**Тема занятия № 8: Модуль 5. Миграции**

1. Генерирование миграций

Для генерирования миграций служит команда makemigrations утилиты manage.py:

Manage.py makemigrations [<список псевдонимов приложении, разделенных & пробелами>] [—name | -п <имя миграции>] [—noinput] [—no-input]

[—dry-run] [—check] [—merge] [—empty] [—no-header]

****

Если псевдонимы приложений не указаны, то будут обработаны модели из всех приложений проекта. Если указать псевдоним приложения или несколько псевдонимов, разделив их пробелами, будут обработаны только миграции из указанных приложений.

Команда makemigrations поддерживает следующие ключи:

□ —name или -п — указывает имя формируемой миграции, которое будет добавлено к порядковому номеру для получения полного имени файла с кодом миграции. Если оно отсутствует, то миграция получит имя по умолчанию;

□ —noinput или -no-input —отключает вывод на экран сведений о формируемой Миграции;

□ -dry-run— выводит на экран сведения о формируемой миграции, но не формирует ее;

□ —check— выводит сведения о том, изменились ли модели после последнего формирования миграций, но не формирует саму миграцию;

□ —merge —используется для устранения конфликтов между миграциями;

□ —empty— создает ’’пустую” миграцию для программирования ее вручную.

’’Пустая” миграция может пригодиться, например, для добавления какого-либо расширения в базу данных postgresql.

□ -no-header (начиная с Django 2.2) — указывает не вставлять в начало модуля Миграции комментарий с указанием версии Django и временной отметки генерирования миграции.

Результатом выполнения команды станет единственный файл миграции, выполняющий над базой данных все необходимые действия: создание, изменение и удаление таблиц, полей, индексов, связей и правил.

Внимание!

Django отслеживает любые изменения в коде моделей, даже те, которые не затрагивают структуры базы данных напрямую. Так, если мы укажем для поля название (параметр verbose name конструктора поля), фреймворк все равно создаст в миграции код, который изменит параметры нижележащего поля таблицы в базе данных. Поэтому крайне желательно продумывать структуру моделей заранее и впоследствии, по возможности, не менять ее.

Для отслеживания, какие миграции уже были выполнены, а какие — еще нет, Django создает в базе данных по умолчанию таблицу djangojnigrations. Править вручную как её структуру, так и хранящиеся в ней записи настоятельно не рекомендуется.

1. Файлы миграций

Все программные модули миграций сохраняются в пакете migrations, расположенном в пакете приложения. По умолчанию они получают имена формата последовательно увеличивающиеся порядковые номера>\_<имя миграции>.ру. Порядковые номера состоят из четырех цифр и просто помечают очередность, в которой формировались миграции. Имя миграции задается в ключе —name (-п) команды Makemigretions— если этот ключ не указан, то фреймворк сам сгенерирует имя следующего вида:

□ initial — если это начальная миграция, т. Е. Создающая самые первые версии всех необходимых структур в базе данных;

□ а\Ло\_<отметка даты и времени формирования миграции>— если это миграции, сформированные после начальной и дополняющие, удаляющие или изменяющие созданные ранее структуры.

Миграции можно переименовывать, но только в том случае, если они до этого ещё ни разу не выполнялись. Дело в том, что имена модулей миграций сохраняются в таблице django\_migrations, и если мы переименуем уже выполненную миграцию, то Django не сможет проверить, была ли она уже выполнена, и выполнит ее снова.

1. Выполнение миграций

Выполнение миграций запускается командой migrate утилиты manage.py:

Manage.py migrate [<псевдоним приложения> [<имя миграции>]]

[—fake-initial] [—noinput] [—no-input] [—fake] [—plan]



Если не указать ни псевдоним приложения, ни имя миграции, то будут выполнены все не выполненные к настоящему моменту миграции во всех приложениях проекта.

Если указать только псевдоним приложения, будут выполнены все миграции в этом приложении, а если дополнительно задать имя миграции, то будет выполнена только эта миграция.

Задавать имя модуля миграции полностью нет необходимости — достаточно записать только порядковый номер, находящийся в начале ее имени:

Manage.py migrate bboard 0001

В команде можно применить такие дополнительные ключи:

□ -fake-initial— пропускает выполнение начальной миграции. Применяется,

Если в базе данных на момент первого выполнения миграций уже присутствуют все необходимые структуры, и их нужно просто модифицировать;

□ —noinput или -no-input — отключает вывод на экран сведений о применении Миграций;

□ —fake — помечает миграции как выполненные, но не вносит никаких изменений в базу данных. Может пригодиться, если все необходимые изменения в базу были внесены вручную;

□ —plan (начиная с Django 2.2)— выводит план миграций— список, перечисляющий миграции в порядке их выполнения.

