Django 最初のアプリケーション 2 (Admin サイトの構築)

[Python][Django][前][次]

Python の概要も分かり易い .

・The Django Book

・http://docs.djangoproject.com/en/dev/intro/tutorial02/#intro-tutorial02

を参考にサンプルアプリケーションを作成してみる

Admin サイトを有効化

・Django の Admin サイトはデフォルトでは有効でない

■有効化手順

"django.contrib.admin" を INSTALLED_APPS セッティングに追加する

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin',
    'mysite.polls'
)
```

python manage.py syncdb を実行。データベースが更新される必要がある

```
# python manage.py syncdb
Creating table django_admin_log
Installing index for admin.LogEntry model
```

mysite/urls.py ファイルを編集し、"Uncomment the next two lines..." 以下のコメントをはずす

```
from django.conf.urls.defaults import *

# Uncomment the next two lines to enable the admin:
from django.contrib import admin   # <- コメントはずす
admin.autodiscover()               # <- コメントはずす

urlpatterns = patterns('',
    # Example:
    # (r'^mysite/', include('mysite.foo.urls')),

    # Uncomment the admin/doc line below and add 'django.contrib.admindocs'
    # to INSTALLED_APPS to enable admin documentation:
    # (r'^admin/doc/', include('django.contrib.admindocs.urls')),

