

SoftwareEngineering

Assignment (5)

Created By :
Dheya Al-Hadhrami

Supervisor :
Eng. Malek Al-Mossanif

Introduction

(Django)

سنغوص أولاً في عالم `QuerySet` لنكتشف كي
يمكنك سحب البيانات من قاعدة البيانات بذكاء وكفاءة،
متجنبين الأخطاء الشائعة التي تبطئ أداء الموقع.

ثم ننتقل إلى الجزء الثاني لنتناول نماذج الإدخال (`Forms`)،
حيث سنقارن بين بناء النماذج يدوياً والاعتماد على `ModelForm`،

Assignment Objectives

- Django QuerySet .

- هندسة مدخلات المستخدم باستخدام نماذج Django .

Part I:

Django QuerySet

ينقسم سلوك دوال QuerySet إلى فئتين رئيسيتين:

- دوال تُرجع QuerySet جديد (قابلة للتسلسل).
- دوال لا تُرجع QuerySet (تُنهِي الاستعلام).

1. دوال تُرجع QuerySet جديد (قابلة للتسلسل)

الدالة	الوصف	مثال
<code>()filter</code>	تُرجع QuerySet جديد يحتوي على الكائنات التي تطابق شروط البحث المحددة.	<code>Article.objects.filter(status='published')</code>
<code>()exclude</code>	تُرجع QuerySet جديد يحتوي على الكائنات التي لا تطابق شروط البحث.	<code>Article.objects.exclude(author__name='John')</code>
<code>()annotate</code>	تُضيف حقلاً محسوباً (تجميعاً) لكل كائن في الـ QuerySet.	<code>Author.objects.annotate(num_articles=Count('article'))</code>
<code>()order_by</code>	تتحكم في ترتيب نتائج الـ QuerySet.	<code>Article.objects.order_by('-pub_date')</code>
<code>()distinct</code>	تزيل الصفوف المكررة من النتائج.	<code>Article.objects.values('author__name').distinct()</code>
<code>()select_related</code>	تحسين الأداء عن طريق جلب الكائنات ذات الصلة (ForeignKey, OneToOne) في استعلام واحد.	<code>Article.objects.select_related('author')</code>
<code>()prefetch_related</code>	تحسين الأداء عن طريق جلب الكائنات ذات الصلة (ManyToMany, Reverse FK) في استعلام منفصل.	<code>Author.objects.prefetch_related('articles')</code>
<code>()only()</code> / <code>defer</code>	تتحكم في الحقول التي يتم جلبها من قاعدة البيانات لتحسين الأداء.	<code>Article.objects.only('title', 'author')</code>

2. دوال لا تُرجع QuerySet (تُنهِي الاستعلام)

الدالة	الوصف	القيمة المُرجعة	مثال
<code>()get</code>	تسترجع كائنًا واحدًا فقط. تثير خطأ إذا لم يتم العثور على أي كائن أو تم العثور على أكثر من واحد.	Model Instance	<code>Article.objects.get(pk=1)</code>
<code>()count</code>	تُرجع عدد الكائنات في الـ QuerySet.	Integer	<code>Article.objects.filter(status='published').count()</code>
<code>first() / last()</code>	تُرجع الكائن الأول أو الأخير في الـ QuerySet.	Model Instance or None	<code>Article.objects.order_by('pub_date').first()</code>
<code>()aggregate</code>	تُرجع قاموسًا من القيم المجمعة (مثل المتوسط، المجموع) على الـ QuerySet بأكمله.	Dictionary	<code>Article.objects.aggregate(avg_rating=Avg('rating'))</code>
<code>()exists</code>	تتحقق مما إذا كان الـ QuerySet يحتوي على أي نتائج. أكثر كفاءة من <code>count() > 0</code> .	Boolean	<code>Article.objects.filter(author=user).exists()</code>
<code>()create</code>	تُنشئ كائنًا جديدًا وتحفظه في قاعدة البيانات في خطوة واحدة.	Model Instance	<code>Author.objects.create(name='Jane Doe')</code>
<code>()update</code>	تُحدّث جميع الكائنات في الـ QuerySet بقيم جديدة.	Integer (rows affected)	<code>Article.objects.filter(status='draft').update(status='published')</code>
<code>()delete</code>	تُحذف جميع الكائنات في الـ QuerySet.	Tuple (count, type counts)	<code>Article.objects.filter(is_archived=True).delete()</code>

Part II:

هندسة مدخلات المستخدم

باستخدام نماذج Django

يوفر Django طريقتين رئيسيتين لإنشاء النماذج لكل منهما حالات استخدام مثالية.

forms.Form (يدوي)

مثالي لنماذج الاتصال، البحث، أو أي بيانات لا يتم حفظها مباشرة في نموذج قاعدة بيانات.

```
# forms.py
from django import forms

class ContactForm(forms.Form):
    name = forms.CharField(max_length=100)
    email = forms.EmailField()
    message = forms.CharField(widget=forms.Textarea)
```

يوفر Django طريقتين رئيسيتين لإنشاء النماذج لكل منهما حالات استخدام مثالية.

`forms.ModelForm` (تلقائي)

الخيار الأفضل لعمليات الإنشاء/التحديث (CRUD) المرتبطة بنماذجك، مما يقلل من تكرار الكود.

```
# forms.py
from django.forms import ModelForm
from .models import Article

class ArticleForm(ModelForm):
    class Meta:
        model = Article
        fields = ['headline', 'content']
```

Thank You

Looking forward to seeing you
in the second assignment ;)