



통합 구현(Spring, Django)

Django 모델 파악하기



한국기술교육대학교
온라인평생교육원

학습내용

- 테이블 정의
- 모델 만들기
- 모델의 활성화

학습목표

- Django에서 테이블을 정의하고 필드를 설정하는 방법을 파악할 수 있다.
- Django에서 모델 객체를 구성하는 방법에 대해 설명할 수 있다.
- Django에서 모델 객체를 데이터베이스와 연동하는 방법에 대해 설명할 수 있다.

테이블 정의

1 설문조사 어플리케이션

1 설문조사 어플리케이션 구성

사람들이 설문 내용을 보고 직접 투표할 수 있는 개방된 사이트

관리자가 설문을 추가, 변경, 삭제할 수 있는 관리용 사이트

설문조사 질문1?
설문조사 질문2?
설문조사 질문3?

1. 초기화면

설문조사 질문1?

1. 선택지
2. 선택지
3. 선택지

선택

2. 설문조사 항목 하나를
선택했을 때 화면
(질문에 대한 선택지 선택)

설문조사 질문1?

1. 선택지 - 5건
2. 선택지 - 2건
3. 선택지 - 3건

3. 2에서 선택지
선택 후 결과 보기
화면

2 필요 모델

설문조사 질문을 저장할 모델

- 질문 문장
- 질문을 생성한 시간

설문조사 선택지를 저장할 모델

- 선택지 항목 문구
- 선택지 선택 횟수

테이블 정의

2 테이블과 필드 구성

1 Question 테이블

컬럼명	타입	제약조건	설명
id	integer	NotNull, PK, Auto Increment	Primary Key
question_text	varchar(200)	NotNull	질문 문장
pub_date	datetime	NotNull	질문을 생성한 시간

설문조사 질문들을 저장할 테이블

2 Choice 테이블

컬럼명	타입	제약조건	설명
id	integer	NotNull, PK, Auto Increment	Primary Key
question_id	integer	NotNull, FK(Question.id), Index	Foreign Key
choice_text	varchar(200)	NotNull	선택지 항목 문구
votes	integer	NotNull	선택지 선택 횟수

설문조사 선택지를 저장할 테이블

모델 만들기

1 테이블 컬럼과 변수형 관계

1 테이블 컬럼과 변수형 맵핑

Django에서는 기본적으로 SQLite3를 지원하며, settings.py의 수정을 통해 다른 데이터베이스도 이용 가능

기본적으로 Django.db.models.model 클래스에 미리 정의되어 있는 필드 클래스에 따름

컬럼	필드 타입	설명
integer	IntegerField(**options)	기본값을 지정 가능함
varchar	CharField(**options)	최대 길이를 지정 가능함
datetime	DateTimeField(**options)	라벨 문구를 지정 가능함
Boolean	BooleanField(**options)	기본값을 지정 가능함
text	TextField(**options)	최대 길이를 지정 가능함

컬럼 테이블

2 설문조사 테이블 맵핑

컬럼명	컬럼 타입	필드 타입
id	integer	PK는 자동 생성해줌
question_text	varchar(200)	models.CharField(max_length=200)
pub_date	datetime	models.DateTimeField('date published')

Question 테이블

모델 만들기

1 테이블 컬럼과 변수형 관계

2 설문조사 테이블 맵핑

컬럼명	컬럼 타입	필드 타입
id	integer	PK는 자동 생성해줌
question_id	integer	models.ForeignKey(Question)
choice_text	varchar(200)	models.CharField(max_length=200)
votes	integer	models.IntegerField(default=0)

Choice 테이블

2 모델 클래스 작성

1 모델의 정의

1 모델은 models.py에 정의

2 하나의 모델은 하나의 클래스에 정의

3 각 모델은 django.db.models.Model이라는 클래스의 서브클래스로 표현됨

모델 만들기

2 모델 클래스 작성

2 모델 클래스 작성

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

mysite/polls/models.py

PK(Primary Key)

