Django night – 21 ottobre 2011

django

Introduzione

Valeria Leonardi

@vleonardi

django

The (Python) Web framework for perfectionists with deadlines

Web-Poll Application

App Polls

- **Edit The file polls/views.py to look like this:**

```
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.

def index(request):
    name = 'Your-Name'
    respString = "Hello, %s. You're at the polls index."%(name)
    resp = HttpResponse(respString)
    return resp
```

App Polls

- **Edit The file polls/urls.py to look like this:**

```
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

- **Edit The file mysite/urls.py to look like this:**

```
from django.conf.urls import url, include
from django.contrib import admin
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^polls/', include('polls.urls')),
]
```

App Polls

- **Edit The file mysite/settings.py to look like this:**

```
INSTALLED_APPS = [
    'polls',  ##include the polls app
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

App Polls

- **Edit The file polls/models.py to look like this:**

```
from __future__ import unicode_literals
from django.db import models
# Create your models here.
class Question(models.Model):
    """
    Database table to include questions associated with the app.
    """
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    """
    Database table to include responses to each question.  One question can have
many choices.  Questions are delegated by a foreign key.
    """
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

App Polls

- Run the commands:
  - **python manage.py makemigrations**

```
Migrations for 'polls':
     polls/migrations/0001_initial.py:
          - Create model Choice
          - Create model Question
          - Add field question to choice
```

  - **python manage.py migrate**

```
   operations to perform:
          Apply all migrations: admin, auth, contenttypes, polls, sessions
          Running migrations:
```

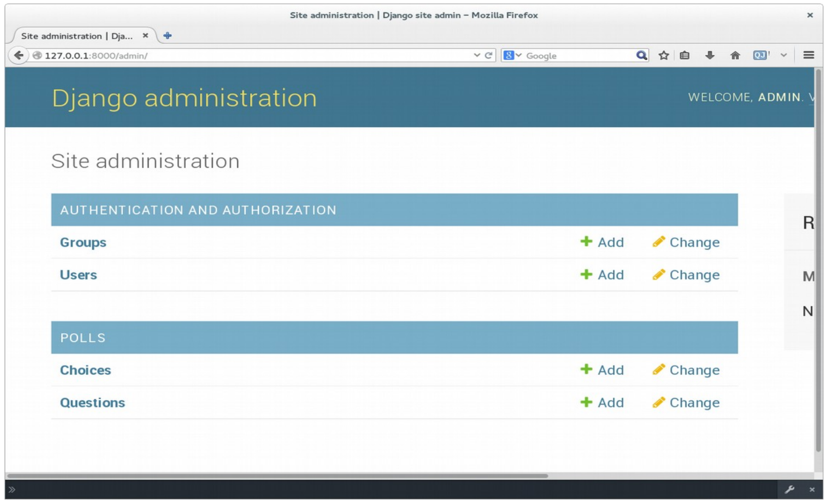
Makemigrations will create sql that will create tables for these models with all of the table attributes and formats. Migrate will then apply the database tables to the db backend through Python.

App Polls

- **Edit The file polls/admin.py to look like this:**

