Django – Part III

2015-05-27

SPARCS 11 undead

Greatly Inspired by SPARCS 10 hodduc

Previously on Django Seminar (Part II)

Yes, it’s been a while!

- Writing Templates
  - Handling HTML Files with Django
  - Extending Templates
  - Template Tags
  - Template Filters

Previously on Django Seminar (Part II)

Yes, it’s been a while!

- Writing Templates
  - Handling HTML Files with Django
  - Extending Templates
  - Template Tags
  - Template Filters
- Two practices: Introduction Relay and Hello, Ahaes!

Previously on Django Seminar (Part I)

- Structure of Web Environment
  - HTTP Requests and HTTP Responses
- Structure of a Django Project
  - Projects and applications
- How to Use Python Virtual Environments
- Simple views.py
- Urls.py
  - URL Routing
  - Dynamic URL (Regular Expression)

Today’s Topic

- The MVC Design Pattern

- Models
    - Handling Structured Data
    - Using Database

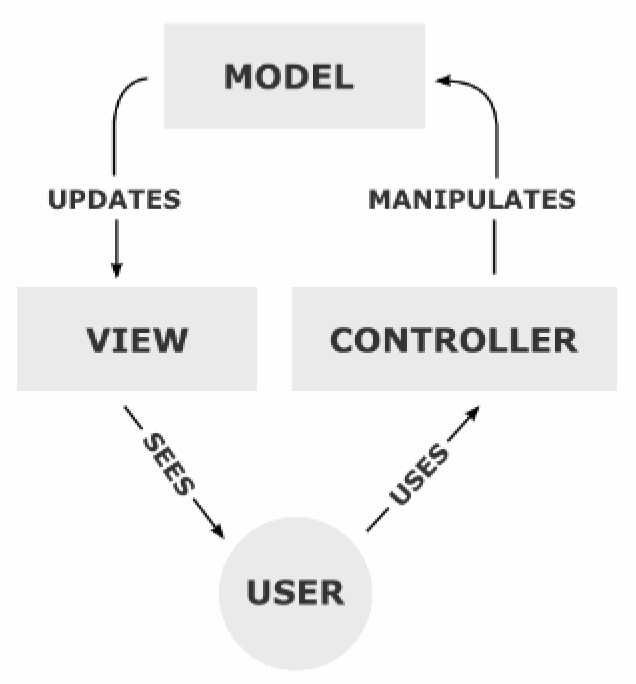
The MVC Design Pattern

And Its Applications on Web and Django

The MVC Design Pattern

The Model-View-Controller Design Pattern

- 사용자는 Controller를 조작
- Controller는 Model을 통해 데이터에 접근
- 필요한 사항을 업데이트하여 View에 전달
- View를 통해 조작에 대한 피드백을 최종전달



The MVC and Web Application

Scenario

1. User가 웹사이트에 접속
2. Controller가 요청된 페이지를 렌더링하기 위해 Model을 참조
3. Model은 Database에서 적절한 데이터를 참조하여 결과 반환
4. Controller는 Model이 반환한 결과를 바탕으로 View를 반영
5. 데이터가 반영된 View가 최종적으로 User에게 보여짐

Django and MVC

- **Controller**
  - views.py에서 작성한 view 함수들
  - 사용자는 urls.py에 명시된 URL을 통해 사용
- **View**
  - HTML로 작성된 template 파일들
  - CSS, JavaScript, media (images)
- **Model**
  - models.py에서 정의된 데이터 구조들
  - settings.py에 명시된 Database
  - Today’s topic!

