

# Python-блог

воскресенье, 4 ноября 2012 г.

## Документация South - Перевод. Часть 1: Основы

Добро пожаловать в руководство. Тут мы постараемся рассказать об основах использования South и дать несколько общих советов как ещё его можно использовать. Если Вы никогда не слышали о библиотеках для миграции, тогда сперва прочитайте раздел **"что такое миграция"**. Это поможет Вам лучше понять для чего предназначены South и его альтернативы, такие как django-evolution.

В этом руководстве мы предполагаем, что South у Вас уже установлен и настроен, иначе смотрите раздел **"установка"**.

### Приступаем

В этом руководстве мы рассмотрим процесс миграции на новое приложение. Вопросы изменения схемы существующего приложения мы рассмотрим позже.

Во-первых, надо отметить, что South работает с каждым приложением по отдельности. Миграции сохраняются в коде приложения (хотя их можно хранить там, где Вы захотите - см "SOUTH\_MIGRATION\_MODULES"). Если для приложения не определена никакая миграция, то South будет его просто игнорировать и оно будет использовать syncdb.

Так что возьмём проект для работы (или сделаем новое, определив БД и прочие настройки) и создадим новое приложение:

```
./manage.py startapp southtut
```

Как и предполагается, эта команда создаст новую папку southtut. Во-первых, добавьте это приложение в INSTALLED\_APPS, после чего откройте созданный southtut/models.py и создайте новую модель:

```
from django.db import models

class Knight(models.Model):
    name = models.CharField(max_length=100)
    of_the_round_table = models.BooleanField()
```

Просто, да? Но теперь, вместо того, чтобы запустить syncdb для создания таблицы для модели в нашей БД, мы создадим для этого миграцию.

### Первая миграция

В South есть несколько способов создать миграцию. Некоторые автоматические, некоторые делают руками. Как стандартный пользователь, Вы скорее всего будете использовать два автоматических способа - --auto и --initial.

--auto смотрит на предыдущую миграцию, смотрит, что изменилось и создаёт миграцию, которая применяет эти изменения. Например, если Вы добавляете поле в модель, то --auto это заметит и сделает миграцию, которая создаст в таблице новую колонку, соответствующую этому полю в модели.

Но, как Вы наверняка заметили, --auto требуется предыдущая миграция, а в нашем новом приложении нет ещё ни одной миграции. Вместо этого мы воспользуемся --initial, который создаёт таблицы и индексы для всех моделей в приложении. Это то, что Вам нужно в самом начале, аналог syncdb, тогда как --auto Вам понадобится уже позже, для обслуживания изменений.

Так что давайте создадим нашу первую миграцию:

```
$ ./manage.py schemamigration southtut --initial
Creating migrations directory at
'/home/andrew/Programs/litret/southtut/migrations'...
Creating __init__.py in
'/home/andrew/Programs/litret/southtut/migrations'...
+ Added model southtut.Knight
Created 0001_initial.py. You can now apply this migration with:
./manage.py migrate southtut
```

(Если на этом этапе Вы получите ошибку, что south\_migrationhistory не существует, значит Вы забыли запустить syncdb после **установки South**)

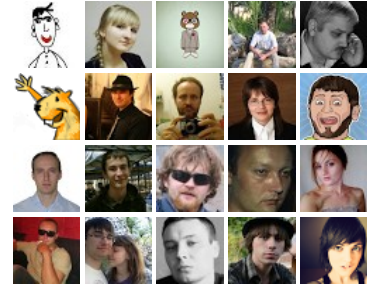
Как Вы можете видеть, эта команда создала для нас папку миграции и создала внутри неё новую миграцию. Всё, что нам осталось - лишь применить её к БД:

```
$ ./manage.py migrate southtut
Running migrations for southtut:
- Migrating forwards to 0001_initial.
> southtut:0001_initial
```

### Google+ Followers

#### Python-lab. Блог о Python

Подписаться



Кто вас добавил (49)

[Весь список](#)

