

# 중복 로그인 막기

## (중급편) 장고 기본 인증 뽐내기

*Ask Company*

# 만들어볼 것

한 **User**당 하나의 로그인만 허용하고 싶어요.

# 관련한 주요 Values

```
def myview(request):  
    # 현재 로그인 User  
    #   - 로그인 상태일 때: AnonymousUser 객체  
    #   - 로그아웃 상태일 때: User 객체  
    request.user  
  
    # 현재 요청에 대한 Client IP  
    request.META.get('REMOTE_ADDR')  
  
    # 현재 로그인 세션 키  
    #   - 로그인 상태일 때: 길이 32의 hexstring  
    #   - 로그아웃 상태일 때: None  
    request.session          # django.contrib.sessions.SessionMiddleware에 의해서 지정  
    request.session.session_key
```

# UserSession 모델

유저가 로그인할 때, **session\_key** 저장

```
# accounts/models.py
class UserSession(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE, editable=False)
    session_key = models.CharField(max_length=40, editable=False)
    created_at = models.DateTimeField(auto_now_add=True)
```

# 유저 로그인 시점 탐지 <sup>1</sup>

**django.contrib.auth.signals.user\_logged\_in** 시그널을 통해 **user** 로그인 시점에 호출될 함수 지정

```
# accounts/models.py

from django.contrib.auth.signals import user_logged_in

def kick_my_other_sessions(sender, request, user, **kwargs):
    # TODO: 구현

user_logged_in.connect(kick_my_other_sessions, dispatch_uid='user_logged_in')
```

---

<sup>1</sup> 호출시점 코드: [django/contrib/auth/ init .py#L161](#)

# User의 다른 세션들을 제거(강제 로그아웃)하고, 새로이 기록

```
# accounts/models.py
```

```
from importlib import import_module
```

```
# backends: db/cached 등
```

```
SessionStore = import_module(settings.SESSION_ENGINE).SessionStore
```

```
def kick_my_other_sessions(sender, request, user, **kwargs):
```

```
    for user_session in UserSession.objects.filter(user=user):
```

```
        session_key = user_session.session_key
```

```
        session = SessionStore(session_key) # get session
```

```
        session.delete()
```

```
    session_key = request.session.session_key
```

```
    UserSession.objects.create(user=user, session_key=session_key)
```

# 강제 로그아웃 시에 메시지 안내하기

# askdjango/settings.py

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'accounts.middleware.KickedMiddleware', # messages framework 사용을 위해  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```



# askdjango/middleware.py

```
from django.conf import settings
from django.contrib import messages
from django.contrib.auth import logout as auth_logout
from django.shortcuts import redirect
from django.utils.deprecation import MiddlewareMixin

class KickedMiddleware(MiddlewareMixin):
    def process_request(self, request):
        kicked = request.session.pop('kicked', None)
        if kicked:
            messages.info(request, '동일 아이디로 다른 브라우저 웹사이트에서 로그인이 감지되어, 강제 로그아웃되었습니다.')
            auth_logout(request)
            return redirect(settings.LOGIN_URL)
```

# accounts/models.py

```
def kick_my_other_sessions(sender, request, user, **kwargs):
    for user_session in UserSession.objects.filter(user=user):
        session_key = user_session.session_key
        session = SessionStore(session_key) # get session
        # session.delete()
        session['kicked'] = True
        # 본디 session middleware에서 save() 처리를 하는 데, 지금은 middleware를 쓰지 않으니 직접 save() 호출
        session.save()

    session_key = request.session.session_key
    UserSession.objects.create(user=user, session_key=session_key)
```

# 더 살펴보기

- 한 아이피 내에서 다수 세션 허용을 해볼 수도 있습니다.

# 장고 서버에 접속한 아이피

```
client_ip = request.META.get('REMOTE_ADDR', None)
```

# 클라이언트와 장고서버 중간에 Proxy server가 있을 경우 (ex: Load Balancer, nginx 등)

```
client_ip = request.META.get('HTTP_X_FORWARDED_FOR', None)
```

*Life is short,  
Use Python3/Django.*

여러분의  
파이썬/장고 페이스메이커가  
되겠습니다. ;)



**ASK**  
C O M P A N Y