Creating Models

Managing Structured Data on Django

Writing Introduction Model

```
~/ $ source env/bin/activate
(env) ~/tutorial $ vi intro/models.py
```

```
from django.db import models

class Person(models.Model):
    name = models.CharField(max_length=10)
    age = models.IntegerField(default=99)
    occupation = models.CharField(max_length=20)
    city = models.CharField(max_length=10)
    town = models.CharField(max_length=10)
```

The Person DB Schema

name	age	occupation	city	town
이창원	24	무직	서울	목동
홍석주	21	학생	대전	관저동
김정민	22	와우저	전라남도	광양
…	…	…	…	…

Django Model Field Reference

- AutoField
- BigIntegerField
- BooleanField
- CharField
- DateTimeField
- EmailField
- FileField
- ImageField

…

Refer to Django Official Documentation for more information
(Version 1.8: https://docs.djangoproject.com/en/1.8/ref/models/fields/)

Adding Intro App to Settings

```
(env) ~/tutorial $ vi tutorial/settings.py
```

```
…
INSTALLED_APPS = (
    …
    ‘helloworld’,
    ‘intro’,
)
…
```

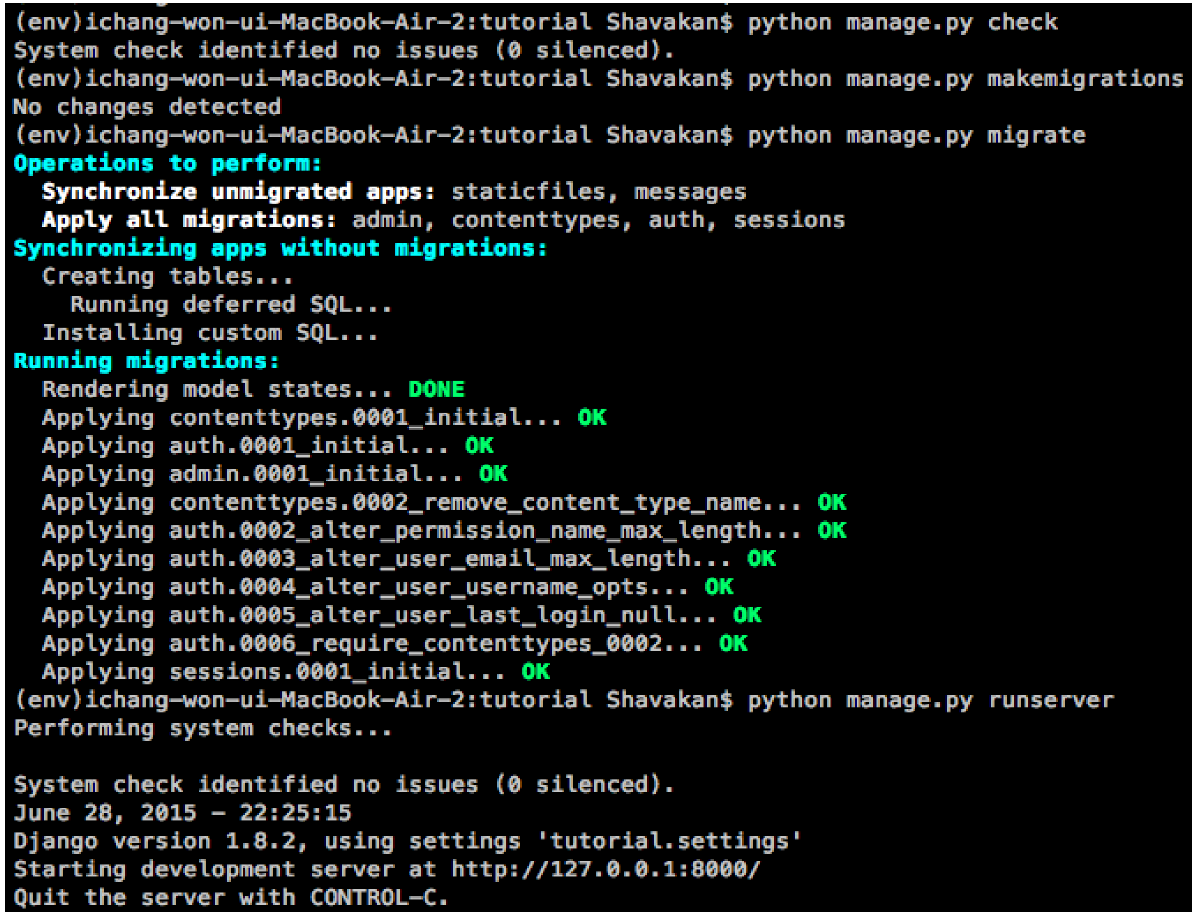
Applying Models to Django

```
# Check for syntax errors
(env) ~/tutorial $ python manage.py check

# Check for schema changes
(env) ~/tutorial $ python manage.py makemigrations

# Apply new schema
(env) ~/tutorial $ python manage.py migrate
```

