**설치**

기본적으로 python, postgresql, redis 를 설치해야 한다.

**~\deploy\requirements.txt** : Dependency를 나타낸 파일로 다음을 명령어로 실행하여 설치한다.

ONLINEJUDGE(디렉토리) >> **Pip install -r deploy\requirements.txt**

**~\Oj\dev\_settings.py** : db와 redis에 대한 설정을 하는 부분을 자신이 설치한 것과 맞게 입력하면 된다.

장고는 원래 프로젝트 생성 시 secret key를 자동으로 생성하는데, 여기에는 없기에 직접 만들어야 한다. 구글에 “Django secret key 생성”을 검색하여 무작위 키를 생성하여 ~\**data\config\secret.key**를 생성 및 입력한다.

준비 되었다면 migration을 진행하여 DB를 준비한다.

**Python manage.py migrate**

장고 서버 실행은 다음과 같다.

**Python manage.py runserver**

**~\oj**

|  |  |
| --- | --- |
| **URL** | **기능** |
| ~/api/ | 일반적인 기능에 대한 주소 |
| ~/api/admin/ | 관리자에 관련된 주소 |

**~\settings.py** : django에 대한 설정을 하는 코드

**~\dev\_settings.py** : 개발 환경에서의 데이터베이스와 호스트에 대한 설정을 하는 코드

**~\utils\api**

~\api.py : django에서 제공하는 APIView 및 다른 것들을 이용하기 편하게 재정의했다.

**~\account**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URL** | **Method** | **입력(\* : 필수)** | **기능** | **출력(JSON)** |
| ~/api/login/ | POST | username\*,  password\*,  tfa\_code | 로그인 | Error  [  NULL,  Your account has been disabled,  tfa\_required  Invalid two factor verification code,  Invalid username or password  ],  Data  [  NULL,  Succeeded  ] |
| ~/api/logout/ | GET | - | 로그아웃 | - |
| ~/api/register/ | POST | username\*,  password\*,  email\* | 회원가입 | error  [  NULL,  Username already exists,  Email already exists,  Register function has been disabled by admin  ],  data  [  NULL,  Succeeded  ] |

**~\problem**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URL** | **Method** | **입력(\* : 필수)** | **기능** | **출력(JSON)** |
| ~/api/problem/ | GET | problem\_id,  limit\*,  tag,  keyword,  difficulty | Id에 맞는 문제목록을 반환한다. | Error  [  NULL,  Problem does not exist,  Limit is needed,  ]  Data  {  Results : 문제 목록  total : 개수  } |
| ~/api/pickone/ | GET | - | 무작위로 문제를 하나 고른다. | Error  [  NULL,  No problem to pick  ]  Data : 문제 데이터 |
| ~/admin/problem/ | GET | id,  rule\_type,  keyword, | 문제 읽기 | Error  [  NULL,  Problem does not exist,  Invalid rule\_type  ]  Data  {  Results : 문제 목록  total : 개수  } |
| POST | \_id,  Title\*,  Description\*,  Input\_description\*,  Output\_description\*,  Samples\*,  Test\_case\_id\*,  Test\_case\_score,  Time\_limit\*,  Memory\_limit\*,  Languages\*,  Template\*,  Rule\_type\*,  Io\_mode\*,  Spj\*,  Spj\_language,  Spj\_code,  Spj\_compile\_ok,  Visible\*,  Difficulty\*,  Tags\*,  Hint,  Source,  Share\_submission\* | 문제 추가 | Error  [  NULL,  Display ID is required,  Display ID already exists  ]  Data : 입력과 같은 데이터 |
| PUT | POST의 입력과 같다 | 문제 수정 | Error  [  NULL,  Problem does not exist,  Display ID is required,  Display ID already exists  ],  Data : NULL |
| DELETE | Id\* | 문제 삭제 | Error  [  NULL,  Invalid parameter, id is required,  Problem does not exist  ],  Data : NULL |