

# Egor Dimitrenko's CV

“Мера интеллекта — способность меняться.” - Альберт Эйнштейн

☎ +7 993-928-5859

✉ [dimeqw@mail.ru](mailto:dimeqw@mail.ru)

🔗 [github.com/Shyyw1e](https://github.com/Shyyw1e)

## Языки и Фреймворки

- Golang (Go)
- Python
- C++

## Инструменты и Технологии

- Git
- GitHub Actions (CI/CD)
- Docker
- HTTP
- gRPC
- Postman
- GraphQL

## Дополнительно

- Английский язык (B2 Upper-Intermediate)
- Самоорганизация
- Работа в команде

## Образование

📖 Информатика и Вычислительная техника/ Компьютерные сети и системы

✍ МГТУ им. Баумана, Бакалавриат

📅 2023-2024

📖 Applied computer science

✍ МГТУ СТАНКИН, Бакалавриат

📅 2024-2028

## Обработка данных

- SQLite (SQL)
- REST API
- JSON
- Парсинг и обработка данных

## Концентрируюсь на

- Приобретение практического опыта
- Расширении набора инструментов и навыков
- Развитии soft skills

# Проекты

## URL-Shortener (Golang)

### Бизнес-задача:

Оптимизировать хранение, обработку и доступ к сокращённым URL.

Решение востребовано в маркетинге, аналитике и контент-менеджменте для удобства и отслеживания.

### Роль и вклад:

- **Разработка API** для обработки запросов на генерацию и перенаправление по сокращённым ссылкам.
- **Хранение данных в SQLite3**, что обеспечивает лёгкость интеграции, надёжность и автономность сервиса.
- **Реализация маршрутизации HTTP-запросов через Chi**, что позволило создать лаконичный и производительный сервер.
- **Автоматизация CI/CD**: GitHub Actions для автоматической сборки, тестирования и деплоя.
- **Тестирование и отладка**: провёл функциональные тесты API через Postman, обеспечил стабильность работы сервиса.

### Технический стек:

- **Язык программирования**: Go (Golang)
- **Фреймворк**: Chi (маршрутизация и обработка HTTP-запросов)
- **База данных**: SQLite3 (хранение связей коротких и оригинальных ссылок)
- **Инструменты**: Git, GitHub Actions (CI/CD), Postman (тестирование API)

### Результаты:

- Упрощение работы с ссылками и их распространения.
- Автоматизация CI/CD для ускорения развёртывания и поддержки.
- Масштабируется с интеграцией аналитики кликов и метрик.

## **Crypto cashflow (Python)**

### **Бизнес-задача:**

Упростить и ускорить процесс отслеживания финансовых транзакций и повысить прозрачность операций для команды трейдеров.  
Реализовать автоматическую выгрузку данных и развитие направления трейд-аналитики.

### **Роль и вклад:**

- **Разработка Python-скрипта** для автоматического сбора и анализа данных о транзакциях (ID, время, валюта, сумма, контрагенты, реквизиты).
- **Интеграция Google Sheets API** для мгновенной выгрузки результатов в защищённую таблицу, обеспечив доступ и возможность редактирования данных в режиме реального времени.
- **Настройка аутентификации через JWT** для снижения рисков несанкционированного доступа к конфиденциальной финансовой информации.
- **Тестирование и доработка продукта:** проведено и сведено более 30 реальных биржевых транзакций.

### **Технический стек:**

- **Язык и библиотеки:** Python, requests, logging, Google Sheets API
- **Безопасность:** JWT
- **Автоматизация:** CI/CD через GitHub Actions

### **Результаты:**

- Снижение времени, затрачиваемого на ручной свод данных до 60%.
- Выстраивание системы полуавтоматической бухгалтерии трейдера.
- Рост точности и скорости принятия решений трейдера.
- Возможность масштабирования для других трейдинговых платформ.

## Yells-post(Golang + GraphQL)

### Бизнес-задача:

Создать современную систему комментариев в стиле Reddit с поддержкой бесконечной вложенности, пользовательских переключателей обсуждений, обновлений в реальном времени (GraphQL-подписки) и пагинации.

### Роль и вклад:

- Проектирование и реализация GraphQL API с помощью gqlgen для CRUD-операций над постами и иерархическими комментариями.
- Внедрение подписок через WebSockets для доставки новых комментариев без опроса.
- Разработка двух стратегий хранения данных: in-memory (map + sync.Mutex) и в PostgreSQL (рекурсивная модель дерева комментариев).
- Написание unit-тестов для ключевых функций.
- Оптимизация работы: решение проблемы N+1 запросов и безопасная обработка глубокой рекурсии.

### Технический стек:

Go, gqlgen, PostgreSQL, Docker, goroutines, каналы, mutexes, пакет testing

 [github.com/Shyyw1e/yells-post](https://github.com/Shyyw1e/yells-post)

## grpc-file-server (Golang, gRPC)

### Бизнес-задача:

Разработать высокопроизводительный и масштабируемый сервис для загрузки, списка и скачивания файлов по gRPC с ограничением одновременных соединений для защиты от перегрузок.

### Роль и вклад:

- Реализация стриминга файлов (upload/download) через gRPC с дроблением на чанки.
- Построение семафоров для ограничения одновременных загрузок (10), скачиваний (10) и запросов списка (100).
- Хранение файлов в памяти и на диске (os, io).
- Создание CLI-клиента с флагами `upload`, `download`, `list`.
- Структурирование проекта по идиоматичному примеру Go (cmd/, internal/, proto/, client/).
- Обработка ошибок с использованием grpc/codes и внедрение структурированного логирования через slog.

### Технический стек:

Go 1.24.1, Protocol Buffers, gRPC, семафоры, файловый I/O, log/slog, пакет flag

### Результаты:

- Создана гибкая микро-сервисная архитектура на gRPC.
- Обеспечена стабильная обработка параллельных операций с троттлингом.
- Реализован лёгкий CLI-интерфейс.
- Проект готов к докеризации, внедрению TLS, аутентификации и облачным хранилищам (S3, MinIO).

 [github.com/Shyyw1e/grpc-file-server](https://github.com/Shyyw1e/grpc-file-server)