Ask Company

리액트와 함께 장고 시작하기 / 장고 Views

뷰 장식자 (View Decorators)

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

장식자 (Decorators)

어떤 함수를 감싸는 (Wrapping) 함수

```
from django.contrib.auth.decorators import login_required
from django.shortcuts import render



@login_required
def protected_view1(request):
    return render(request, 'myapp/secret.html')



def protected_view2(request):
    return render(request, 'myapp/secret.html')

protected_view2 = login_required(protected_view1)
```

몇 가지 장고 기본 Decorators

django.views.decorators.http

require_http_methods, require_GET, require_POST, require_safe

지정 method가 아닐 경우, HttpResponseNotAllowed 응답 (상태코드 405) 반환

django.contrib.auth.decorators

user_passes_test : 지정 함수가 False를 반환하면 login_url로 redirect

login_required : 로그아웃 상황에서 login_url로 redirect

permission_required : 지정 퍼미션이 없을 때 login_url로 redirect

django.contrib.admin.views.decorators

staff_member_required : staff member가 아닐 경우 login_url로 이동

https://docs.djangoproject.com/en/3.0/topics/http/decorators/

CBV에 장식자 입히기 #1

가독성이 좋지 않아요.

요청을 처리하는 함수를 Wrapping하기

```
from django.contrib.auth.decorators import login_required
from django.views.generic import TempalteView

class SecretView(TemplateView):
    template_name = 'myapp/secret.html'

view_fn = SecretView.as_view()

secret_view = login_required(view_fn)    # 이미 생성된 함수에 장식자 입힐 수도 있어요.
```

CBV에 장식자 입히기 #2

괜히 dispatch 재정의

```
from django.contrib.auth.decorators import login_required
from django.utils.decorators import method_decorator
from django.views.generic import TempalteView


class SecretView(TemplateView):
    template_name = 'myapp/secret.html'

    # 클래스 멤버함수에는 method_decorator를 활용
    @method_decorator(login_required)
    def dispatch(self, *args, **kwargs):
        return super().dispatch(*args, **kwargs)

secret_view = SecretView.as_view()
```

CBV에 장식자 입히기 #3

```
from django.contrib.auth.decorators import login_required
from django.utils.decorators import method_decorator
from django.views.generic import TempalteView

# 클래스에 직접 적용할 수 도 있어요.
@method_decorator(login_required, name='dispatch')
class SecretView(TemplateView):
    template_name = 'myapp/secret.html'

secret_view = SecretView.as_view()
```

인생은 짧습니다.
파이썬/장고를 쓰세요.

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

-

Ask Company