Урок 4. Миграции

Весь код по этой теме вы сможете изучить в GitLab.

В этом видео мы поговорим о миграциях БД. Выясним, что это такое и для чего они нужны. Разберём инструменты для накатывания миграций, рассмотрим их достоинства и недостатки.

Миграция — это процесс обновления структуры базы данных. Она используется для контроля версионности БД.

Добавление колонки в таблицу или удаление из неё, смена типа одного атрибута, удаление индекса — всё это меняет структуру базы. Если БД инициализирована ранее, то изменения в рамках ORM не отразятся на уровне базы данных. В этом случае нужно подключиться к базе и написать запросы на все эти обновления. Если у вас пять копий базы данных с актуальной версией, то нужно составить запросы на обновление для каждой из них.

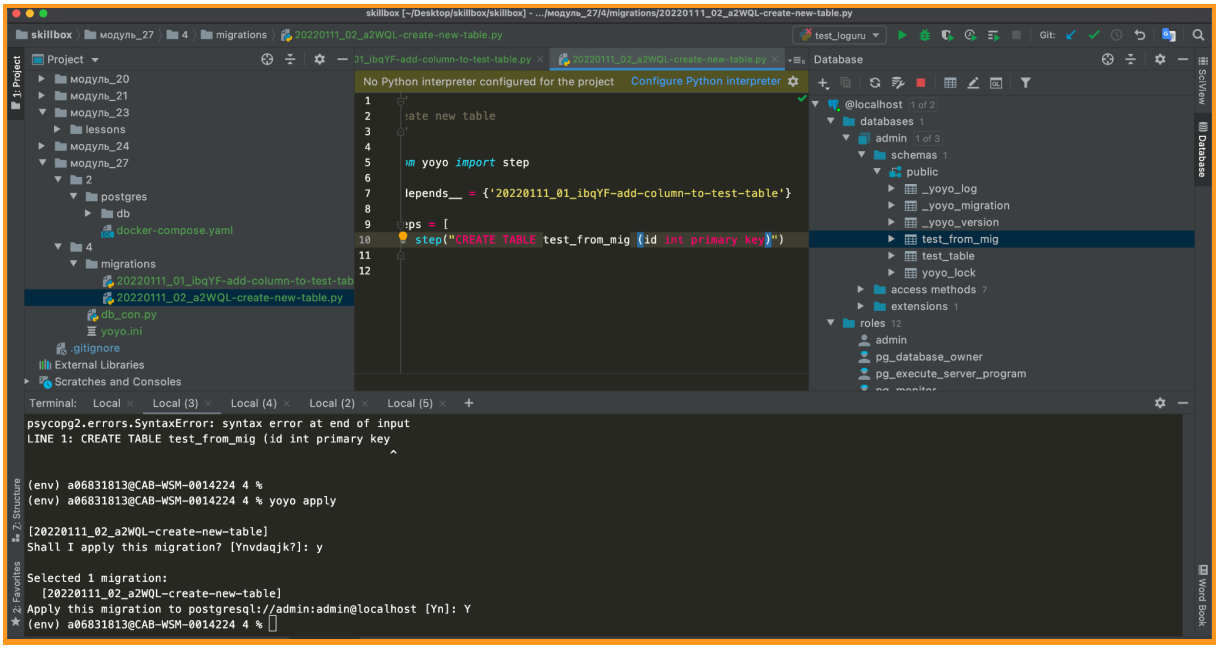
Такие проблемы легко решаются с помощью инструментов для миграции структуры базы данных. Основные принципы миграции БД:

- разовое выполнение каждого изменения: миграция либо выполнится полностью, либо не выполнится совсем;
- определённый порядок изменений;
- возможность вернуться к предыдущему состоянию. При добавлении поля и создании индекса, который содержит это поле, первое действие БД — выполнить одну команду ALTER TABLE, второе — создать поле, а потом индекс.

Первый инструмент, который мы разберём — yoyo-migrations. Он пригодится тем, кто не использует ORM, а пишет запросы БД вручную или при помощи сервисов. Например, некоторые разработчики специально отказываются от ORM, чтобы сэкономить ресурсы и увеличить производительность запросов.

В таких случаях удобно применять миграции, составленные вручную, — нужно не ориентироваться на модели ORM, а самостоятельно писать необходимые SQL-инструкции. Затем с помощью yoyo-migrations необходимо последовательно применить изменения структуры БД. В результате получается чистая, понятная и полностью управляемая структура таблиц.

Рассмотрим, как работает этот инструмент.



У этого решения существуют как преимущества, так и недостатки. Разберём их подробнее.

Преимущества:

- структурирование файлов с изменениями БД;
- фиксация миграций в отдельной таблице _yoyo_migration;
- выполнение каждой миграции в отдельной транзакции;
- проверка возможности выполнения миграции.

Недостатки:

- ручное описание запросов на чистом SQL.

Следующий инструмент — alembic.

Alembic — это инструмент миграции баз данных для использования с SQLAlchemy.

У этого решения существуют как преимущества, так и недостатки. Разберём их подробнее.

Преимущества:

- простое внедрение, реализован для SqlAlchemy;
- автоматическая генерация кода миграций на основе ORM-моделей;
- возможность использования python-функций внутри кода с миграциями;
- фиксация миграций в отдельной таблице alembic_version.

Недостатки:

- миграции не фиксируют изменения имён таблиц и колонок.

В этом видео мы познакомились с понятием миграции БД. Рассмотрели yoyo-migrations и alembic, на практике поработали с каждым инструментом и обсудили их достоинства и недостатки.

В этом модуле мы познакомились с СУБД PostgreSQL, проанализировали его работу относительно других СУБД, узнали структуру БД. Также мы запустили базу с помощью docker и docker-compose, изучили основные конфиги для настройки СУБД, поработали с командой psql. Рассмотрели синхронное и асинхронное подключение к базе, миграции структуры БД и полезные инструменты.