```
from django.contrib import admin
from .models import Question, Choice
admin.site.register(Question)
admin.site.register(Choice)
```

- **Run python manage.py runserver**
- **The admin page now includes admin capabilities for the registered tables**



Site administration | Django site admin – Mozilla Firefox
Site administration | Dja...
127.0.0.1:8000/admin/
Google
Django administration
WELCOME, ADMIN.

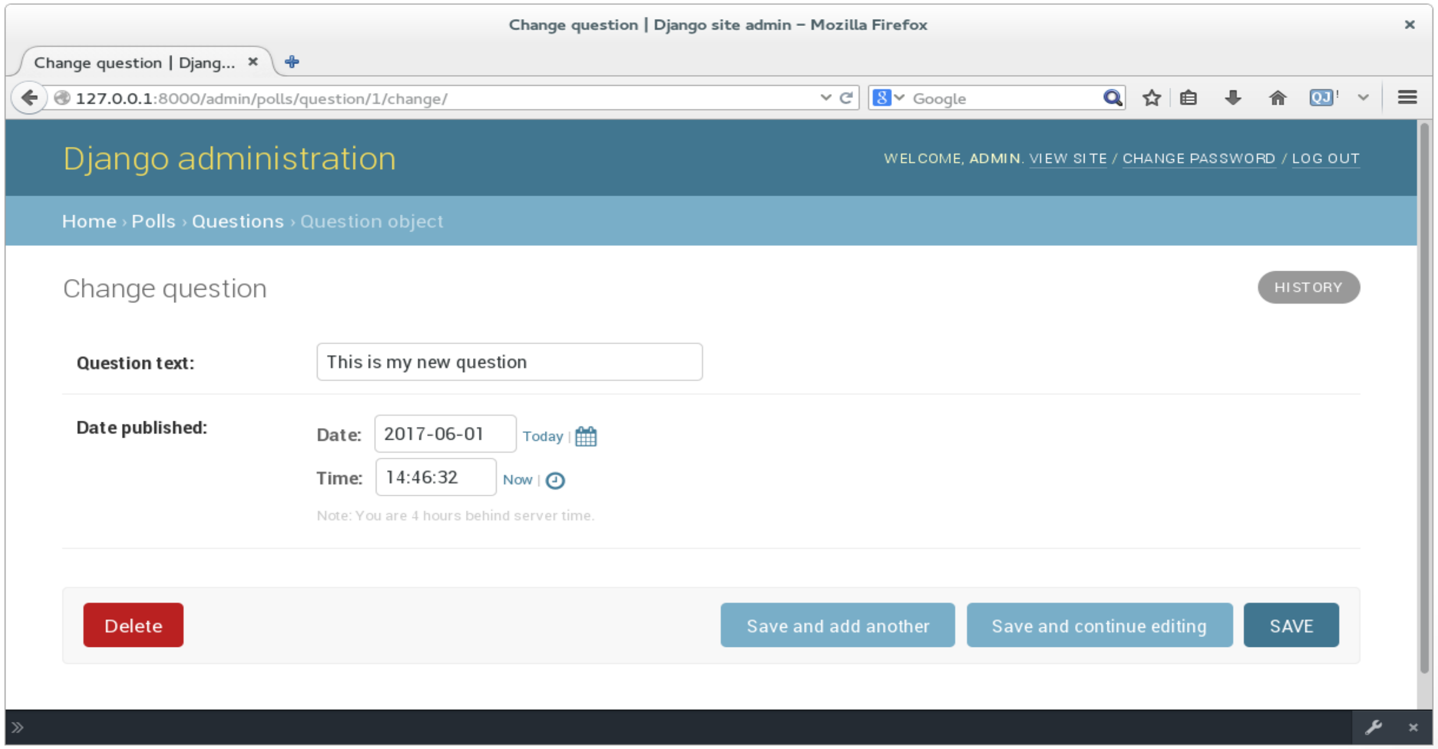
Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	Add	Change
Users	Add	Change

POLLS		
Choices	Add	Change
Questions	Add	Change

App Polls

- **Create, Update, Delete (CRUD) any table entries**


Change question | Django site admin – Mozilla Firefox

127.0.0.1:8000/admin/polls/question/1/change/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Polls › Questions › Question object