    # Uncomment the next line to enable the admin:
    (r'^admin/(.*)', admin.site.root),   # <- コメントはずす
)
```

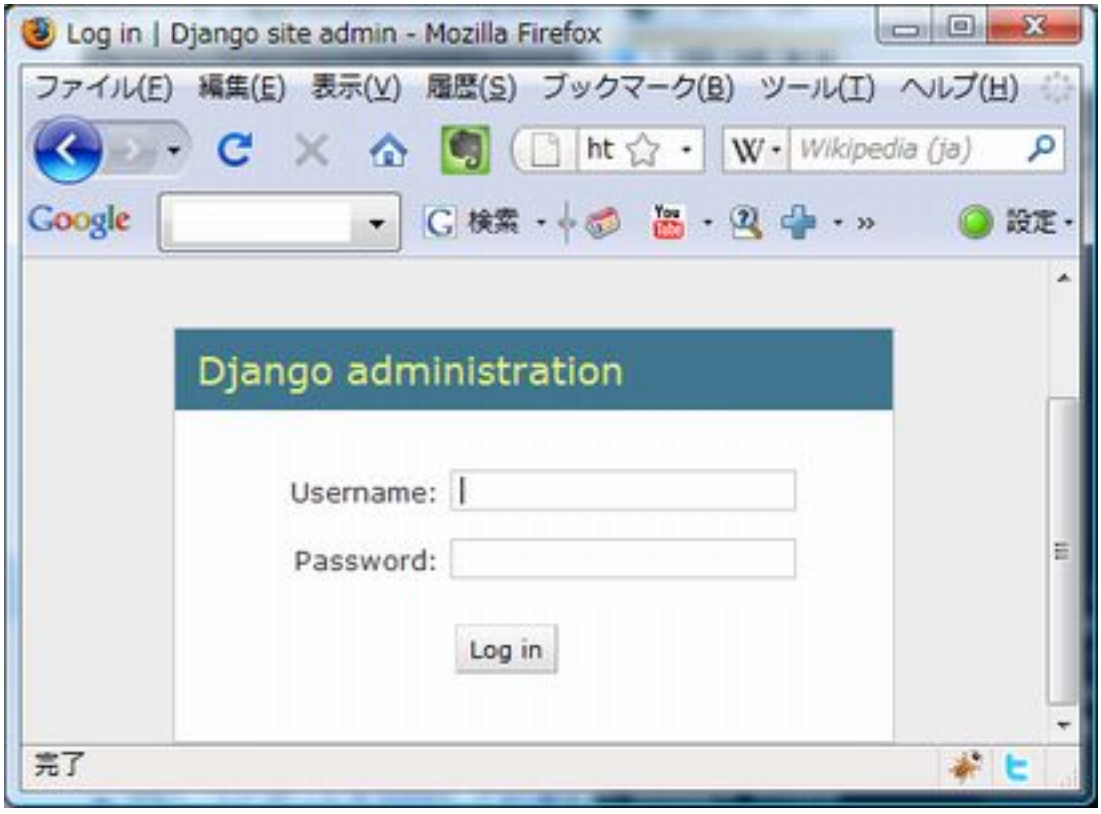
開発サーバー起動

・アドレスとポートを指定して起動

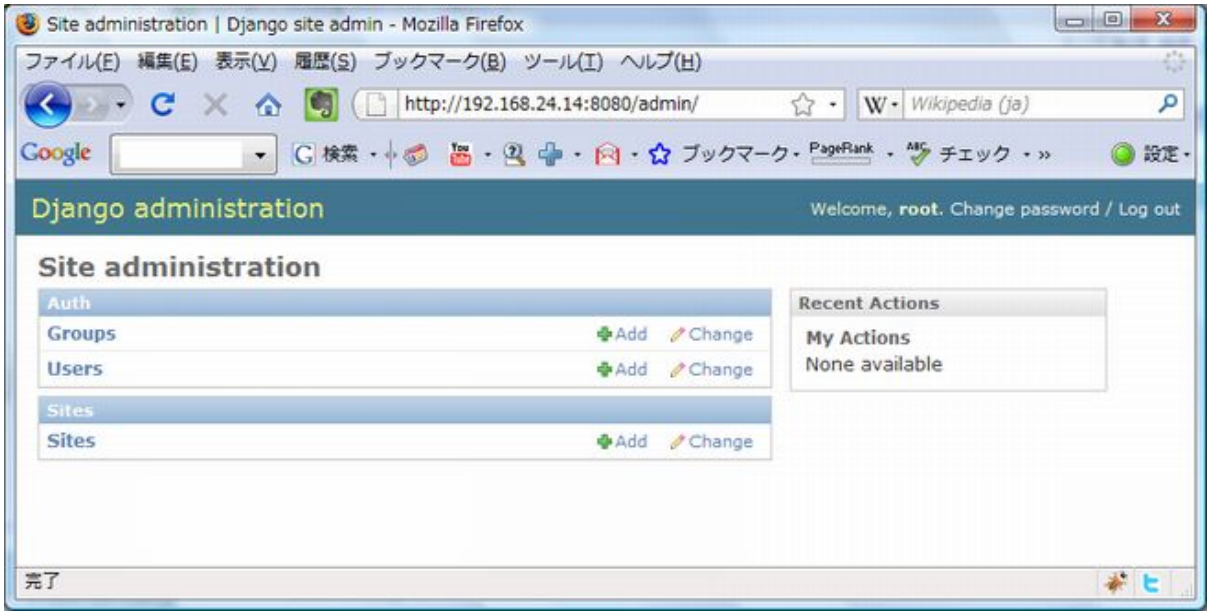
```
# python manage.py runserver 192.168.24.14:8080
Validating models...
0 errors found

Django version 1.0.2 final, using settings 'mysite.settings'
Development server is running at http://192.168.24.14:8080/
Quit the server with CONTROL-C.
```

■ Admin サイトへログイン

アクセス http:// ホスト : ポート /admin/



Django 最初のアプリケーション 1 で設定したユーザ ID とパスワードでログイン



作成した、Poll オブジェクトを Admin サイトに追加する

・mysite/polls に admin.py ファイルを作成し、以下の内容を記述

