

Ask Company



리액트와 함께 장고 시작하기 / 장고 Form

# ModelForm

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

# ModelForm

장고 Form을 상속

지정된 Model로부터 필드정보를 읽어들이며, Form Fields를 세팅

내부적으로 Model Instance를 유지

유효성 검증에 통과한 값들로, 지정 Model Instance로의 저장(save) 지원  
(Create 또는 Update)

```
class PostForm(forms.ModelForm):  
    class Meta:  
        model = Post  
        # 전체 필드 지정. 혹은 list로 읽어올 필드명 지정  
        fields = '__all__'
```

# Form/ModelForm

```
from django import forms
from .models import Post

class PostForm(forms.Form):
    title = forms.CharField()
    content = forms.CharField(widget=forms.Textarea)
```

# 생성되는 Form Field는 PostForm과 거의 동일

```
class PostModelForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = ['title', 'content']
```

# ModelForm.save(commit=True)

Form의 cleaned\_data를 Model Instance 생성에 사용하고, 그 Instance를 리턴

commit=True

model instance의 save() 및 form.save\_m2m()을 호출

form.save() != instance.save()

commit=False

instance.save() 함수 호출을 지연시키고자할 때 사용

<https://github.com/django/django/blob/2.1/django/forms/models.py#L443>

# ModelForm.save 컨셉 구현

```
from django import forms
from .models import Post
```

```
class PostForm(forms.Form):
    title = forms.CharField()
    content = forms.CharField(widget=forms.Textarea)

    def save(self, commit=True):
        post = Post(**self.cleaned_data)
        if commit:
            post.save()
        return post
```

컨셉만 보세요.

Production 레벨에 사용할만한 코드는 아닙니다.

<https://github.com/django/django/blob/2.1/django/forms/models.py#L443>

# View에서의 ModelForm 처리 (New)

```
from django.shortcuts import render
from .forms import PostModelForm
from .models import Post

def post_new(request):
    if request.method == 'POST':
        form = PostModelForm(request.POST, request.FILES)
        if form.is_valid():
            post = form.save()
            return redirect(post)
    else:
        form = PostModelForm()
    return render(request, 'myapp/post_form.html', {
        'form': form,
    })
```

# View에서의 ModelForm 처리 (Edit)

```
from django.shortcuts import get_object_or_404, render
from .forms import PostModelForm
from .models import Post

def post_edit(request, pk):
    post = get_object_or_404(Post, pk=pk)

    if request.method == 'POST':
        form = PostModelForm(request.POST, request.FILES, instance=post)
        if form.is_valid():
            post = form.save()
            return redirect(post)
    else:
        form = PostModelForm(instance=post)
    return render(request, 'myapp/post_form.html', {
        'form': form,
    })
```

# ModelForm.save(commit=False) 예시 (1)

```
from django.db import models
```

```
class Comment(models.Model):  
    author = models.CharField(max_length=20)  
    message = models.TextField()  
    ip = models.CharField(max_length=15)
```

```
# myapp/forms.py
```

```
class CommentForm(forms.ModelForm):  
    class Meta:  
        model = Comment  
        # ip필드는 유저로부터 입력받지 않고, 프로그램으로 채워넣을 것이기에 제외  
        fields = ['author', 'message']
```



# ModelForm.save(commit=False) 예시 (2)

```
def comment_new(request):  
    if request.method == 'POST':  
        form = CommentForm(request.POST, request.FILES)  
        if form.is_valid():  
            comment = form.save(commit=False) 필수필드인 ip필드 값을 아직 채우지 않았기에  
            comment.ip = request.META['REMOTE_ADDR'] # IP를 기록하고  
            comment.save() # save()  
            return redirect('/')  
        else:  
            form = CommentForm()  
  
    return render(request, 'myapp/comment_form.html', {'form': form})
```

# ModelForm.save(commit=False) 예시 (3)

```
def comment_edit(request, pk):
    comment = get_object_or_404(Comment, pk=pk)

    if request.method == 'POST':
        form = CommentForm(request.POST, request.FILES, instance=comment)
        if form.is_valid():
            comment = form.save()
            return redirect('/')
    else:
        form = CommentForm(instance=comment)

    return render(request, 'myapp/comment_form.html', {'form': form})
```

이미 comment.ip필드에 값이 채워져있기에  
form.save() 만으로도 충분한 처리

# 당부: Form을 끝까지 써주세요.

```
form = CommentForm(request.POST)
if form.is_valid():
    # request.POST : 폼 인스턴스 초기 데이터
    message = request.POST['message'] # request.POST를 통한 접근 : BAD !!!
    comment = Comment(message=message)
    comment.save()
return redirect(post)
```

```
form = CommentForm(request.POST)
if form.is_valid():
    # form.cleaned_data : 폼 인스턴스 내에서 clean함수를 통해 변환되었을 수도 있을 데이터
    message = form.cleaned_data['message'] # form.cleaned_data를 통한 접근 : GOOD !!!
    comment = Comment(message=message)
    comment.save()
return redirect(post)
```

Life is short.  
You need Python and Django.

I will be your pacemaker.