Change question

Question text: This is my new question

Date published:

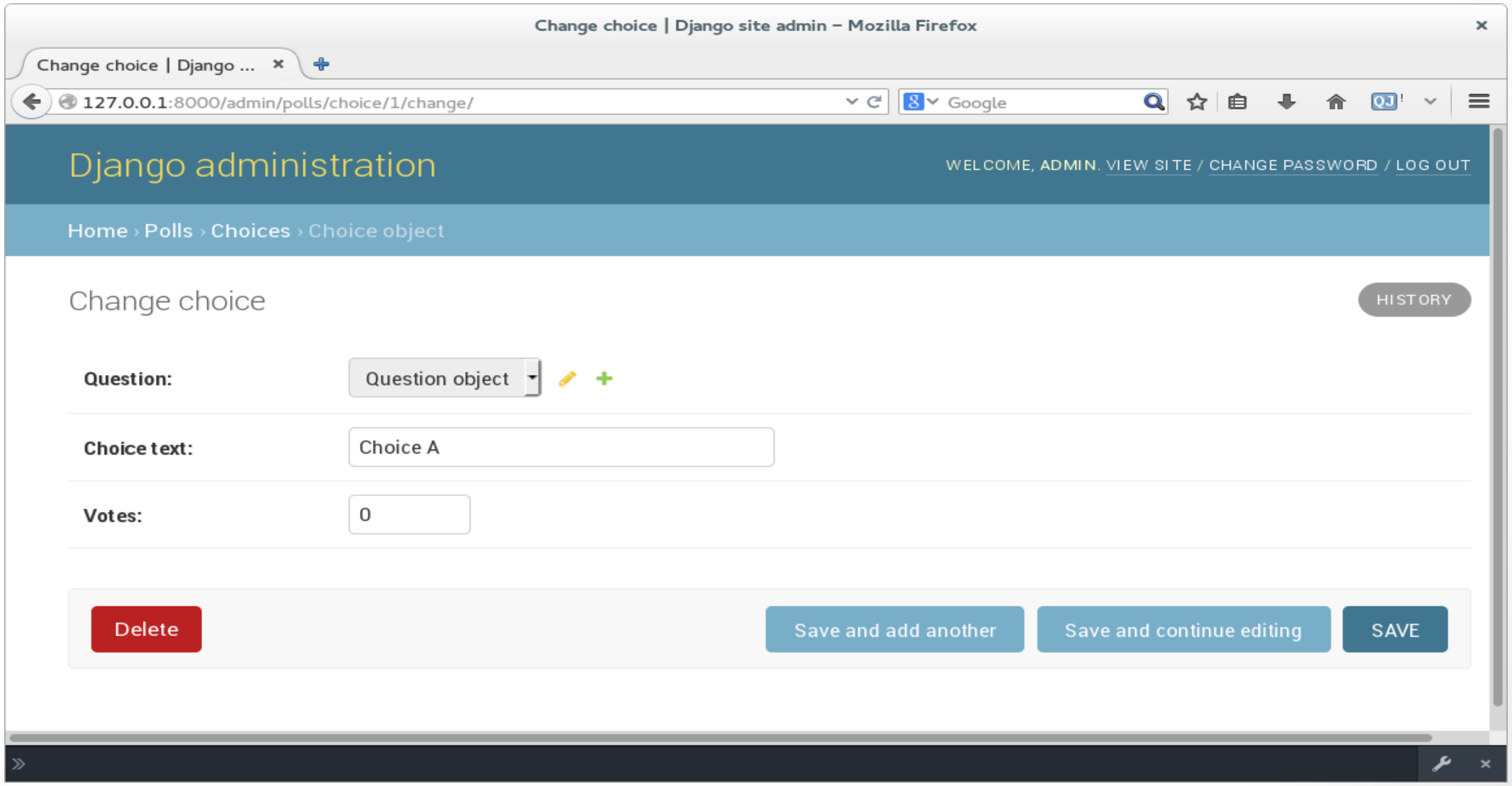
Date: 2017-06-01 Today

Time: 14:46:32 Now

Note: You are 4 hours behind server time.

App Polls

- **Create, Update, Delete (CRUD) any table entries**


Change choice | Django site admin – Mozilla Firefox

127.0.0.1:8000/admin/polls/choice/1/change/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Polls › Choices › Choice object

Change choice

Question:	Question object
Choice text:	Choice A
Votes:	0

App Polls

Edit The file polls/models.py to look like this:

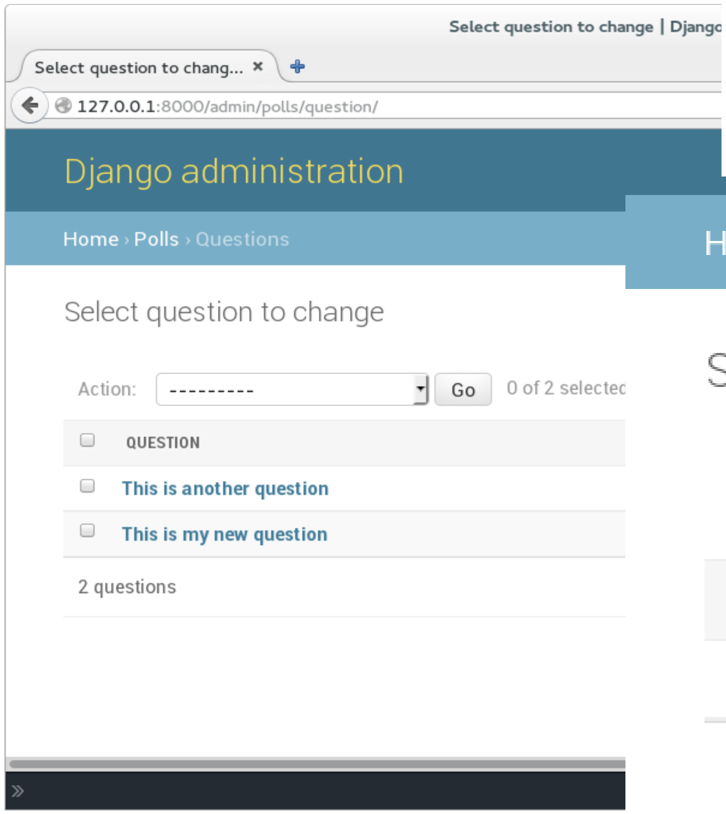
```
from django.contrib import admin
from .models import Question, Choice
admin.site.register(Question)
admin.site.register(Choice)


