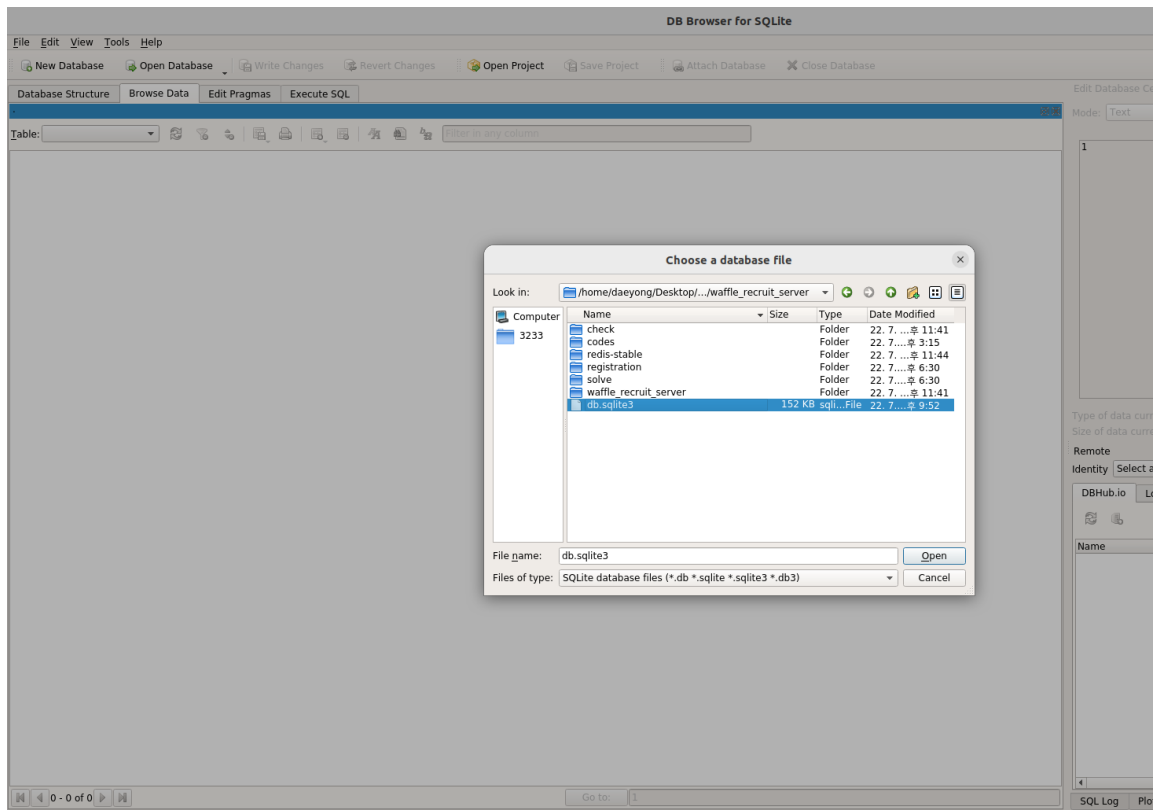
```
1 import axios from 'axios';
2
3 // TODO .env.production
4 const isProduction = false;
5
6 export const requester = axios.create({
7   baseURL: isProduction ? 'https://recruit-api.wafflestudio.com' : 'http://localhost:8080',
8   xsrfCookieName: 'csrftoken',
9   xsrfHeaderName: 'X-CSRFToken',
10   withCredentials: true,
11 });
12
```

- yarn install && yarn start는 더 잘 아실테니..

(선택) DB 확인하기

만약 직접 DB를 조회해보고 싶으시다면,

- <https://sqlitebrowser.org/dl/> 링크 따라서 sqlite3 설치하시고,
- 아까 4. 서버켜기에서 빨간 글씨로 (migrate되지 않은 파일 있음) 비슷한 내용이 땀을 겁니다. 거기에서 하라는 대로 `python3 manage.py migrate` 인가 (요거 커맨드가 정확히 기억이 안나는데 그냥 거기 하란대로 따라 치시면 됩니다.) 그럼 짹짹 뭔가가 되고, 해당 경로 (waffle_recruit_server/ 에) db.sqlite3 파일이 생겼을 겁니다.
- 위 설치한 sqlite3 브라우저를 열고 db.sqlite3파일 импорт 하시면 됩니다.



- 그 다음에 browse data 탭에서 table 선택하시면 가입된 유저, solve한 사람 등등 확인할 수 있어요~

