

# Миграции

## Создание первой (пустой) миграции.

После создания пустой миграции, в БД создастся таблица `alembic_version` в которой хранится идентификатор текущей версии.

```
alembic revision -m "Empty Init"
```

## Обновление базы данных после создания миграции

```
alembic upgrade head
```

## Хэш текущей миграции

```
alembic current
```

## История миграций.

Считываются все файлы миграций. Т.е. в случае загрузки предыдущего состояния, последующие миграции будут отображаться до тех пор, пока ненужные файлы миграций не будут удалены.

```
alembic history
```

## Возврат к предыдущему состоянию

```
alembic downgrade migration_id
```

Для возврата в стартовое состояние, нужно выполнить

```
alembic downgrade base
```

Если что-то пошло не так, для возврата в нулевое состояние нужно удалить из таблицы `alembic_version` текущий номер.

```
DELETE FROM public.alembic_version;
```

## Пропуск состояния

```
alembic stamp migration_id
```

## Экспорт в формате sql

```
alembic upgrade migration_id_start:migration_id_stop --sql > migration.sql
```

## Автогенерация миграции

```
alembic revision --autogenerate -m "Added Cookie model"
```

Поддерживаемые и неподдерживаемые действия при автоматической миграции

Тип элемента	Поддерживаемые	Неподдерживаемые
Таблицы	Добавление и удаление	Изменение имени
Столбец	Добавление, удаление, изменение нулевого статуса на столбце	Изменение имени
Индекс	Основные изменения в индексах и явно обозначенных уникальных ограничениях, поддержка автоматической генерации индексов и уникальных ограничений	
Ограничения		Ограничения без явного имени
Ключи	Переименование	
Типы		Типы, которые явно не поддерживаются базой данных

Чтобы alembic увидел класс данных, необходимо его непосредственно импортировать. Импорт всей директории не работает.

## Ручное создание миграций

На примере изменения имени таблицы

- Изменить имя.
- Выполнить генерацию миграции

```
alembic revision -m "Renaming table"
```

- В созданном файле изменить upgrade/downgrade

```
def upgrade():  
    op.rename_table('old_name', 'new_name')
```

```
def downgrade():
    op.rename_table('new_name', 'old_name')
```

- Выполнить миграцию

```
alembic upgrade head
```

### Команды alembic

add_column	Добавить столбец
alter_column	Изменить тип столбца, имя или значение по-умолчанию
create_check_constraint	Добавить ограничение
create_foreign_key	Добавить внешний ключ
create_index	Создать индекс
create_primary_key	Создать основной ключ
create_table	Создать таблицу
create_unique_constraint	Создать ограничение уникальности
drop_column	Удалить столбец
drop_constraint	Удалить ограничение
drop_index	Удалить индекс
drop_table	Удалить таблицу
execute	Выполнить сырую SQL команду
rename_table	Переименовать таблицу

Revision #6  
Created 26 July 2024 02:53:23 by Admin  
Updated 3 November 2024 15:28:36 by Admin