На каждую выполненную миграцию в таблицу django\_migrations базы данных добавляется отдельная запись, хранящая имя модуля миграции, имя приложения, в котором она была создана, дату и время выполнения миграции.

1. Слияние миграций

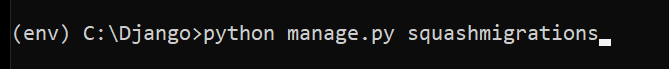
Если в модели неоднократно вносились изменения, после чего на их основе генерировались миграции, то таковых может накопиться довольно много. Чтобы уменьшить количество миграций и заодно ускорить их выполнение на ’’свежей” базе данных, рекомендуется осуществить слияние миграций— объединение их в одну.

Для этого достаточно отдать команду squashmigrations утилиты manage.py:

Manage.py squashmigrations <псевдоним приложения> [<имя первой миграции>]

<имя последней миграции> [—squashed\_name <имя результирующей миграции>]

[—no-optimize] [—noinput] [—no-input] [—no-header]

****

Обязательными ДЛЯ указания ЯВЛЯЮТСЯ ТОЛЬКО псевдоним приложения И имя последней миграции из числа подвергаемых слиянию. В этом случае будут обработаны все Миграции, начиная с самой первой из сформированных (обычно это начальная Миграция) и заканчивая указанной в команде. Пример:

Manage.py squashmigrations bboard 0004

Если задать имя первой миграции из подвергаемых слиянию, то будут обработаны Миграции, начиная с нее. Более ранние миграции будут пропущены. Пример:

Manage.py squashmigrations testapp 0002 0004

Рассмотрим ключи, поддерживаемые командой:

□ —squashed name — задает имя миграции, которая будет получена в результате слияния. Если оно отсутствует, то модуль результирующей миграции получит имя вида <имя первой миграции>\_вциавьед\_<имя последней миграции>.ру;

□ -no-optimize— отменяет оптимизацию кода миграции, что применяется для уменьшения его объема и повышения быстродействия. Рекомендуется указывать этот ключ, если слияние миграций выполнить не удалось или если результирующая миграция оказалась неработоспособной;

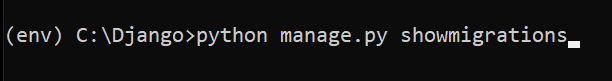
□ —noinput или -no-input — отключает вывод на экран сведений о слиянии миграций;

□ —no-header (начиная с Django 2.2) — предписывает не вставлять в начало модуля миграции, получаемой в результате слияния, комментарий с указанием версии Django и временной отметки слияния.

1. Вывод списка миграций

Чтобы просмотреть список всех миграций, имеющихся в проекте, следует отдать команду showmigrations утилиты manage.py:

Manage.py showmigrations [<список псевдонимов приложений, разделенных пробелами>] [—plan] [—р]



Если не указать псевдоним приложения, будут выведены все имеющиеся в проекте миграции с разбиением по приложениям. Если указать псевдоним приложения, то будут выведены только миграции из этого приложения. При задании списка псевдонимов приложений, разделенных пробелами, ВЫВОДЯТСЯ ТОЛЬКО миграции ИЗ ЭТИХ приложений, опять же, с разбиением по отдельным приложениям.

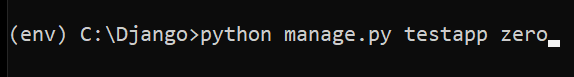
Список миграций при выводе сортируется в алфавитном порядке. Левее имени каждой миграции выводится значок [X], если миграция была выполнена, и [ ] —в противном случае.

Команда showmigrations поддерживает ключи —plan и -р, указывающие команде вместо списка вывести план миграций. План представляет собой список, отсортированный в последовательности, в которой Django будет выполнять миграции.

1. Отмена всех миграций

Наконец, Django позволяет нам отменить все миграции в приложении, тем самым удалив все созданные ими в базе данных структуры. Для этого достаточно выполнить команду migrate утилиты manage.py, указав в ней имя нужного приложения и zero в качестве имени миграции:

Manage.py migrate testapp zero



К сожалению, отменить отдельную, произвольно выбранную миграцию невозможно.