

# Django

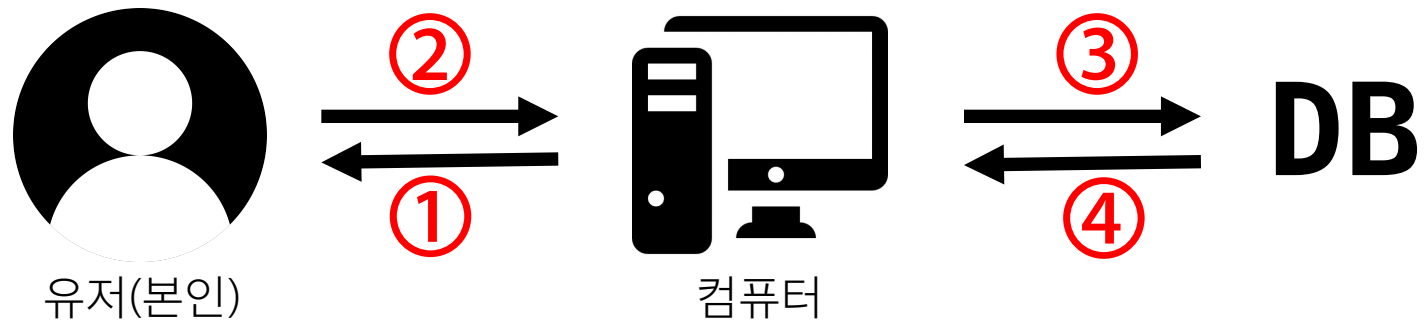
## 회원가입 인증메일 구현하기

20180913 피로그래밍 세션

# 순서

- 1 (본인) 인증
- 2 회원가입/로그인 직접 해보기
- 3 이메일 인증 순서
- 4 구현하기

## (본인)인증



- ① 암호 생성
- ② 암호 전달
- ③ DB와 대조
- ④ response

회원가입/로그인 직접 해보고 오기

# 이메일 인증 순서

## 1 EmailConfirm 모델,

key(models.CharField)와 is\_confirmed(models.BooleanField) 필드 생성

## 2 첫 로그인 시, email 보냄 (이때, **key** 생성)

- server : DB에 key 저장
- client : (본인만 접근가능한) **email로 key 전송받음** / <a> 활용

## 3 <a> 링크로 들어가면 ( GET으로 가져온 key == DB에 저장된 key ) 판단

- TRUE -> EmailConfirm.is\_confirmed = true 로
- FALSE -> 예외처리 ex. 잘못된 접근입니다.

---

## 4 인증 기능을 활용하는 방법

- 함수로 (views) : 아예 페이지에 못들어오게 접근 차단
- 템플릿언어(html)으로 : 인증 여부에 따라 두 가지 페이지가 존재할 때

# 구현하기

## 1 settings.py

```
EMAIL_HOST = 'smtp.gmail.com'  
EMAIL_HOST_USER = 'pirogramming@gmail.com'  
EMAIL_HOST_PASSWORD = 'django1!'  
EMAIL_PORT = 587  
EMAIL_USE_TLS = True
```

## 2 accounts/views.py

```
def send_confirm_mail(user):  
def email_confirm_sent(request):  
def confirm_email(request):
```

## 3 accounts/models.py

```
class EmailConfirm(models.Model):
```

## 4 accounts/hash\_generator.py

```
def generate_random_string(length=60):
```

## 5 accounts/templates/accounts /email\_confirm\_sent.html

## 6 accounts/urls.py

```
path('confirm/', views.confirm_email, name='confirm_email'),  
path('confirm/sent/', views.email_confirm_sent,  
      name='email_confirm_sent'),
```

## 7 accounts/views.py

```
def login_view(request):  
    ... login_and_redirect_next(request, user)  
def login_and_redirect_next(request, user):
```

## 8 활용하기(html)