15 декабря 2008 в 19:10

# South — новый клёвый syncdb

Django*

Я совсем недавно начал работать с Django и меня практически сразу же взбесила ущербная команда syncdb, которая ничего толком не синхронизирует, умеет лишь создавать таблицы для новых моделей.

А добавление или удаление полей в уже существующие модели превращается в настоящий pain in ass — приходится «подсматривать» за тем, как ORM создала бы таблицы заново (manage.py sqlall) и вручную делать ALTER TABLE для изменившихся столбцов таблиц.

То есть, ORM в Django так или иначе предполагает активный трах с SQL shell, потому что в процессе прототипирования эти поля в моделях изменяются просто пачками.

Погуглив, я нашел несколько способов автоматизации изменения схемы БД и в итоге остановился на South. Эта утилита автоматизирует процесс db schema migration. (а название утилиты, насколько я понял, обыгрывает термин «миграция» — ведь зимой все птицы летят на йух :)).

Я выбрал South из-за того, что оно хранит историю миграций и можно делать undo/redo, а так же корректировать код миграции руками.

Полный туториал можно почитать здесь: south.aeracode.org/wiki/Tutorial, я лишь вкратце поясню, как оно работает.

## Установка

1. Сливаем svn'ом код утилиты отсюда: https://svn.aeracode.org/svn/south/trunk

2. Кладем его в python/lib/site-packages или еще куда-нибудь, где его будет видно

3. В settings.INSTALLED_APPS добавляем 'south'

4. Делаем привычный manage.py syncdb и видим, что он стал каким-то другим

## Использование

Теперь представим ситуацию: в уже существующую модель нам надо добавить пару полей. К примеру, таких:

```
class CompanyProduct (models.Model):

  # .. бла-бла, какие-то старые поля ..

  download_url = models.URLField (
    u'Ссылка для скачивания',
    blank = True, null = True)

  system_requirements = models.TextField (
    u'Системные требования',
    blank = True, null = True)

* This source code was highlighted with Source Code Highlighter.
```

Создаём миграцию:

```
python manage.py startmigration [имя_вашего_app] add_download_section
  --add-field CompanyProduct.download_url
  --add-field CompanyProduct.system_requirements
```

```
Creating __init__.py in '[path_to_app]\migrations'...
Created 0001_add_download_section.py.
```

Применяем её:

```
python manage.py migrate [имя_вашего_app]
```

```
Running migrations for [имя_вашего_app]:
- Migrating forwards to 0001_add_download_section.
> [имя_вашего_app]: 0001_add_download_section
  = ALTER TABLE "products" ADD COLUMN "download_url" varchar(200) NULL; []
  = ALTER TABLE "products" ADD COLUMN "system_requirements" text NULL; []
- Loading initial data for [имя_вашего_app].
```

Движимые научным любопытством, смотрим, что нам сгенерил South в 0001_add_download_section.py:

```
from south.db import db
from django.db import models
from [имя_вашего_app].models import *

class Migration:

  def forwards(self):

    # Adding field 'CompanyProduct.download_url'
    db.add_column('products', 'download_url',
        models.URLField (u'Ссылка для скачивания', blank = True, null = True))

    # Adding field 'CompanyProduct.system_requirements'
    db.add_column('products', 'system_requirements',
        models.TextField (u'Системные требования', blank = True, null = True))


  def backwards(self):

    # Deleting field 'CompanyProduct.download_url'
    db.delete_column('products', 'download_url')

    # Deleting field 'CompanyProduct.system_requirements'
    db.delete_column('products', 'system_requirements')
```

## Косяк

В моём случае напильником пришлось добавить вывод "# coding=utf-8" в начало миграционного скрипта (см. /south/management/commands/startmigration.py, строка 290), иначе он фейлился из-за unicode-строк в нём.

django, south, syncdb, миграция, SQL, ORM

+33 77 alex_blank 50,9



комментарии (34) отслеживать новые: в почте в трекере

 **miolini**, 15 декабря 2008 в 19:26 # **–5**

Как хорошо с hibernate.