- 클래스에 지정해주지 않아도 Django에서 자동으로 NotNull, AutoIncrement 속성으로 작성해 줌
- 이 때, 이름은 테이블 명의 소문자를 접두어로 해서 생성되며, Question 테이블의 PK 클래스 변수는 question_id

FK(Foreign Key)

- 항상 다른 테이블의 PK에 연결되며, 해당 테이블의 클래스를 지정해줌
- on_delete=models.CASCADE 옵션으로 인해 참조하고 있는 Question 테이블의 레코드가 삭제되면 Choice 테이블의 레코드도 삭제됨

모델의 활성화

1 테이블 생성

1 모델 설정 후 해야 할 일

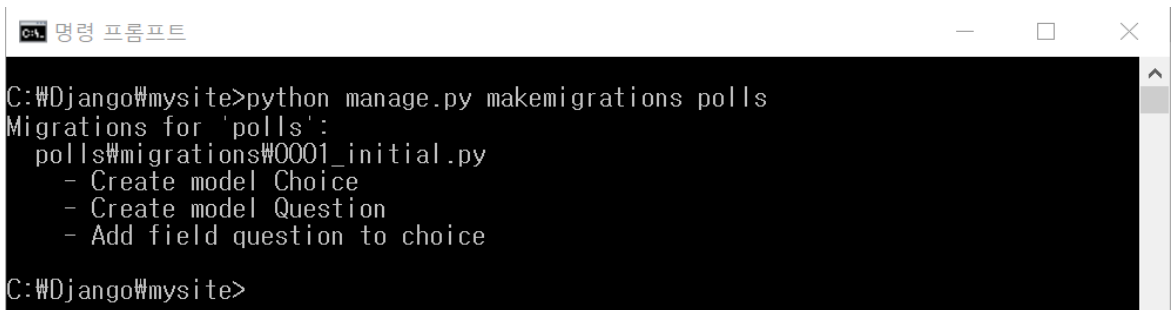
어플리케이션에 대한 데이터베이스
Scheme 생성

Question과 Choice 객체에
접근하기 위한 Python 데이터베이스
접근 API를 생성

2 데이터베이스 Scheme 생성

1 프로젝트 폴더에서 'python manage.py makemigrations 어플리케이션명' 입력

- 모델 변경(혹은 새로운 모델 생성) 및 변경사항을 저장하고 싶다는 것을 알림



```

C:\WDjango\mysite>python manage.py makemigrations polls
Migrations for 'polls':
  polls\migrations\0001_initial.py
    - Create model Choice
    - Create model Question
    - Add field question to choice
C:\WDjango\mysite>
  
```

모델 변경 화면

2 mysite/polls/migrations/0001_initial.py 생성 확인

```

from __future__ import unicode_literals
from django.db import migrations, models
import django.db.models.deletion

class Migration(migrations.Migration):
    initial = True
    dependencies = [
    ]
  
```


모델의 활성화

1 테이블 생성

2 데이터베이스 Scheme 생성

```
operations = [
    migrations.CreateModel(
        name='Choice',
        fields=[
            ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
            ('choice_text', models.CharField(max_length=200)),
            ('votes', models.IntegerField(default=0)),
        ],
    ),
    ...생략...
```

3 프로젝트 폴더에서 'python manage.py migrate' 입력

모든
migration들을
수집하여
이를 실행

모델에서의
변경사항들과
데이터베이스
Scheme의
동기화

앞서 입력한
실제 테이블을
생성하는 행위

```
C:\#Django\mysite>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, polls, sessions
Running migrations:
  Applying polls.0001_initial... OK
C:\#Django\mysite>
```