```
from mysite.polls.models import Poll
from django.contrib import admin

admin.site.register(Poll)
```

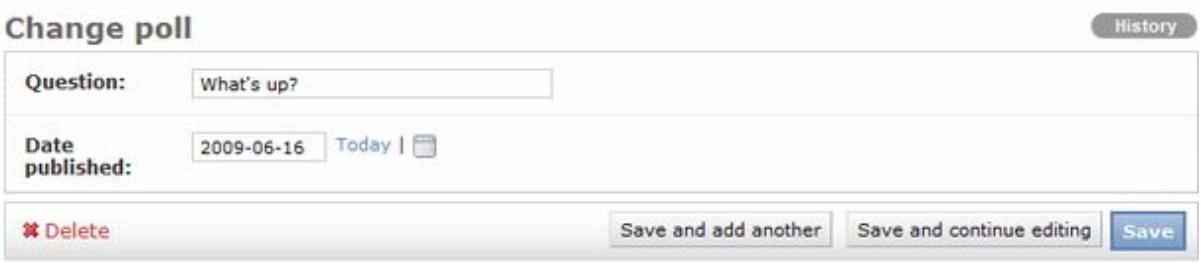
・サーバーを再起動

■機能の確認

・Poll を登録したので Django のインデックスページに表示される

・Polls をクリックすると、チェンジリストページが開く。このページでは、データベース上のすべての Polls が表示される
・そのうちの 1 つを選択して変更することができる

・編集するために、"What's up" を選択



覚書

・Poll モデルから、form は自動生成される
・モデルのフィールド型 (DateTimeField, CharField) は、適切な HTML の Input タグとして表示される
・DateTimeField は、"Today" リンクや、カレンダーポップアップへのリンクが表示され、時間の場合、"Now" リンクや、時間を選択するポップアップへのリンクが表示される

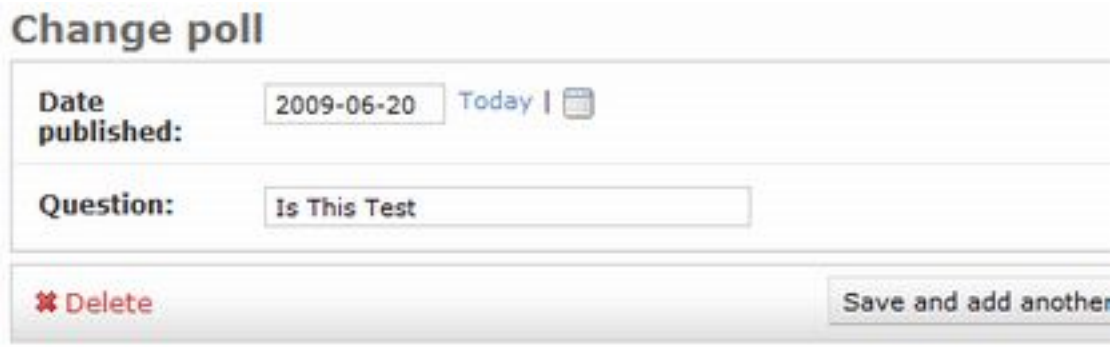
画面下部のオプション	内容
Save	保存して、チェンジリストへ戻る
Save and continue editing	保存して、リロード
Save and add other	保存して、ブランクページを開く
Delete	削除確認へ

Admin Form のカスタマイズ

・admin form の見た目や働きをカスタマイズしたい場合、/mysite/polls/admin.py を以下のように変更する

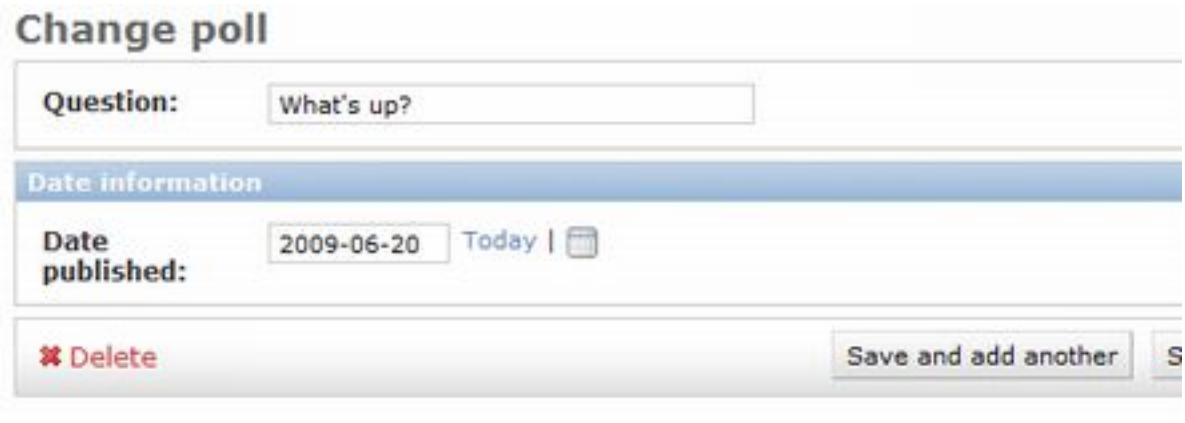
■フィールドの表示順を変更する

