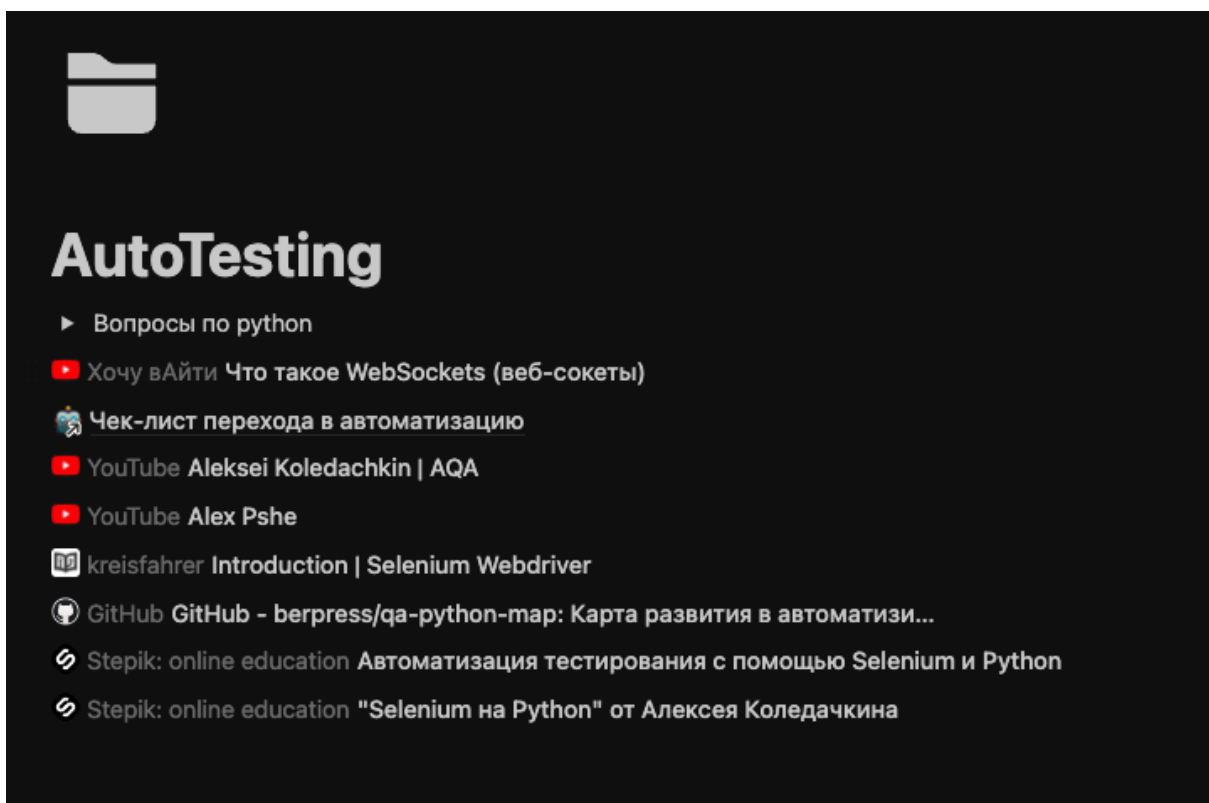
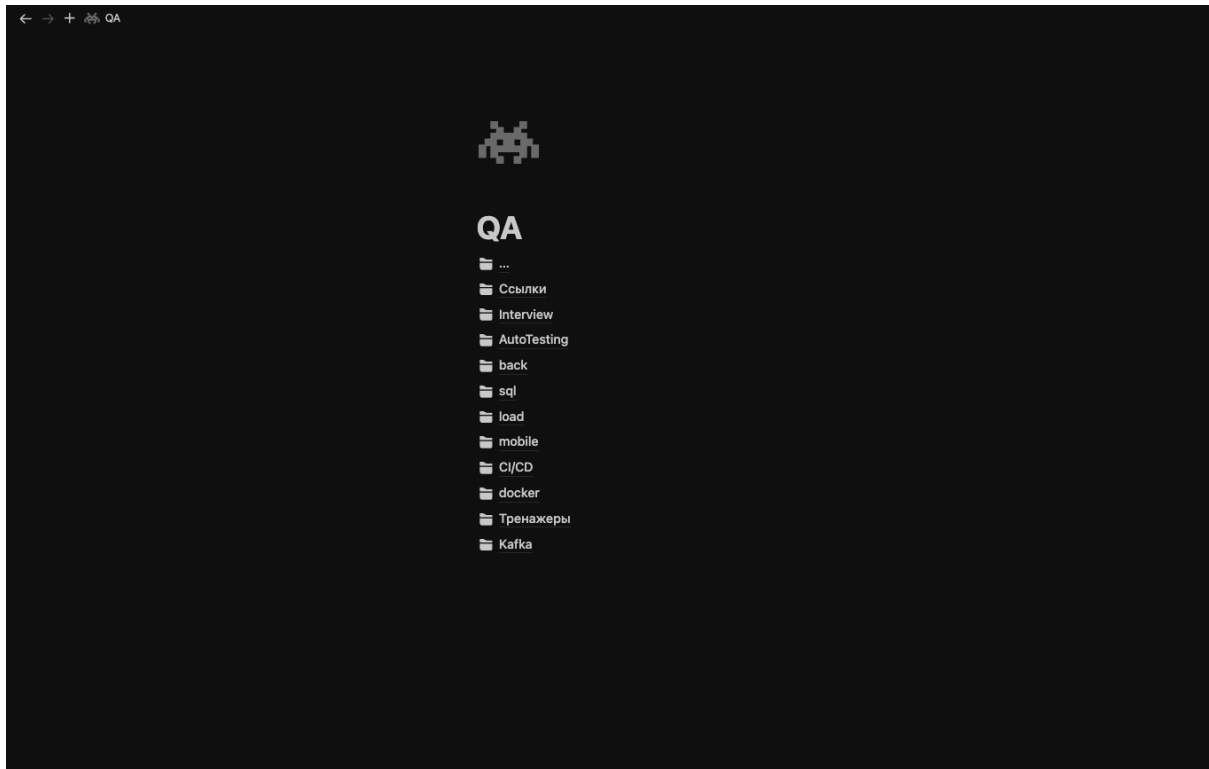