 **alex_blank**, 15 декабря 2008 в 19:29 # ↑ **0**

мне больше нравится db4o, сразу объектная БД и с LINQ в .NET дружит (т.е. запросы к ней поддерживаются синтаксисом языка, в отличие от Hibernate)

 **miolini**, 15 декабря 2008 в 19:33 # ↑ **0**

Пока у вас не появится более менее сложные запросы на большие куски данных — всё будет хорошо. Это же odbms.

 **romik**, 8 апреля 2009 в 13:45 # ↑ **0**

Как будто Hibernate умеет обновлять схему. Те же проблемы.

 **miolini**, 8 апреля 2009 в 14:06 # ↑ **0**

Что значит «как будто?

 **romik**, 8 апреля 2009 в 14:17 # ↑ **0**

А что, умеет? hibernate.hbm2ddl.auto = update не считается, так как оно умеет только добавлять новые поля. Или уже научилось менять и удалять?

 **miolini**, 8 апреля 2009 в 14:21 # ↑ **0**

Почему же не считается? Он приводит схему базы к такому виду, чтобы маппинги работали. А на менять и удалять есть create-drop.

 **romik**, 8 апреля 2009 в 14:29 # ↑ **0**

Потому что после update получаются всякие безобразия. То not-null поле из базы не удалилось и вставка не работает, то размер поля сменился и *иногда* что-нибудь падает. Особенно напрягает слово *иногда*, так как про то, что схема могла измениться, к этому моменту уже забываешь.

В общем, update штука ошибкоопасная, хоть и полезная в процессе разработки. Для production придётся менять схему вручную или разрабатывать автомигратор a-la south.

 **miolini**, 8 апреля 2009 в 14:31 # ↑ **0**

Не понял — вы в продакшен хотите использовать какие-то вариации на тему обновления схемы? С ума сошли?

 **romik**, 8 апреля 2009 в 22:49 # ↑ **0**

А вы считаете, что в продакшн схема никогда не должна меняться?

 **krig**, 15 декабря 2008 в 20:19 # **+3**

У Джанго, конечно, не все хорошо в плане обновления схемы, но и не так плохо, как вы описываете.
`python manage.py reset [appname]`
И схема будет обновлена.
Есть, правда, и огромный недостаток, который не позволяет выполнять команду на продакш сервере — вся данные из таблиц приложения очищаются.

А вобще, за инструмент огромное спасибо! Сам некоторое время столкнулся с такой проблемой — пока что делал как написано в мануале Джанго — писал руками альтеры =)

 **alex_blank**, 15 декабря 2008 в 20:25 # ↑ **+1**

ну, так резет ведь убивает содержимое базы, а это даже при начальной разработке сильно мешает — приходится заново 'рыбу' (тестовые данные) генерить

 **Riz**, 15 декабря 2008 в 21:15 # ↑ **+2**

С одной стороны это конечно гимморой, но с другой стороны, не стоит считать это прихотью разработчиков. При миграции как минимум нарушается логика для полей который не могут иметь пустое или нулевое значение.

Вот жаль что для фикстур нельзя дефолтные значениядля неизвестных полей указывать.

 **iwuvjhdva**, 16 декабря 2008 в 07:58 # ↑ **+1**

Fixtures частично спасают ситуацию.

alex_blank, 15 декабря 2008 в 20:29 # ↑ +3

btw, я бы побоялся использовать South «как есть» на продакшене — мало ли, какую херню он сгенерит

неплохо бы там сделать режим предпросмотра SQL statements при миграции…

spoof, 15 декабря 2008 в 22:32 # ↑ –5

а зачем менять бд на продакшене? :) по-идее все должно быть уже стабильно и вноситься маленькие фиксы :)

alex_blank, 15 декабря 2008 в 22:54 # ↑ +3

это в теории :)

l0rda, 15 декабря 2008 в 20:34 # +5

Я использую deseb(http://code.google.com/p/deseb/) и не имею никакого геморроя.

НЛО прилетело и опубликовало эту надпись здесь

l0rda, 15 декабря 2008 в 22:32 # ↑ 0

а зачем наследоваться от User?

