FIELD NOTE / DJANGO

Menggunakan Legacy Database dalam Django

Petruknisme · 10 Jun, 2017 · 2 min to read

- [Panduan Memulai Belajar Django][1]
    - [Perbedaan App dan Project pada Django][2] {.entry-title}

Pada kesempatan sebelumnya saya sudah membahas tentang seluk beluk Django sebagai perkenalan untuk proses belajar di [Sinau Development][3] dan buku yang saya sedang kerjakan [Panduan Memulai Belajar Django][4]. Materi tersebut masih belum berlanjut dikarenakan beberapa hambatan terutama masalah waktu. Kita akan coba untuk melompat materi mengenai Database untuk Django.

Dalam Django, kita bisa melakukan perancangan database dengan menggunakan models object atau pun langsung. Apa yang akan saya bahas kali ini adalah point kedua, dimana kita akan menggunakan existing database sebagai models untuk nantinya django bertransaksi data. Tentunya kita harus membuat django project terlebih dahulu:

```
$ django-admin startproject latihan
```

Bila sudah terbuat project maka akan seperti berikut:

Project structure

```
$ tree latihan
latihan
|-- latihan
|   |-- __init__.py
|   |-- settings.py
|   |-- urls.py
|   `-- wsgi.py
`-- manage.py

1 directory, 5 files
```

Kita akan memakai default database scheme dari Django itu sendiri, tapi bisa juga menggunakan database yang lain. Agar lebih memudahkan dalam tulisan kali ini, maka akan menggunakan default

Licensed under CC BY-SA 4.0

database scheme dari django dan menambahkan beberapa table tambahan. Masuk kedalam direktori lalu lakukan migrasi database:

```
cd latihan && python manage.py migrate
```

Maka akan menampilkan hasil seperti berikut:

```
$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying sessions.0001_initial... OK
```

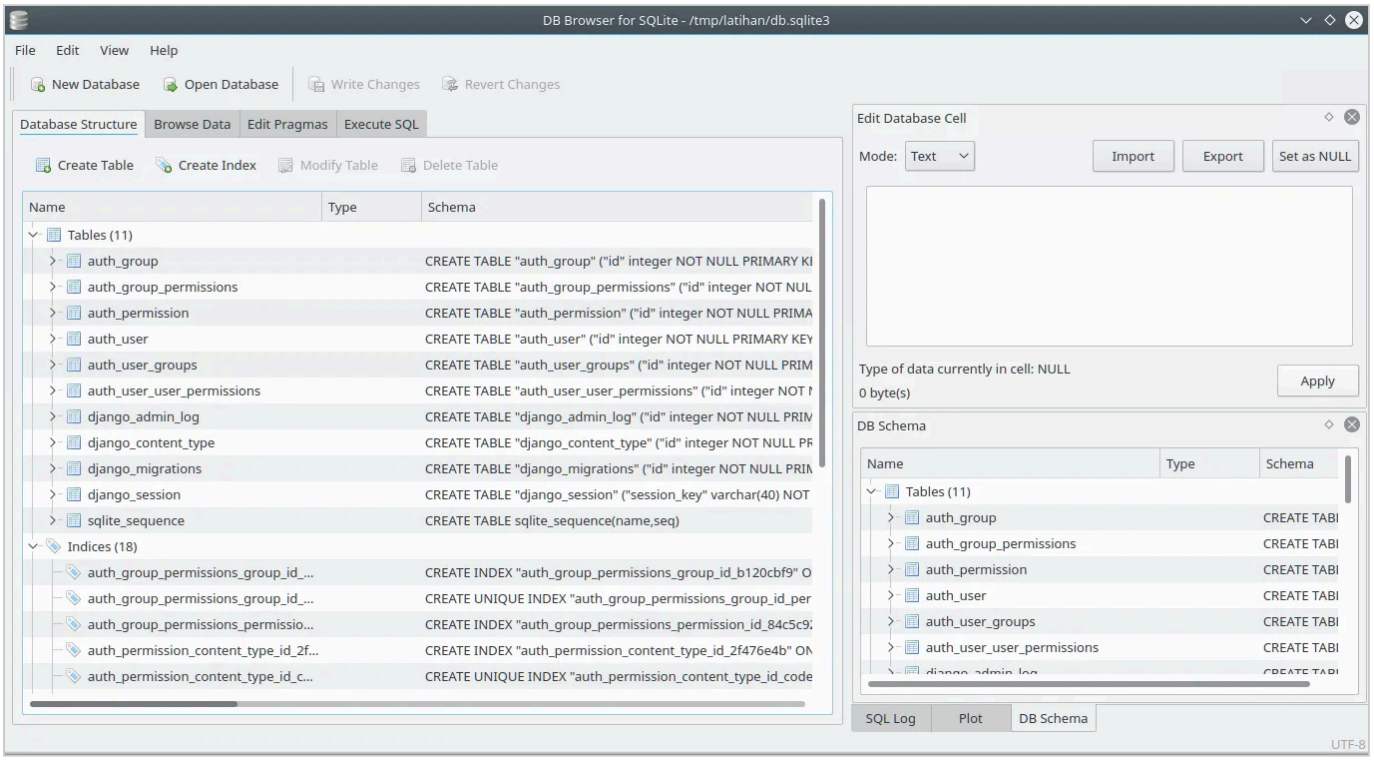
Dengan sendiri tercipta sebuah berkas database yang menggunakan SQLite3

```
$ tree latihan/
latihan/
|-- db.sqlite3
|-- latihan
|   |-- __init__.py
|   |-- __init__.pyc
|   |-- settings.py
```