```
from mysite.polls.models import Poll
from django.contrib import admin

# 以下を追加
class PollAdmin(admin.ModelAdmin):
        fields = ['pub_date', 'question']

#PollAdmin 引数を追加
admin.site.register(Poll,PollAdmin)
```



■フィールドが多数ある場合など、フィールドセットに分割

```
class PollAdmin(admin.ModelAdmin):
fieldsets = [(None,               {'fields':['question']}),
             ('Date information',  {'fields':['pub_date']}),
 ]
```



■開閉可能にする

・Django が提供する collapse クラスを使用する

```
class PollAdmin(admin.ModelAdmin):
fieldsets = [(None,               {'fields':['question']}),
             ('Date information',  {'fields':['pub_date'],'classes':['collapse']}),
 ]
```

■関連オブジェクトの追加

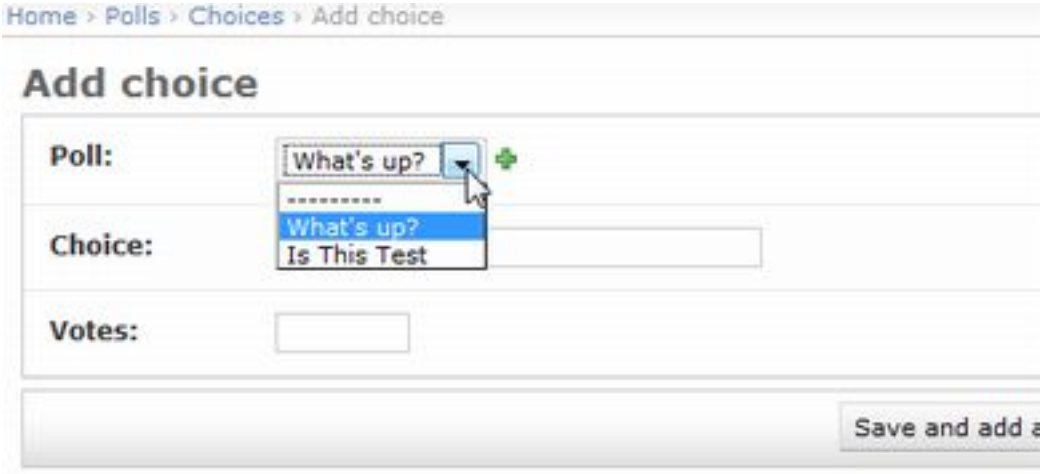
・Poll は複数の Choice 持つが、Poll の admin ページでは Choice が表示されていない。

1 つ目の方法

・Poll と同様 Choice も admin に 追加する