не имел таких проблем, на 1.0 я не тестировал, проект крутится на предыдущей версии
но в целом отличная штука, проблем не имею с ним

НЛО прилетело и опубликовало эту надпись здесь

voidus, 16 декабря 2008 в 01:20 # ↑ 0

Всё пофиксили, замечательно работает.

voidus, 16 декабря 2008 в 01:21 # ↑ +1

В смысле, что с Django 1.0 работает… с вашей проблемой не сталкивался.

kmike, 15 декабря 2008 в 21:36 # +3

Выбирал из south и django-evolution.
Остановился на последнем, т.к. можно вручную миграции не описывать, ну и идеологии немножко разные у них.

spoof, 15 декабря 2008 в 22:33 # ↑ 0

я тоже на django-evolution остановился

termt, 15 декабря 2008 в 21:40 # +3

туда же django evolution

dbf, 15 декабря 2008 в 21:57 # +1

Так есть же deseb где не надо руками писать миграции, а он сам сравнивает модель со структурой бд.

НЛО прилетело и опубликовало эту надпись здесь

sgtpep, 16 декабря 2008 в 01:25 # +1

Слишком наворочено, решение добавляет лишних уровней абстракции.
Я обхожусь скриптом, который удаляет все таблицы, запускает syncdb и загружает начальные данные (fixtures).

Пример CMD-скрипта:

```
echo SHOW TABLES; > dump.sql
C:\Server\usr\local\mysql5\bin\mysql_run_to_import_dumps.exe -D ИМЯ_БД < dump.sql > tables.txt
echo ; > dump.sql
REM Тут у нас обход каждой строки в списке имен таблиц
for /f "tokens=* delims=\r\n skip=1" %%t IN (tables.txt) do (
    REM Тут мы можем для отдельных таблиц добавить условия, чтобы, например, пропустить DROP
```

```
    echo DROP TABLE %%t; >> dump.sql
)
C:\Server\usr\local\mysql5\bin\mysql_run_to_import_dumps.exe -D ИМЯ_БД < dump.sql
del tables.txt
del dump.sql

manage.py syncdb --noinput
manage.py loaddata fixtures/data.json
```

Единственный минус — надо практически вручную вносить изменения в fixtures, но использовать fixtures — это завсегда полезная практика.

 **Lolka**, 16 декабря 2008 в 11:54 # **0**

Всегда пользовался django-evolution. Уже даже приспособился к некоторым его «особенностям». Было бы классно, если бы в самой джанге был полноценный мигратор, как в Ruby on Rails, к примеру.

 **mykir**, 5 января 2009 в 12:25 # **+1**

Автор!!! КОНКРЕТНОЕ БОЛЬШОЕ ЧЕЛОВЕЧЕСКОЕ СПАСИБИЩЕ!!!

п.с. начинающий питоно-дажнгер прогер столкнувшийся с такойже проблемой.

 **DmitryKoterov**, 29 ноября 2009 в 02:23 # **0**

Вообще, странно оно как-то… Джанге уже несколько лет от роду, а встроенного мигратора (хоть какого-то, хоть ручного) — нет, команды запуска консольной программы (типа python manage.py my_cron_script) — тоже нет… Неужели невостребовано? Надеюсь, в новых версиях появится.

 **Sergei_Erjemin**, 24 апреля 2013 в 16:17 # **0**

А у меня не заработало… Все установилось, вызов `./manage.py migrate` изменился, но, блин, новые или удаленные поля в модели не заставляют базу перестроится… Говорит в консолке: "./manage.py migrate"… и самое обидное, что это тоже не помогает.

Я, конечно, совсем чайник в Джанге, но хочется понять что не так делаю?

 **alekseyshavrak**, 5 сентября 2013 в 15:01 # **0**

Не обязательно сливать svn'ом код утилиты, можно проще через ssh — pip install south

МТС просит государство защитить себя от Skype

Как HTTPS обеспечивает безопасность соединения: что должен знать каждый Web-разработчик

Электромобиль Tesla S сломал оборудование для проведения краш-тестов