Licensed under CC BY-SA 4.0

```
|   |-- settings.pyc
|   |-- urls.py
|   |-- urls.pyc
|   `-- wsgi.py
`-- manage.py
```

Saya menggunakan DB Browser for SQLite [http://sqlitebrowser.org/ (http://sqlitebrowser.org/)] untuk manage database SQLite3. Dan bila kita membuka database yang sudah terbuat maka struktur database nya akan terlihat seperti berikut:



Sebagai contoh table, kita tambahkan table dengan Query ataupun dengan import SQL file bila ada seperti berikut:

```
CREATE TABLE `latihan` (
  `id`  INTEGER PRIMARY KEY AUTOINCREMENT,
  `nama`  TEXT,
  `alamat`  TEXT
);
```

Jangan lupa disave yak. Write Changes(ctrl + s)

Ok, bila sudah berhasil maka akan muncul 1 table baru. Oh iya kalau database nya berbeda, harus diatur terlebih dahulu di file settings.py bagian:

settings.py

```
# Database
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Mulai membuat models database kita dengan menggunakan fitur *inspectdb* di django:

```
$ python manage.py inspectdb > latihan/models.py
```

Nah, kalau ga ada error, nanti akan terbuat file baru bernama models.py di direktori latihan dengan isi sebagai berikut:

models.py

```
# This is an auto-generated Django model module.
# You'll have to do the following manually to clean this up:
#   * Rearrange models' order
#   * Make sure each model has one field with primary_key=True
#   * Make sure each ForeignKey has `on_delete` set to the desired behavi
#   * Remove `managed = False` lines if you wish to allow Django to creat
# Feel free to rename the models, but don't rename db_table values or fie
from __future__ import unicode_literals

from django.db import models


class AuthGroup(models.Model):
    id = models.IntegerField(primary_key=True)  # AutoField?
```

Licensed under CC BY-SA 4.0

```
15      name = models.CharField(unique=True, max_length=80)
16
17      class Meta:
18          managed = False
19          db_table = 'auth_group'
20
21
22  class AuthGroupPermissions(models.Model):
23      id = models.IntegerField(primary_key=True)  # AutoField?
24      group = models.ForeignKey(AuthGroup, models.DO_NOTHING)
25      permission = models.ForeignKey('AuthPermission', models.DO_NOTHING)
26
27      class Meta:
28          managed = False
29          db_table = 'auth_group_permissions'
30          unique_together = (('group', 'permission'),)
31
32
33  class AuthPermission(models.Model):
34      id = models.IntegerField(primary_key=True)  # AutoField?
35      content_type = models.ForeignKey('DjangoContentType', models.DO_NOTHI
36      codename = models.CharField(max_length=100)
37      name = models.CharField(max_length=255)
38
39      class Meta:
40          managed = False
41          db_table = 'auth_permission'
42          unique_together = (('content_type', 'codename'),)
43
44
45  class AuthUser(models.Model):
46      id = models.IntegerField(primary_key=True)  # AutoField?
47      password = models.CharField(max_length=128)
48      last_login = models.DateTimeField(blank=True, null=True)
49      is_superuser = models.BooleanField()
50      first_name = models.CharField(max_length=30)
51      last_name = models.CharField(max_length=30)
52      email = models.CharField(max_length=254)
53      is_staff = models.BooleanField()
54      is_active = models.BooleanField()
55      date_joined = models.DateTimeField()
56      username = models.CharField(unique=True, max_length=150)
```

Licensed under CC BY-SA 4.0

