

Яндекс.Практикум

Миграции в PostgreSQL +  
интеграционное тестирование с БД

Яндекс Практикум

# Что нас сегодня ждет

- Обсудим миграции схем БД
- Поговорим про интеграционное тестирование слоя взаимодействия с данными
- Разберем ваши вопросы

# Миграции схемы БД

Схема БД – содержание + структура + ограничения целостности БД.

Схема может меняться вместе с системой, которая ее использует.

Миграции схемы БД – механизм, позволяющий:

- автоматизировать изменения схемы
- сделать их инкрементальными

Миграции позволяют:

- тестировать изменения схемы + ассоциировать их с изменениями кода
- автоматизировать upgrade и downgrade системы

# Инструменты миграций

Мы будем использовать <https://github.com/golang-migrate/migrate>

Другие инструменты:

- <https://github.com/rubenv/sql-migrate>
- <https://github.com/pressly/goose>

См. прочие опции здесь:

<https://awesome-go.com/database-schema-migration/>

# Интеграционные тесты

Интеграционное тестирование - тестирование сразу группы модулей. Зачастую интеграционные тесты могут иметь внешние зависимости (например, требовать подключения к БД).

Юнит-тестов не всегда может быть достаточно:

<https://tenor.com/view/integration-test-fail-doors-gif-11069607>

Спасибо за внимание!

Вопросы?

# etcd

etcd – распределенное KV-хранилище, использующее алгоритм Raft для достижения согласованности данных.

Raft – алгоритм для решения задачи консенсуса.

Описывает:

- процедуру выбора лидера
- репликацию

Хороший обзор Raft: <http://thesecretlivesofdata.com/raft/>

# etcd и CAP

etcd по умолчанию не дает гарантии линеаризуемости: узел отдает свое кэшированное состояние, которое может не совпадать с состоянием большинства. С этой точки зрения etcd может рассматриваться как AP

С другой стороны, можно запретить узлам сообщать состояние, если оно еще не согласовано. Это увеличит latency и сделает систему CP.

Более подробный разбор гарантий:

<https://aphyr.com/posts/316-jepsen-etcd-and-consul>