Applying Models to Django



```
(env)ichang-won-ui-MacBook-Air-2:tutorial Shavakan$ python manage.py check
System check identified no issues (0 silenced).
(env)ichang-won-ui-MacBook-Air-2:tutorial Shavakan$ python manage.py makemigrations
No changes detected
(env)ichang-won-ui-MacBook-Air-2:tutorial Shavakan$ python manage.py migrate
Operations to perform:
  Synchronize unmigrated apps: staticfiles, messages
  Apply all migrations: admin, contenttypes, auth, sessions
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
  Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... 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 sessions.0001_initial... OK
(env)ichang-won-ui-MacBook-Air-2:tutorial Shavakan$ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
June 28, 2015 - 22:25:15
Django version 1.8.2, using settings 'tutorial.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Creating Objects

```
~/tutorial $ python manage.py shell
>>> from intro.models import Person
>>> p1 = Person(name=‘ChangWon Lee’, age=24,
occupation=‘Idle’, city=‘Seoul’, town=‘Mokdong’)
>>> p1.save()


# Add at least 5 people around you
```

Retrieving All Objects

```
>>> person_list = Person.objects.all()
>>> len(person_list)
5
>>> shavakan = person_list[0]
>>> shavakan.name
u’ChangWon Lee’
>>> shavakan.occupation
u’Idle’
```

Retrieving Objects with Filters

```
>>> person_list = Person.objects.filter(city=‘Seoul’)
>>> len(person_list)
1
>>> person_list[0].name
u’ChangWon Lee’
```

Saving Changes to Objects

```
>>> person_list = Person.objects.all()
>>> len(person_list)
5
>>> shavakan = person_list[0]
>>> shavakan.age += 1
>>> shavakan.age
25
>>> shavakan.save()
```

Removing Objects

```
>>> person_list = Person.objects.all()
>>> len(person_list)
5
>>> shavakan = person_list[0]
>>> shavakan.delete()
>>> len(person_list)
4
```

Changing Database Setting

```
(env) ~/tutorial $ vi tutorial/settings.py
```

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

Changing Database Setting

```
(env) ~/tutorial $ vi tutorial/settings.py
```

```
…
DATABASES = {
    ‘mysql’: {
        ‘ENGINE’: ‘django.db.backends.mysql’,
        ‘NAME’: ‘user_data’,
        ‘USER’: ‘mysql_user’,
        ‘PASSWORD’: ‘priv4te’,
    }
}
…
```

Django Admin

Simple way to manage model objects

Django Admin

```
(env) ~/tutorial $ vi tutorial/settings.py
```

```
…
INSTALLED_APPS = (
    …
    ‘django.contrib.admin’,
    …
)
…
```

Django Admin

```
(env) ~/tutorial $ vi tutorial/urls.py
```

```
from django.contrib import admin



…
urlpatterns = [
    …
    url(r’^admin/’, include(admin.site.urls’)),
    …
]
…
```

Django Admin