```
57
58          class Meta:
59              managed = False
60              db_table = 'auth_user'
61
62
63      class AuthUserGroups(models.Model):
64          id = models.IntegerField(primary_key=True)  # AutoField?
65          user = models.ForeignKey(AuthUser, models.DO_NOTHING)
66          group = models.ForeignKey(AuthGroup, models.DO_NOTHING)
67
68          class Meta:
69              managed = False
70              db_table = 'auth_user_groups'
71              unique_together = (('user', 'group'),)
72
73
74      class AuthUserUserPermissions(models.Model):
75          id = models.IntegerField(primary_key=True)  # AutoField?
76          user = models.ForeignKey(AuthUser, models.DO_NOTHING)
77          permission = models.ForeignKey(AuthPermission, models.DO_NOTHING)
78
79          class Meta:
80              managed = False
81              db_table = 'auth_user_user_permissions'
82              unique_together = (('user', 'permission'),)
83
84
85      class DjangoAdminLog(models.Model):
86          id = models.IntegerField(primary_key=True)  # AutoField?
87          object_id = models.TextField(blank=True, null=True)
88          object_repr = models.CharField(max_length=200)
89          action_flag = models.PositiveSmallIntegerField()
90          change_message = models.TextField()
91          content_type = models.ForeignKey('DjangoContentType', models.DO_NOTHI
92          user = models.ForeignKey(AuthUser, models.DO_NOTHING)
93          action_time = models.DateTimeField()
94
95          class Meta:
96              managed = False
97              db_table = 'django_admin_log'
98
```

Licensed under CC BY-SA 4.0

```
 99
100  class DjangoContentType(models.Model):
101      id = models.IntegerField(primary_key=True)  # AutoField?
102      app_label = models.CharField(max_length=100)
103      model = models.CharField(max_length=100)
104
105      class Meta:
106          managed = False
107          db_table = 'django_content_type'
108          unique_together = (('app_label', 'model'),)
109
110
111  class DjangoMigrations(models.Model):
112      id = models.IntegerField(primary_key=True)  # AutoField?
113      app = models.CharField(max_length=255)
114      name = models.CharField(max_length=255)
115      applied = models.DateTimeField()
116
117      class Meta:
118          managed = False
119          db_table = 'django_migrations'
120
121
122  class DjangoSession(models.Model):
123      session_key = models.CharField(primary_key=True, max_length=40)
124      session_data = models.TextField()
125      expire_date = models.DateTimeField()
126
127      class Meta:
128          managed = False
129          db_table = 'django_session'
130
131
132  class Latihan(models.Model):
133      id = models.IntegerField(blank=True, null=True)
134      nama = models.TextField(blank=True, null=True)
135      alamat = models.TextField(blank=True, null=True)
136
137      class Meta:
138          managed = False
139          db_table = 'latihan'
```

Licensed under CC BY-SA 4.0

Karena **id** merupakan *primary_key*, maka kita perlu merubahnya menjadi:

```
id = models.IntegerField(primary_key=True)
```

Model yang kita inginkan sudah terbuat. Kita akan coba pakai model tersebut untuk menampilkan data dari database, sebelumnya tambahkan beberapa data dummy di table latihan yang sudah terbuat.

