

# Django - Query set 쿼리 셋

## 쿼리 고도화 (JOIN)

- **select\_related**

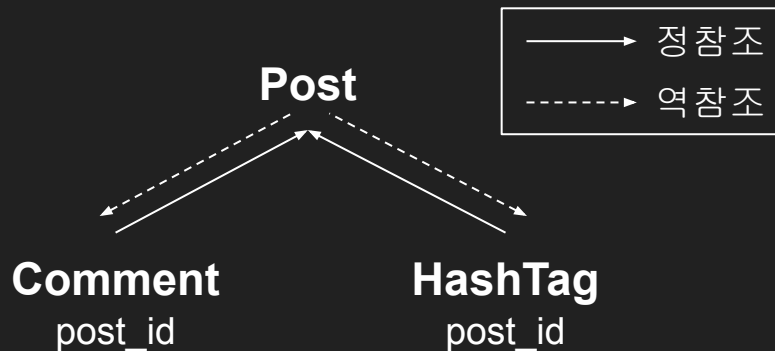
정 참조: ForeignKey, OneToOne

역 참조: OneToOne

- **prefetch\_related**

정 참조: ManyToMany

역 참조: ForeignKey, ManyToMany



**Comment, HashTag -> Post**

- **select\_related**

**Post -> Comment, HashTag**

- **prefetch\_related**

## Django - Query set 쿼리 셋

### Detail View

- post\_id

### Comment, HashTag -> Post

- select\_related  
Comment.objects.select\_related('post').filter(post\_\_pk=pk)  
HashTag.objects.select\_related('post').filter(post\_\_pk=pk)

### Post -> Comment, HashTag

- prefetch\_related  
Post.objects.prefetch\_related('comment\_set', 'hashtag\_set').get(pk=pk))

## Django 관계

REST Framework API -> 통신  
-> HTTP 통신

클라이언트, 서버  
요청: (서버) url + method -> 브라우저: 클라이언트

외부 클라이언트 이용 -> Postman  
curl http -> 터미널

## API (요구)문서

### 첫 화면 목록에 대한 리스트

**url:** /blog/  
**method:** GET

#### 성공

- msg: 리스트 목록 전송 완료, code: 200

#### 실패

- msg: 서버 오류, code: 500

### 로그인

**url:** /user/login/  
**method:** POST

#### 성공

- msg: 로그인 성공, code: 200

#### 실패

- msg: 없는 아이디, code: 400
- msg: 로그인 형식 오류, code: 400

## Django - 세션, 쿠키

세션: 서버

쿠키: 클라이언트 (브라우저) -> 세션 ID

→ Django에서 세션 유효 기간(만료 시점) 설정하기

```
SESSION_ENGINE = 'django.contrib.sessions.backends.db'
```

```
SESSION_COOKIE_AGE = 86400
```

```
SESSION_EXPIRE_AT_BROWSER_CLOSE = True
```

## Django - static files, media files

**Static:** 서버에서 제공하는 파일들 (html, css, js, image -> 용량이 큼, 잘 안 변함)

`settings.py`  
`STATIC_URL`

**Media:** 사용자에게 받아온 파일들 (image, file)

# Django - static files, media files

## Static

1. static 파일을 업로드
2. django -> 고정해서 사용: `python manage.py collectstatic`  
static 경로에 있는 파일들을 한 곳으로 모아준다 ->  
`staticfiles`

app1  
static/  
css1-1

app2  
static/  
css2-1

`python manage.py collectstatic`

staticfiles  
app1  
static/  
css1-1  
app2  
static/  
css2

# Django - Throttling

## Throttling

- 로그인한 유저: 한 가지 요청을 일정 시간 동안 50번
- 로그인하지 않은 유저: 한 가지 요청을 일정 시간 동안 5번

## REST Framework - Throttling

- settings.py



# Django - Translate

## Translate

- 언어 -> 단어, 문장 -> 번역 기능 + 템플릿 엔진(태그)

1. settings.py 안에

```
LANGUAGE_CODE = 'en-us'  
LANGUAGES = [  
    ('en', 'English'),  
    ('kr', 'Korean')  
]
```

추가

2. templates 안에 {% load i18n %} {% trans " %" } 바꾸고 싶은 부분 표시
3. python manage.py makemessages -l kr
4. po 파일의 메시지 직접 입력 (영어 -> 한국어)
5. python manage.py compilemessages