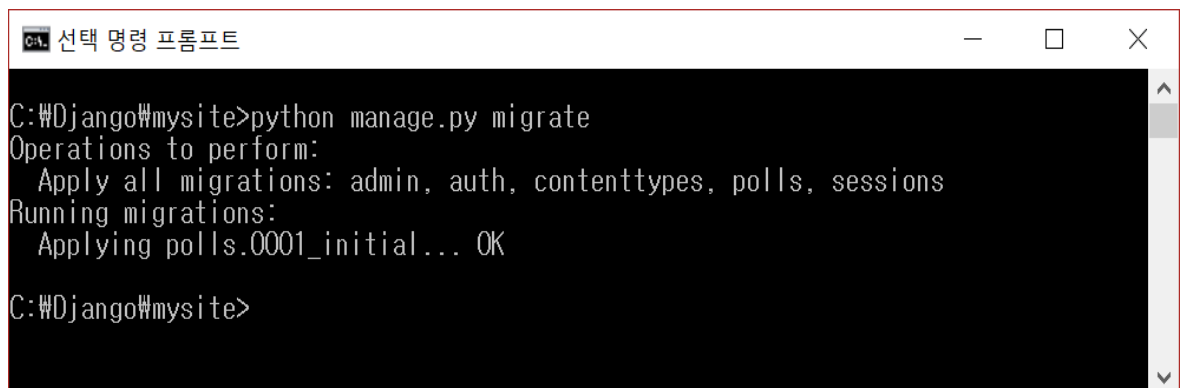
데이터베이스 Scheme 생성 화면

모델의 활성화

1 테이블 생성

2 데이터베이스 Scheme 생성

- 1 models.py에서 모델 변경
- 2 makemigrations 명령을 통해 변경사항에 대해 migration 생성
- 3 migrate 명령을 통해 변경사항을 데이터베이스에 반영



```
C:\Django\mysite>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, polls, sessions
Running migrations:
  Applying polls.0001_initial... OK

C:\Django\mysite>
```

데이터베이스 동기화 화면

모델의 활성화

2 테이블 확인

1 Admin Site에서 모델 등록

Admin Site에 Question, Choice 객체를 관리할 수 있도록 등록

mysite/polls/admin.py를 다음과 같이 편집

```
from django.contrib import admin
from .models import Question
from .models import Choice
```

```
# Register your models here.
admin.site.register(Question)
admin.site.register(Choice)
```

<http://127.0.0.1:8000/admin>에 접속, 로그인 후 등록 확인

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

POLLS

Choices	+ Add	Change
Questions	+ Add	Change

Admin Site 테이블 화면

1. 테이블 정의

- 설문조사 어플리케이션
 - 사람들이 설문 내용을 보고 직접 투표할 수 있는 개방된 사이트
 - 관리자가 설문을 추가, 변경, 삭제할 수 있는 관리용 사이트
- 테이블과 필드 구성
 - Question 테이블 : 설문조사 질문들을 저장할 테이블
 - Choice 테이블 : 설문조사 선택지를 저장할 테이블

2. 모델 만들기

- 테이블 컬럼과 변수형 관계
 - Django에서는 기본적으로 SQLite3를 지원하며, settings.py의 수정을 통해 다른 데이터베이스도 이용할 수 있음
 - 기본적으로 Django.db.models.model 클래스에 미리 정의되어 있는 필드 클래스에 따름
- 모델 클래스 작성
 - 모델은 models.py에 정의
 - 하나의 모델은 하나의 클래스에 정의
 - 각 모델은 django.db.models.Model이라는 클래스의 서브클래스로 표현
 - 모델 클래스는 PK(Primary Key), FK(Foreign Key) 방식으로 작성

3. 모델의 활성화

- 테이블 생성

모델 설정 후 해야 할 일

: 어플리케이션에 대한 데이터베이스 scheme 생성

: Question과 Choice 객체에 접근하기 위한 Python 데이터베이스 접근 API를 생성

- 테이블 확인

- Admin Site에서 모델 등록

: Admin Site 에 Question, Choice 객체를 관리할 수 있도록 등록하여 mysite/polls/admin.py를 편집

- Admin Site에서 확인

: <http://127.0.0.1/admin>에 접속, 로그인 후 등록 확인