from __future__ import unicode_literals
from django.db import models
# Create your models here.
class Question(models.Model):
    """
    Database table to include questions associated with the app.
    """
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')
    def __str__(self):
return self.question_text

class Choice(models.Model):
    """
    Database table to include responses to each question.  One question can have many choices.
Questions are delegated by a foreign key.
    """
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
    def __str__(self):
return self.question_text
```

App Polls

Tables now appear with question text and choice text.



Select question to change | Django

Select question to chang... ×

127.0.0.1:8000/admin/polls/question/

Django administration

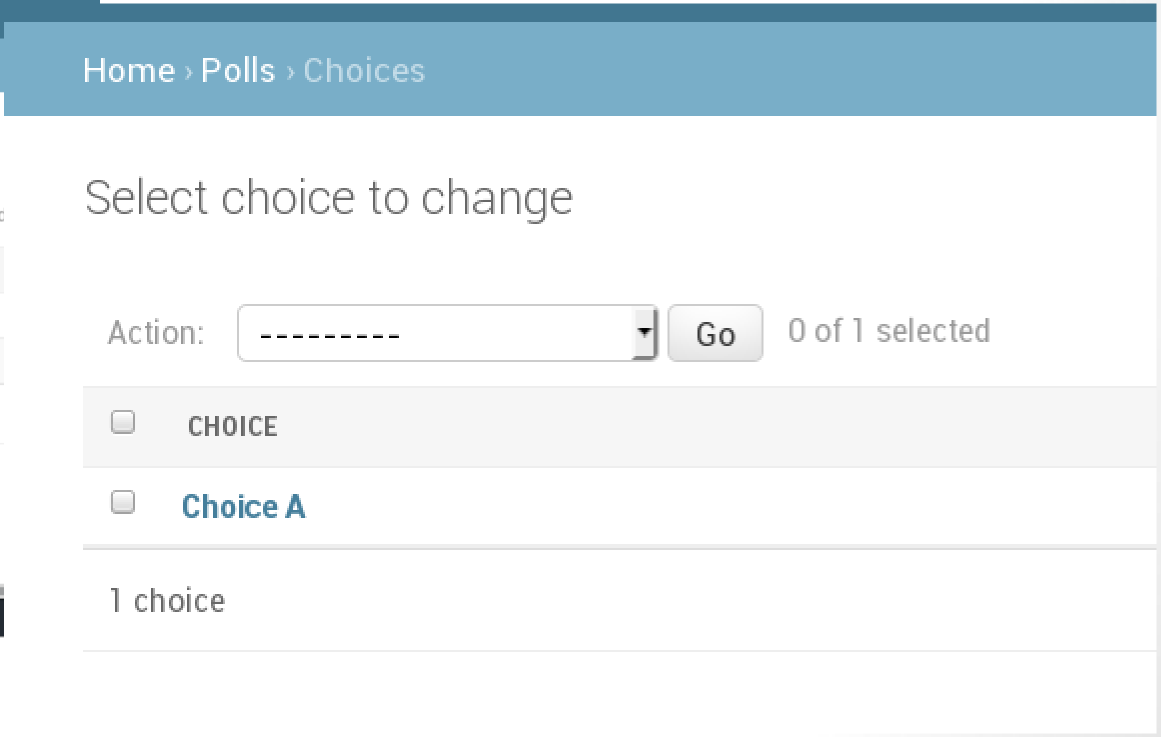
Home › Polls › Questions

Select question to change

Action: --------- Go 0 of 2 selected

QUESTION
This is another question
This is my new question

2 questions



Home › Polls › Choices

Select choice to change

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

CHOICE
Choice A

1 choice

App Polls

- Run the commands:
  - **python manage.py shell**

This will run the django project in a Python shell that will replicate the Django commands in the backend

