Ajuntament de Barcelona

Cibernàrium

Django: El framework web pels exigents amb deadlines

Capacitació Tecnològica per a Professionals i Empreses

Formador
Isaac Gallego Pla

Barcelonactiva

Temes per la sessió 3

- 1 – Managers
- 2 – Cache
- 3 – Middleware
- 4 – Signals
- 5 – Commands
- 6 – Més conceptes?
- 7 – Testing
- 8 – Projecte 3: Interconexió

1 – Managers

- 1.1 – Centralitzem les queries
- 1.2 – Creant el manager
- 1.3 – Afegint funcions de query
- 1.4 – Canvi de comportament per defecte

1.1 – Centralitzem les queries

- Els managers, principalment, son gestors de consultes
- La seva funció principal és fer de proxy de comunicació entre el model i la base de dades
- És una bona pràctica encapsular les queries necessàries per l'aplicació en el manager
  - Això ens ofereix un punt centralitzat d'errors

1.2 – Creant el manager

- Hereda de django.db.Manager
- El manager per defecte es diu “objects” (ja el coneixeu)
- Podem sobreescriure'l, o tenir diversos managers per cada model

1.3 – Afegint funcions de query

- Les funcions de query son funcions de classe
- Una bona estratègia és generar una funció per cada query complicada que tinguem al sistema
- Es poden encadenar

1.3 – Afegint funcions de query

```
class ExampleManager(models.Manager):
    def query1(self):
        return self.filter(field1 = "value1")

    def query2(self):
        return self.all().order_by("-field2")
```

1.4 – Canvi de comportament per defecte

- get_queryset()
  - Funció que filtra el comportament per defecte
  - Molt útil per generar filtrats en managers per defecte
- Exemple

```
def get_queryset(self):
    return super(ExampleManager, self).get_queryset().filter(author='Roald Dahl')
```

2 – Cache

- 2.1 – Configuració
- 2.2 – Tipus de cache disponibles

2.1 – Configuració

- La configuració es centralitza en settings.py
- Tenim disponibles diferents tipus de cache
  - Memcache
  - Fitxers
  - DB
  - ...
- La configuració varia segons el tipus de cache escollit
- La documentació oficial es troba a:

  https://docs.djangoproject.com/en/dev/topics/cache/

2.2 – Tipus de cache disponibles

- Cache per site
  - Genera una cache global, de tot el site
- Cache per view
  - Genera una caché per vista
- Cache per fragment de template
  - Genera una caché per un fragment de plantilla
- Cache de baix nivell
  - Genera una cache personalitzada i controlada mitjançant claus a mida

3 – Middleware

- 3.1 – Què son?
- 3.2 – Middlewares implementats
- 3.3 – Exemple d'un middleware senzill

3.1 – Què son?

- Els middleware son peces de software que intercepten un flux de dades en un punt determinat
- Serveixen per canviar el comportament de l'aplicació o afegir noves funcionalitats transversals
- Tenim diversos middlewares pre-implementats

  https://docs.djangoproject.com/en/1.5/topics/http/middleware/

  https://docs.djangoproject.com/en/1.5/ref/middleware/

3.2 – Middlewares implementats

- Cache middleware: per gestionar els 3 primers nivells de cache
- Gzip middleware: genera les response en format comprimit (sempre que el browser ho accepti)
- Locale middleware: ofereix suport de multiidioma
- Session middleware: ofereix suport de sessions
- Transaction middleware: ofereix control transaccional lligat al cicle request/response
- ...

3.3 – Exemple d'un middleware senzill

```
class MyMiddleware(object):
    def process_request(self, request):
        if request.user.is_authenticated:
            profile = request.user.profile
            profile.today_requests += 1
            profile.save()
    def process_exception(self, request, exception):
        if request.user.is_authenticated:
            profile = request.user.profile
            profile.today_exceptions += 1
            profile.save()
```

```
MIDDLEWARE_CLASSES = (
    ...
    'myapp.mymodule.MyMiddleware',
    )
```

4 – Signals

- 4.1 – Què son?
- 4.2 – Exemple d'ús

4.1 – Què son?