## Лабораторная работа №2 “База знаний”


Автор: Воложанин В.О.


ПО: notion





## back


 [t.me/qa\\_chillout](https://t.me/qa_chillout) Как тестировать gRPC


+  [t.me/qa\\_chillout](https://t.me/qa_chillout) Тестирование GraphQL



## sql

 Winderton ВСЁ про Базы Данных в одном месте / ХАЙЛАЙТЫ

 Vlad Mishustin Вся база SQL для начинающих за 1 час

 Симулятор SQL | [karpov.courses](https://karpov.courses)



## CI/CD

- ▶ Aleksei Koledachkin | AQA ПОСТРОЕНИЕ ПРОЦЕССА АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ С 0 ДО CI/CD ...
- ▶ Artsiom Rusau QA Life CI/CD для тестировщика + Jenkins
- ▶ PurpleSchool | Anton Larichev GitHub Actions для CI/CD - мини курс за 30 минут
- ▶ Oleh Pendrak Jenkins CI/CD | Автоматизация тестирования с нуля
- ▶ Podlodka Лайв-кодинг: GitLab CI as Quality Gates: CI глазами тестиров...
- ▶ Слёрм Вебинар «CI/CD: первый пайплайн и перспективы»




## docker


- ▶ PurpleSchool | Anton Larichev Docker: создание Dockerfile и оптимизация image
- ▶ Egor Repnikov Всё, что нужно знать о Docker | Уроки Docker
- ▶ Владилен Минин Docker для Начинающих - Полный Курс
- ▶ RomNero Docker с 0 до 100%. Всё, что нужно знать.
- 🔗 Stepik: online education Docker для начинающих + практический опыт
- 📺 Docker | karpov.courses





# Тренажеры

 QA Studio | Devtools-тренажёр

 BuildIn.AI

 QA Playground team QA Playground

+  Симулятор SQL | karpov.courses

 Peter Cottle Learn Git Branching

← → + QA / Kafka



## Kafka

### Resources

+ :: Resources

[Простыми словами про брокеров сообщений](#)

[Кратко](#)

[Клиент-серверная архитектура](#)

[Монолит vs Микросервисы](#)

[Монолит](#)

[Микросервисы](#)

[Передача данных между микросервисами](#)

[Проблемы взаимодействия по HTTP](#)

[1. Синхронность](#)

[2. Сложно масштабировать](#)

[3. Возможна потеря данных](#)

[Брокер сообщений \(Kafka\)](#)

[Преимущества использовать брокеры сообщений](#)

[1. Асинхронность](#)

[2. Легко масштабировать](#)

[3. Нет потери данных](#)

[Гарантии доставки сообщений в Kafka](#)

[At most once \(максимум 1 раз\)](#)

[At least once \(минимум 1 раз\)](#)

[Идемпотентность](#)

[Exactly-once \(конкретно 1 раз\)](#)

[Почему Kafka не теряет данные?](#)

[Topic](#)

[Partition](#)

[Зачем они нужны?](#)

[1. Быстрее потребляем информацию](#)

[2. Масштабирование + производительность](#)