```
from mysite.polls.models import Poll, Choice

from django.contrib import admin

class PollAdmin(admin.ModelAdmin):
        fieldsets = [
                (None    ,               {'fields':['question']}),
                ('Date information',     {'fields':['pub_date'],'classes':['collapse']}),
        ]
admin.site.register(Poll,PollAdmin)
admin.site.register(Choice)
```

・Choice の追加に、Poll を指定するオプションが表示される

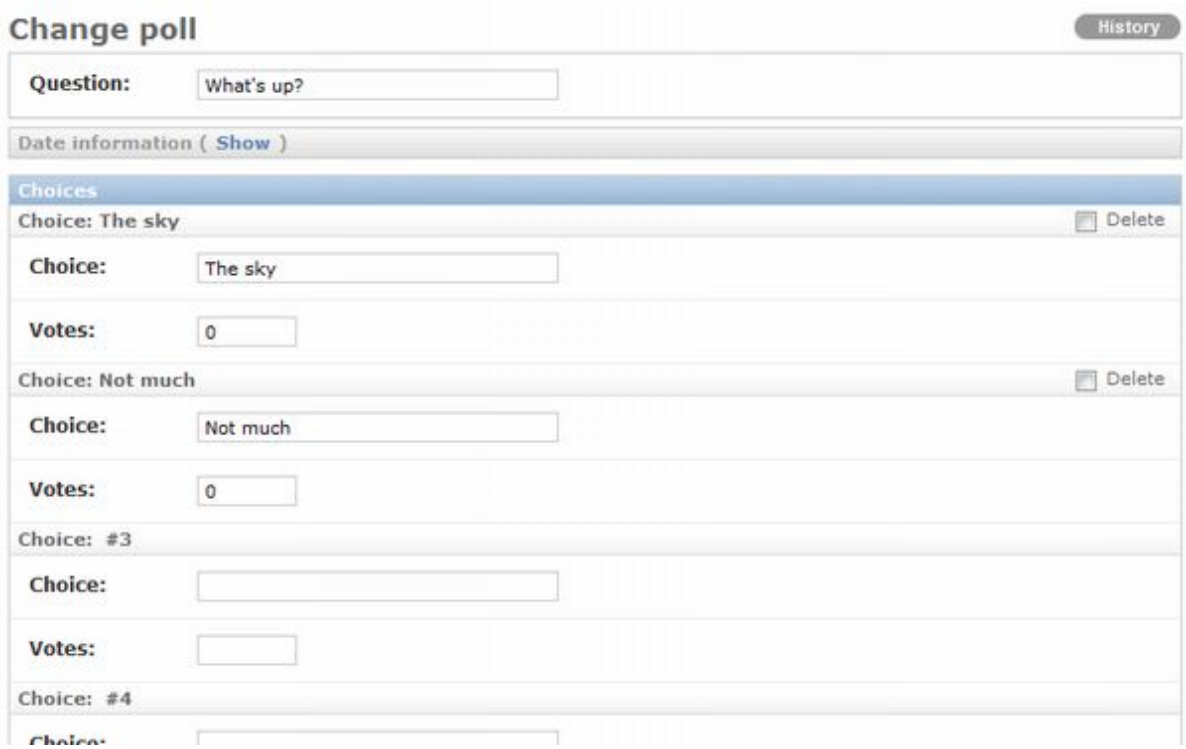


・Django は外部キーを把握しており、プルダウンにすべての Poll を選択可能にする

2 つ目の方法

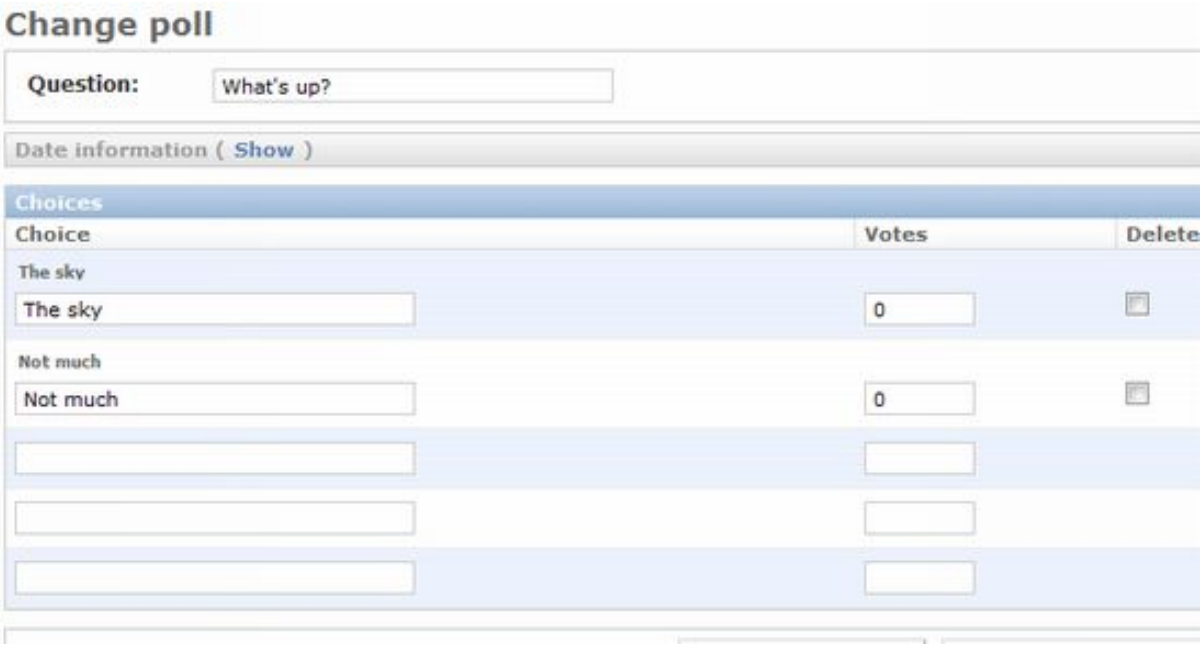
・1 つ目の方法は非効率。Poll オブジェクトから、Choice オブジェクトを追加できるようにするには、以下のようにする