- Els signals son l'enfocament de Django per crear una estructura de programació basada en events
- Es gestiona a nivell d'events emergents i callbacks
- Un objecte defineix quins events genera
- Altres objectes es subscriuen a aquests events i ofereixen una funció de callback que fa una feina determinada

  https://docs.djangoproject.com/en/dev/topics/signals/

4.2 – Exemple d'ús

```
from django.db.models.signals import post_save

def update_other_model(sender, **kwargs):
    instance = kwargs['instance']
    if kwargs['created']:
        om = OtherModel()
        om.main_instance = instance
        om.save()
post_save.connect(update_other_model, sender=MyModel)
```

5 – Commands

- 5.1 – Què son?
- 5.2 – Exemple d'ús

5.1 – Què son?

- Son scripts python amb l'entorn del projecte precarregat
- Molt útils per realitzar tasques en background
  - Carrega de dades
  - Tasques de cron
  - Procés de blocs de dades
  - ...

https://docs.djangoproject.com/en/dev/howto/custom-management-commands/

5.2 – Exemple d'ús

```
# -*- coding: utf-8 -*-
from django.core.management.base import BaseCommand, CommandError

class Command(BaseCommand):
    args = '<arg1>,<arg2>'
    help = 'A help message'

    def handle(self, *args, **options):
        for em in ExampleModel.objects.all():
            em.change_date = datetime.datetime.utcnow()
            em.save()
```

```
python manage.py mycommand
```

6 – Més conceptes?

- Context processors
  - Processadors de context, que porten càlculs i noves variables a l'entorn del context de template
- Widgets pròpis
  - Personalització de widgets per als formularis
- Wizard forms
  - Formulariz en format wizard
- Framework de gelocalització
- Framework de missatges

6 – Més conceptes?

- Django Resources

  https://code.djangoproject.com/wiki/DjangoResources
- Altres

  https://www.djangopackages.com
- CMS

  https://www.django-cms.org/en/

7 – Testing

- Django està molt orientat al testing
- Cada app auto-genera un mòdul específic per el test
- Es basen en una suite estàndar de Python

  https://docs.djangoproject.com/en/dev/topics/testing/overview/

8 – Projecte 3: Interconexió

- Aquest projecte tracta sobre la interconexió dels dos projectes anteriors
- Es generarán habilitats avançades utilitzant algunes de les eines que hem descrit en aquesta sessió
- El projecte consisteix, per tant, d'una sèrie de millores i ampliacions del projecte Blog i el projecte Diari
- Per facilitar la feina, proporcionem els dos anteriors projectes complerts

  https://mega.co.nz/#!NVQ2zbAJ!TNl34FN5MSK9GoGVXRtB7T1t5lcjBESsqNMNPzG5zjQ

  https://mega.co.nz/#!ZBJm3bRb!PJmFuhfV1t9DpFNRTS80PBW2vB0ZlJwdSqf6vnuDXEs

8 – Projecte 3: Interconexió

- Funcionalitats
  - Creació de managers per tots els models existents
  - Creació d'un feed RSS per el blog
  - Generació de cache per el detall de blog (només el contingut)
  - Generació de cache de noticies senceres
  - Middleware per capturar les peticions web: es farà un recompte de les URIs visitades
  - Es conectarà la generació de noves entrades al blog, creant una notícia al Diari, mitjançant signals

8 – Projecte 3: Interconexió

- Funcionalitats
  - Creació d'un Command per gestionar l'arxivat en el diari: les noticies amb més de 8 setmanes de vida, s'arxivaran en un nou model NoticiaAntiga, que ja no sortirà a la web
- Ampliació (opcional):
  - Creació de tests per comprovar que les principals vistes funcionen
  - Integrar django-cms en el blog
  - Generar subscripció al diari

Django

Gràcies i fins aviat!

Barcelonactiva

Ajuntament de Barcelona

bcn.cat/barcelonactiva
bcn.cat/cibernarium