### Flatrr this blog

0

### Ярлыки

2.7 (4) **2.x (23)** 3.3 (6) 3.x (7) [\\_\\_cause\\_\\_ \(1\)](#) [\\_\\_context\\_\\_ \(1\)](#) [\\_\\_del\\_\\_ \(2\)](#) [\\_\\_main\\_\\_ \(1\)](#) [\\_\\_qualname\\_\\_ \(1\)](#) [apache \(1\)](#) [Bug tracker \(1\)](#) [bug? \(1\)](#) [C \(1\)](#) [cffi \(1\)](#) [cliff \(2\)](#) [code object \(2\)](#) [collections \(1\)](#) [compile \(1\)](#) [core development \(2\)](#) [CPython \(1\)](#) [CPython 2.x \(1\)](#) [css \(1\)](#) [csv \(1\)](#) [del \(2\)](#) [demakein \(1\)](#) [diffib \(1\)](#) [Distribution \(1\)](#) [distutils \(1\)](#) [django \(9\)](#) [django-model \(8\)](#) [doctest \(1\)](#) [е-книга \(2\)](#) [easy\\_install \(1\)](#) [egg \(1\)](#) [Environment \(1\)](#) [exception \(2\)](#) [exec \(1\)](#) [fabric \(3\)](#) [flask \(3\)](#) [freebsd \(1\)](#) [ftplib \(1\)](#) [Game Theory \(1\)](#) [gmail \(1\)](#) [google api \(1\)](#) [google docs \(1\)](#) [google drive api \(1\)](#) [GUI \(9\)](#) [html \(1\)](#) [http \(2\)](#) [IMAP Exchange \(1\)](#) [IMAPClient \(2\)](#) [inspect \(1\)](#) [IPC \(1\)](#) [inspect.stack \(1\)](#) [kwargs \(1\)](#) [liburl \(2\)](#) [liburl2 \(2\)](#) [line\\_profiler \(1\)](#) [list \(2\)](#) [list.remove\(\) \(1\)](#) [logging \(3\)](#) [logsna \(1\)](#) [kml \(1\)](#) [math \(1\)](#) [mechanize \(1\)](#) [memory\\_profiler \(1\)](#) [mercurial \(1\)](#) [metaPDF \(1\)](#) [moinmoin2 \(7\)](#) [moinmoin2 backup \(1\)](#) [moinmoin2 индексы \(1\)](#) [moinmoin2 настройка \(3\)](#) [multiprocessing \(1\)](#) [namedtuple \(2\)](#) [Nikola \(3\)](#) [Nuitka \(1\)](#) [Numpy \(1\)](#) [ObjectListView \(1\)](#) [objgraph \(1\)](#) [ods \(1\)](#) [OpenSSH \(1\)](#) [os.path.join \(1\)](#) [paramiko \(3\)](#) [parser \(1\)](#) [patch \(1\)](#) [pdf \(6\)](#) [pdfw \(1\)](#) [PEP \(4\)](#) [persona \(1\)](#) [Phoenix \(1\)](#) [PIL \(1\)](#) [Pillow \(1\)](#) [pip \(1\)](#) [pisa \(1\)](#) [pkg\\_resources \(1\)](#) [plumbum \(1\)](#) [psutil \(1\)](#) [pubsub \(3\)](#) [py3k \(1\)](#) [pybooklet \(1\)](#) [PyCharm \(1\)](#) [pypdf \(1\)](#) [pylibfidi \(1\)](#) [PyMOTW \(1\)](#) [PyOpenGL \(1\)](#) [pyPDF \(2\)](#) [pypdf2 \(1\)](#) [PyPI \(1\)](#) [pypy \(1\)](#) [pyru numpy \(1\)](#) [pyqmativ \(1\)](#) [pyramid \(1\)](#) [pyramid\\_persona \(1\)](#) [python 3.x \(2\)](#) [python-qrcode \(1\)](#) [pythoncom \(1\)](#) [pyvideo \(1\)](#) [PyWin32 \(3\)](#) [QR \(1\)](#) [raise \(2\)](#) [raise from \(2\)](#) [reportlab \(1\)](#) [requests \(2\)](#) [Requirement \(1\)](#) [RPyC \(1\)](#) [SetupTools \(1\)](#) [south \(8\)](#) [sphinx \(2\)](#) [ssh \(1\)](#) [sys.argv \(1\)](#) [threading \(1\)](#) [threadsafe \(1\)](#) [Tkinter \(1\)](#) [tracelib \(1\)](#) [twisted \(2\)](#) [twitter \(1\)](#) [virtualenv \(1\)](#) [virtualenvwrapper \(2\)](#) [web \(3\)](#) [web2py \(1\)](#) [webbrowser \(1\)](#) [wheezy.core \(1\)](#) [wiki \(6\)](#) [windows \(3\)](#) [Windows 7 \(1\)](#) [winshell \(2\)](#) [Wizard \(1\)](#) [WorkingSet \(1\)](#) [wx.CallAfter \(1\)](#) [wx.CallLater \(1\)](#) [wx.Notebook \(1\)](#) [wx.PostEvent \(1\)](#) [wx.Timer \(1\)](#) [wxPython \(16\)](#) [wxPython in Action \(8\)](#) [xhtml \(1\)](#) [xhtml2pdf \(1\)](#) [xml \(1\)](#) [yield \(1\)](#) [yield from \(1\)](#) [Zope Interface \(1\)](#) [администрирование \(1\)](#) [алгоритм \(1\)](#) [алгоритм Луна \(1\)](#) [анализ данных \(1\)](#) [аргументы \(1\)](#) [библиотеки \(34\)](#) [библиотеки 2.x \(3\)](#) [вакансии \(1\)](#) [взлом \(1\)](#) [генераторы \(1\)](#) [декоратор \(1\)](#) [документация \(24\)](#) [игра \(1\)](#) [итератор \(1\)](#) [кросс-платформенность \(1\)](#) [логи \(2\)](#) [мехрограммное взаимодействие \(1\)](#) [модули \(2\)](#)

- Loading initial data for southtut.