```
from mysite.polls.models import Poll, Choice
from django.contrib import admin

class ChoiceInline(admin.StackedInline):
        model = Choice
        extra = 3

class PollAdmin(admin.ModelAdmin):
        fieldsets = [
                (None    ,               {'fields':['question']}),
                ('Date information',     {'fields':['pub_date'],'classes':['collapse']}),
        ]
        inlines = [ChoiceInline]

admin.site.register(Poll,PollAdmin)
#admin.site.register(Choice)
```



・StackedInline -> TabularInline に変更すると、以下のような表示になる

```
class ChoiceInline(admin.TabularInline):
```



チェンジリストのカスタマイズ

■一覧にカラムを表示

・Django ではデフォルトで、オブジェクトの str() メソッドを一覧に表示しているが、役に立つ別のフィールドを list_display オプションを利用して表示することもできる

```
class PollAdmin(admin.ModelAdmin):
        fieldsets = [
                (None    ,               {'fields':['question']}),
                ('Date information',    {'fields':['pub_date'],'classes':['collapse']}),
        ]
        inlines = [ChoiceInline]
        # この行を追加
        list_display = ('question', 'pub_date', 'was_published_today')
```

■カラムの名称を変更

・カラムヘッダーをクリックすることでソートできるが、任意に追加した was_published_today メソッドなどでは利用不可
・デフォルトではカラムにメソッド名が表示されているが、short_description 属性にて変更できる

/mysite/polls/models.py

```
class Poll(models.Model):
            :
```

```
        def was_published_today(self):
                return self.pub_date == datetime.date.today()
        # 以下を追加
        was_published_today.short_description = 'Published today?'
```

■フィルターの追加

```
    class PollAdmin(admin.ModelAdmin):
            fieldsets = [
                    (None    ,               {'fields':['question']}),
                    ('Date information',     {'fields':['pub_date'],'classes':['collapse']}),        ]
                  inlines   =   [ChoiceInline]              list_display   =   ('question',   'pub_date',
'was_published_today')
            # 以下を追加
            list_filter = ['pub_date']
```

pub_date に対するフィルターサイドバーが表示される

■検索フィールドの追加

以下を追加

```
    search_fields = ['question']
```

■日付フィールドでのドリルダウン

以下を追加

```
    date_hierarchy = 'pub_date'
```

Admin のルック＆フィールの変更

・Django のテンプレートシステム機能によって、ルック＆フィールを簡単に変更できる
・mysite/settings.py 設定ファイルを開き、Django テンプレートの設定を TEMPLATE_DIRS で変更する
・デフォルトで、TEMPLATE_DIRS は空になっている

```
TEMPLATE_DIRS = (
    # Put strings here, like "/home/html/django_templates" or "C:/www/django/templates".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
    "/home/my_username/mytemplates",
)
```

・TEMPLATE_DIRS で指定したフォルダの下に admin フォルダを作成し、/django/contrib/admin/templates/admin/base_site.html をコピーする

```
# cd /usr/local/lib/python2.6/site-packages/django/contrib/admin/templates/admin
# cp base_site.html /home/my_username/mytemplates/admin
```

・コピーしたファイルを適当に変更

```
{% load i18n %}
{% load i18n %}

{% block title %}{{ title }} | {% trans 'My Sample site admin' %}{% endblock %}

{% block branding %}
<h1 id="site-name">{% trans 'My Sample  administration' %}</h1>
{% endblock %}

{% block nav-global %}{% endblock %}
```

・タイトルが変更された

[前][次]