```
from polls.models import Question, Choice # Import the model classes we just wrote.
Question.objects.all() # extract the objects created
<QuerySet [<Question: This is my new question>, <Question: This is another
question>]>
# Create a new Question.
# Support for time zones is enabled in the default settings file, so
# Django expects a datetime with tzinfo for pub_date. Use timezone.now()
# instead of datetime.datetime.now() and it will do the right thing.
from django.utils import timezone
q = Question(question_text="What's new?", pub_date=timezone.now())
# Save the object into the database. You have to call save() explicitly.
q.save()
q.id
q.question_text, q.pub_date
"What's new?", datetime.datetime(2012, 2, 26, 13, 0, 0, 775217, tzinfo=<UTC>)
# Change values by changing the attributes, then calling save().
q.question_text = "What's up?"
q.save()
```

App Polls

Edit The file polls/models.py to look like this:

```
from django.utils import timezone
class Question(models.Model):
    """
    Database table to include questions associated with the app.
    """
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')
    def __str__(self):
        """
        The text that appears in the object represented
        """
        return self.question_text
    def was_published_recently(self):
        """Determine if the question was published in the last day."""
        return self.pub_date >= timezone.now() - datetime.timedelta(days=1)
```

App Polls

- Run the commands:
  - **python manage.py shell**

Test the newly created functionality

```
from polls.models import Question, Choice
# Make sure our __str__() addition worked.
Question.objects.all()
<QuerySet [<Question: This is my new question>, <Question: This is another
question>, <Question: What's new?>]>
q = Question.objects.get(pk=1)  ##get object by primary key
q.pub_date
datetime.datetime(2017, 6, 1, 14, 46, 32, tzinfo=<UTC>)
q.was_published_recently()
False
q.choice_set.create(choice_text='Not much', votes=0)
q.choice_set.create(choice_text='The sky', votes=0)
c = q.choice_set.create(choice_text='Just hacking again', votes=0)
c
<Choice: Just hacking again>
q.choice_set.all()
<QuerySet [<Choice: Not much>, <Choice: The sky>, <Choice: Just hacking
again>]>
```

App Polls

- Run the commands:
  - **python manage.py dbshell**

  This will run the django project database in its native shell. In this case, sqlite

```
SQLite version 3.8.9 2015-04-08 12:16:33
Enter ".help" for usage hints.
sqlite> .headers on  /* will change output to include header row */
sqlite> .mode columns  /*change output to column mode */
sqlite> .tables  /* show all tables */
auth_group                  django_admin_log
auth_group_permissions      django_content_type
auth_permission             django_migrations
auth_user                   django_session
auth_user_groups            polls_choice
auth_user_user_permissions  polls_question
sqlite> select * from polls_question;  /* execute table lookup to list all entries*/
id          question_text               pub_date
----------  -----------------------     -------------------
1           This is my new question     2017-06-01 14:46:32
2           This is another questio     2017-06-30 12:00:00
3           What's new?                 2017-06-15 15:17:03
sqlite>  .quit  /*quit the shell*/
```