Теперь South создал новую таблицу в нашей модели - можете проверить если хотите, и добавить нескольких Knight при помощи ./manage.py shell.

Изменение модели

До сих пор мы не делали ничего, с чем бы не смог справиться syncdb. Теперь пришло время изменить нашу модель. Давайте добавим к ней ещё одно поле:

```
from django.db import models

class Knight(models.Model):
    name = models.CharField(max_length=100)
    of_the_round_table = models.BooleanField()
    dances_whenever_able = models.BooleanField()
```

Теперь, если мы не будем использовать миграции, так просто добавить новую колонку к таблице southtut\_knight уже не получится. Но при помощи South мы сможем сделать это всего в два шага: создать миграцию для отражения изменения и затем применить её: Во-первых, создадим миграцию при помощи --auto:

```
$ ./manage.py schemamigration southtut --auto
+ Added field dances_whenever_able on southtut.Knight
Created 0002_auto__add_field_knight_dances_whenever_able.py. You can n
ow apply this migration with: ./manage.py migrate southtut
```

(Обратите внимание, что South автоматически подбирает имя для миграции. Вы можете задать для миграции своё имя, указав его в качестве другого аргумента). Теперь давайте её применим:

```
$ ./manage.py migrate southtut
Running migrations for southtut:
- Migrating forwards to 0002_auto__add_field_knight_dances_whenever_a
ble.
> southtut:0002_auto__add_field_knight_dances_whenever_able
- Loading initial data for southtut.
```

После этого наша новая колонка создана и опять же это можно проверить.  
**Конвертация существующего приложения**  
Иногда, особенно когда Вы добавляете South в проект, Вы хотите использовать его для уже существующего приложения - там, где уже есть созданные таблицы. В этом и состоит отличие работы с существующим приложением от работы с новым приложением. И о том, как справиться с такой задачей смотрите страницу "конвертация приложения (пока не переведено)".  
Теперь, когда Вы знаете как использовать South, можно перейти ко второй части руководства.

Автор: [Ishayahu Lastov](#) на 15:21

+3 Рекомендовать в Google

Ярлыки: [django](#), [django-model](#), [south](#), [документация](#)

0

Комментариев нет:

Отправить комментарий

Введите комментарий...

Подпись комментария: 

Аккаунт Google

Публикация

Просмотр

новость (3) **обзоры** (4) области видимости (1)  
пакеты (2) парсинг (1) производительность (1)  
развёртывание (3) **релиз** (16) сеть Фейстеля  
(1) системные утилиты (1) скачивание файлов (1)  
создание блогов (1) создание сайтов (2) стили кода  
(1) табуляция (1) требования (1) утилиты (1) хакинг (1)  
ярлыки (3)

Постоянные читатели

Присоединиться к сайту

Инструменты Google для создания интернет-сообществ

Участники (22)

Уже являетесь участником? [Войти](#)

Follow by Email

Email address...

Submit

67

Подпишитесь на

Сообщения

Комментарии

Обо мне

**Ishayahu Lastov**

Подписаться

[Просмотреть профиль](#)

- Архив блога
- 2013 (25)

▼ 2012 (104)

▼ Ноябрь (15)

pisa / xhtml2pdf, конвертер, написанный на Python ...

PyBooklet: создание PDF с двумя страницами на лист...

wxPython in Action. Часть 3. Продвинутое использо...

Обзор книги: Steve Holden: Python для анализа данн...

Релиз: "Menno's Musings": IMAPClient 0.9.1 (Перево...

Документация South - Перевод. Часть 3: Дополнитель...

Документация South - Перевод. Часть 2. Более сложн...

Документация South - Перевод. Часть 1: Основы

Как создать генератор номера счёта? (Перевод)

Передача аргументов с неверным идентификатором (Пе...

Обновление статуса РуЗк №7 (Перевод)

Релиз Pillow 1-7-8 (Перевод)

Документация South - Перевод. Установка

Документация South - Перевод. Что такое миграция?

Документация South - Перевод. Про South

- ▶ Октябрь (13)
- ▶ Сентябрь (10)
- ▶ Август (4)
- ▶ Июль (23)
- ▶ Июнь (9)
- ▶ Май (17)
- ▶ Апрель (4)
- ▶ Март (9)

Следующее

Главная страница

Предыдущее

Подписаться на: [Комментарии к сообщению \(Atom\)](#)

Общее количество просмотров страницы



Yandex.metrika


Yandex.money

Перевод документации + Хостинг

Пожелания и т.д.

50 руб.

Поддержать

Яндекс  
ДЕНЬГИ 

Перевод проекту [Python-lab.ru](#)

GAlytic



Шаблон "Simple". Технологии [Blogger](#).