

Django - Mixins

- Django에서 미리 정의된 클래스
- 인증, 권한, 폼 등 하나의 부가 기능들을 추가로 사용할 수 있게 제공
 - **LoginRequiredMixin**: 인증(로그인), 인증되지 않은 경우 로그인 페이지로 리디렉션
 - **PermissionRequiredMixin**: 권한, 사용자에게 필요 권한이 없으면 액세스 거부 (HTTP 403 Forbidden)
 - **UserPassesTestMixin**: 사용자가 특정 조건을 통과할 수 있는 지 테스트
 - **FormMixin**: 폼 처리 기능
 - **TemplateView, ListView, DetailView, CreateView, UpdateView, DeleteView**

Django - class (views.py)

Request -> Response

```
class List(View):  
    def get(self, request):  
        return redirect or render
```

request.user -> Anonymous User

request.user -> id

Django auth

- login(request, user): user->auth.models.User
- logout(request)
- authenticate(request, username, password)

Server - Session, Cookie

JWT
OAuth

데이터 저장 -> 사용자, 서버 간 원활한 통신을 위해

Session: 서버쪽에 저장

-> 보안이 필요한 정보

Cookie: 클라이언트쪽에 저장 ->
브라우저

-> 사라져도 괜찮은 정보 (가지고 있으면
편한)

HTTP 통신 -> Stateless

: 요청 -> 응답 ---> 편리하게 상태를
유지

서버 - 세션

쿠키 - 세션 ID (서버에서 발급, HTTP
header에 넣어서 요청)

Server - Cache

데이터 저장 -> 사용자, 서버 간 원활한
통신을 위해

Cache 저장소: 자주 사용되는(용량이 큰, 자주 바뀌지 않는) 데이터를 미리 불러와서
저장해놓는 곳

CSS, JS, Images -> 브라우저 캐시 저장소 or 캐시 서버