

중복 로그인 막기 (개선)

(중급편) 장고 기본 인증 뽐내기

Ask Company

현재 이슈

로그인이 되어있는 상황에서 다시 로그인을 시도할 경우, **user_logged_in** 시그널 리시너 내에서 참조되는 **request.session.session_key**값은 **None**이 됩니다.

그래서, 다음 코드에서 IntegrityError가 발생합니다.

```
# accounts/models.py
session_key = request.session.session_key # 여기에서 None
UserSession.objects.create(user=user, session_key=session_key)
```

대응 방법 1) 중복 로그인을 원천적으로 막습니다.

`/accounts/login/` 뷰에서 처리 전에, 로그인 상황일 경우 다른 페이지로 `redirect`시킵니다.

그런데, 일반 유저 로그인 (`/accounts/login/`) 후에, admin 로그인 (`/admin/`)을 시도하는 경우는 막을 수 없습니다.

그래서, 이 방법은 불가.

대응 방법 2) 기존 `user_logged_in` 시그널 리시버 로직 개선

`user_logged_in` 시그널 리시버 내에서 참조되는 값

- `user` : 로그인을 수행한 user 객체
- `request.session` : **로그인 전에** 생성된 세션
 - 로그아웃 상황에서 로그인을 했을 경우, 세션을 재활용하며 **`session_key`가 재생성**됩니다. (OK)
 - 로그인 상황에서 다른 아이디로 로그인을 했을 경우, **로그인 전 세션을 flush** 하기에 **`session_key`가 `None`**이 됩니다.

그럼, 로그인 상황에서 다른 아이디로 로그인을 했을 경우 **session_key**는 언제 획득을 할 수 있습니까?

1. 현재 세션에 session_key가 없을 경우, SessionMiddleware의 process_response(request, response) 함수 내 **request.session.save()**를 통해서 세션키가 할당됩니다.
2. 즉, SessionMiddleware의 response 처리 후에 접근이 가능합니다.

```
# accounts/models.py
def kicked_my_other_sessions(sender, request, user, **kwargs):
    # 단순히 로그인 플래그만 남깁니다.
    user.is_user_logged_in = True

user_logged_in.connect(kicked_my_other_sessions)

# accounts/middleware.py
from .models import UserSession

class KickMiddleware(MiddlewareMixin):
    def process_response(self, request, response):
        # 로그인 플래그를 확인합니다.
        is_user_logged_in = getattr(request.user, 'is_user_logged_in', False)

        if is_user_logged_in:
            for user_session in UserSession.objects.filter(user=request.user):
                session_key = user_session.session_key
                session = SessionStore(session_key)
                # session.delete()
                session['kicked'] = True
                session.save()
                user_session.delete()

            session_key = request.session.session_key # 로그인 전 세션
            UserSession.objects.create(user=request.user, session_key=session_key)

        return response
```

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'accounts.middleware.KickMiddleware',    # SessionMiddleware 응답 처리 후에 호출되어야만 합니다.  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    # ...  
]
```

*Life is short,
you need Python3 and Django.*

여러분의
파이썬/장고 페이스메이커가
되겠습니다. ;)



ASK
C O M P A N Y