```
INSERT INTO `latihan`(`id`,`nama`,`alamat`) VALUES (NULL,'Budi','Jakarta')
INSERT INTO `latihan`(`id`,`nama`,`alamat`) VALUES (NULL,'Bella','Bandung'
INSERT INTO `latihan`(`id`,`nama`,`alamat`) VALUES (NULL,'Bena','Bogor');
```

Buat sebuah file baru bernama *views.py* didirektori yang sama dengan *models.py*:

views.py

```
from django.http import HttpResponse, HttpResponseRedirect, Http404
from django.shortcuts import render_to_response, get_object_or_404

from models import Latihan

def views_latihan(request):
    latihan_list = Latihan.objects.all().order_by('id')
    return render_to_response('index.html', {'latihan_data': latihan_list}
```

Kita perlu membuat templates terlebih dahulu, buat direktori `templates` dengan file `index.html` seperti berikut:

index.html

```
{% for data in latihan_data %}
      {{ data.id }}
      {{ data.nama }}
      {{ data.alamat }}
  <hr/>
{% endfor %}
```

Licensed under CC BY-SA 4.0

Tahap selanjutnya kita buat url routing di urls.py agar views dan models yang kita buat bisa dipanggil lewat URL yang diinginkan. Ubah file urls.py menjadi seperti berikut:

urls.py

```
from django.conf.urls import url
from django.contrib import admin

from views import views_latihan

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', views_latihan),
]
```

Daftarkan Latihan sebagai installed app didalam *settings.py*:

settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'latihan',
]
```

Bila semua sudah selesai, jalankan django webserver:

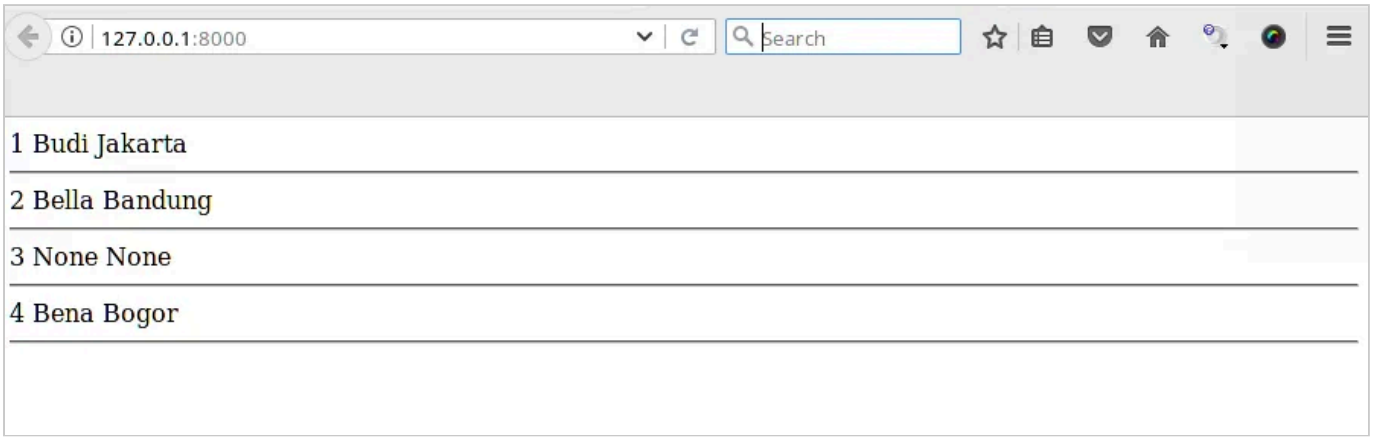
```
$ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
June 10, 2017 - 06:06:46
Django version 1.11.2, using settings 'latihan.settings'
Starting development server at http://127.0.0.1:8000/
```

Licensed under CC BY-SA 4.0

```
8 | Quit the server with CONTROL-C.
```

Tadaaaaa, models dari legacy database kita sudah berhasil terbuat:



Mungkin itu saja yang bisa saya bahas pada tulisan kali ini. Semoga artikel ini bermanfaat. Koreksi bila ada kesalahan

- [1] https://petruknisme.com/2016/11/30/panduan-memulai-belajar-django/ (https://petruknisme.com/2016/11/30/panduan-memulai-belajar-django/)
- [2] https://petruknisme.com/2016/12/01/perbedaan-app-dan-project-pada-django/ (https://petruknisme.com/2016/12/01/perbedaan-app-dan-project-pada-django/)
- [3] https://t.me/sinaudev (https://t.me/sinaudev)
- [4] https://www.gitbook.com/book/aancw/panduan-memulai-belajar-django/details (https://www.gitbook.com/book/aancw/panduan-memulai-belajar-django/details)

Django # Python # Tutorial # Web

Licensed under CC BY-SA 4.0