```
(env) ~/tutorial $ vi intro/admin.py
```

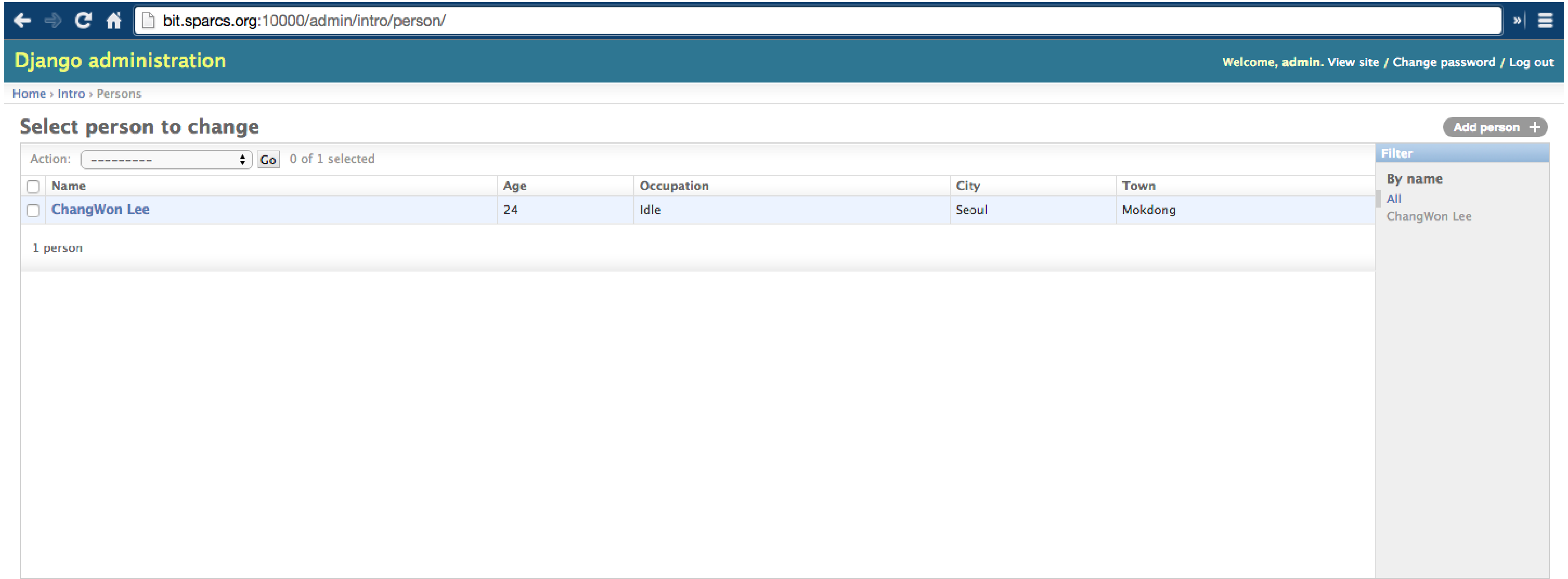
```
from django.contrib import admin
from intro.models import Person


class PersonAdmin(admin.ModelAdmin):
    list_display = ('name', 'age', 'occupation', 'city', 'town')
    list_filter = ('name',)


admin.site.register(Person, PersonAdmin)
```

Django Admin

```
(env) ~/tutorial $ python manage.py createsuperuser
(env) ~/tutorial $ python manage.py runserver 0.0.0.0:10000
```



bit.sparcs.org:10000/admin/intro/person/

Django administration

Welcome, admin. View site / Change password / Log out

Home › Intro › Persons

Select person to change

Action: ---------- Go 0 of 1 selected

□	Name	Age	Occupation	City	Town
□	ChangWon Lee	24	Idle	Seoul	Mokdong

1 person

Filter

By name

All

ChangWon Lee

Today's Practice: Introduction Model

- Introduction Relay를 수정하자
  - /intro/<strA>/<strB>/<strC>/
  - /intro/<strA>/<strB>/<strC>/<strD>/<strE>/
  - URL Parameter를 이용하여 Person Model을 저장하여 보자.

  - /intro/
  - 나는 서울시 목동의 이창원이다! (24세, 무직)
  - Person Model에 저장된 모든 Object를 보여주자.
  - views.py에서는 Object 데이터만 보내고,
    출력 형식은 templates에서 지정하여야 한다.

Today’s Practice: Introduction Model

- Introduction Relay를 수정하자
  - /intro/<strA>/<strB>/<strC>/
  - /intro/<strA>/<strB>/<strC>/<strD>/<strE>/
  - URL Parameter를 이용하여 Person Model을 저장하여 보자.

  - /intro/
  - 나는 서울시 목동의 이창원이다! (24세, 무직)
  - Person Model에 저장된 모든 Object를 보여주자.
  - views.py에서는 Object 데이터만 보내고,
    출력 형식은 templates에서 지정하여야 한다.
    - `{‘people’: Person.objects.all()}`