

Python Hack-a-thon #2 Django ハンズオン

事前にインストールしておくもの

- Python 2.5 or 2.6
- Django 1.1
- IPython
- SQLite 3

ドキュメントなど

- [Python ドキュメント](#)
- [Django ドキュメント](#)

ゲストブックアプリを動かしてみよう

サンプルのゲストブックアプリケーションを動かしてみます。

startproject

はじめに、アプリケーションを動作させるためのプロジェクトを作成します。

ターミナル(or コマンドプロンプト)から次のように入力します。

```
django-admin.py startproject プロジェクト名
```

プロジェクト名は半角英数で入力してください。アンダースコアは利用できます(Pythonのモジュール名として有効な名前を利用したほうが良いです)

プロジェクト名とアプリケーションの名前が同じにならないように注意してください。

日本語を含むパスでは、うまく動作しないことがあります。

これで、プロジェクト名のディレクトリが作成されます。

インストールの仕方によっては django-admin.py が django-admin になっているかもしれません。

プロジェクトの設定を行う

プロジェクト内の settings.py を編集します。編集項目は以下の通りです。

```
import os
BASE_DIR = os.path.dirname(os.path.abspath(__file__)) # プロジェクトディレクトリを取得

DATABASE_ENGINE = 'sqlite3' # データベースエンジンはSQLite3
DATABASE_NAME = os.path.join(BASE_DIR, 'data.db') # データベースファイル
TIME_ZONE = 'Asia/Tokyo' # タイムゾーンは東京
LANGUAGE_CODE = 'ja' # 言語は日本語
```

アプリケーションを追加する

guestbook アプリケーションをプロジェクトのディレクトリにコピーします。続いて settings.py の INSTALLED_APPS に guestbook を追加します。一緒に Django の管理アプリケーションもインストールしておきます。

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin', # これを追加
    'guestbook', # これを追加
)
```

これでアプリケーションをプロジェクトに追加できました。

アプリケーションのURLを有効にする

アプリケーションのURLを有効にするため、プロジェクト内の `urls.py` を編集します。 `urls.py` を次のように書き換えます。

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

from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    (r'^admin/', include(admin.site.urls)),
    (r'', include('guestbook.urls')),
)
```

データベースへ反映させる

インストールしたアプリケーションのモデルをデータベースに反映させます。ターミナルで以下のコマンドを実行します。

```
python manage.py syncdb
```

管理ユーザの作成を聞かれた場合、作成しておいてください。

これでデータベースへの反映ができました。

開発用サーバを起動して動かしてみる

開発用サーバを起動するには、ターミナルで以下のコマンドを実行します。

```
python manage.py runserver
```

デフォルトでは `127.0.0.1:8000` で起動します。

Webブラウザから、 <http://127.0.0.1:8000/> へアクセスするとゲストブックアプリケーションを利用できます。

管理画面は <http://127.0.0.1:8000/admin/> でアクセスできます。

対話シェルを利用してみる

DjangoではPythonの対話シェルを利用して、データベース等にアクセスすることができます。利用するには、以下のコマンドを実行します。

```
python manage.py shell
```

使うと幸せになれるアプリ

django-debug-toolbar

django-debug-toolbar を使うと、テンプレートやSQLのデバッグなどが楽になります。

次のコマンドでインストールできます。

```
easy_install django-debug-toolbar
```

使用するには、 settings.py を編集します。

```
MIDDLEWARE_CLASSES = (
    'django.middleware.common.CommonMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'debug_toolbar.middleware.DebugToolbarMiddleware', # これを追加
)

INSTALLED_APPS = (
    # 中略
    'debug_toolbar', # これを追加
)

# 以下を追加
INTERNAL_IPS = (
    '127.0.0.1',
)
```

以上です。開発サーバを起動してWebブラウザでページを表示してみてください。サイドバーが追加されているはずです。

django-command-extensions

django-command-extensions を使うと manage.py に便利なコマンドが多数追加されます。

<http://code.google.com/p/django-command-extensions/> からダウンロードできます。

インストールは、ダウンロードしたアーカイブを展開し、そのディレクトリに移動して以下のコマンドを実行します。

```
python setup.py install
```

使用するには、 settings.py を編集します。

```
INSTALLED_APPS = (
    # 中略
    'django_extensions', # これを追加
)
```

manage.py の help コマンドでコマンド一覧を見てみるとコマンドが増えていることが確認できます。