

# Титов Глеб

## Java Developer

+79136259505 ◇ [GitHub](#) ◇ [Telegram](#) ◇ [Gmail](#) ◇ Omsk

### SUMMARY

Начинающий Java-Backend разработчик с горящими глазами. Готов развиваться, изучать новые технологии и получать опыт.

### SKILLS

<b>Languages</b>	Java, SQL
<b>Frameworks</b>	Spring, Spring Boot, Spring Data, Junit5.
<b>Databases</b>	PostgreSQL, MySQL.
<b>Tools</b>	Docker, Git, Maven, Apache Kafka, Grafana, Prometheus, Liquibase.
<b>Addition</b>	Python, C, C++, C#

### EDUCATION

Омский государственный технический университет

Факультет информационных технологий и компьютерных систем

Направление Программная инженерия

Обучение 2022-2026

### LANGUAGES

Русский (native), Английский (B2)

### COURSES & CERTIFICATES

- Java Backend Academy от Тинькофф (Полугодичная программа, 2024–2025)
- Java Core/Multithreading/Servlets/Spring от JavaRush

### PROJECTS

#### • [Link Tracker](#)

Стек: Java 23, Spring Boot 3.4.2, Spring WebFlux, Spring Data JPA, PostgreSQL 17, Redis, Apache Kafka, Docker, Liquibase, microservices, Telegram Bot API, GitHub API, Stack Overflow API, Resilience4j, Bucket4j, Grafana, Prometheus

Описание: микросервисная система мониторинга веб-ссылок, состоящая из 3 сервисов: Bot Service, Scrapper Service, Report Service с интеграцией внешних API и системой аналитики.

- Реализовал паттерн для микросервисной архитектуры - разделение на Bot Service для взаимодействия с Telegram и Scrapper Service для мониторинга ссылок
- Для аутентификации пользователей использовал Telegram Bot API с командным интерфейсом управления ссылками

3. Реализовал сервис ScrapperService - для периодического мониторинга GitHub репозиториев и Stack Overflow вопросов с обнаружением изменений
4. Создал GitHub API интеграцию через GitHubClient для отслеживания коммитов, pull requests, issues в репозиториях
5. Реализовал Stack Overflow API интеграцию через StackOverflowClient для мониторинга новых ответов на вопросы
6. Настроил асинхронную обработку через Apache Kafka для отправки уведомлений между сервисами
7. Внедрил паттерны устойчивости: Circuit Breaker и Retry с Resilience4j для надежной работы с внешними API
8. Реализовал Rate Limiting с Bucket4j для защиты от перегрузок (10 запросов в секунду на IP)
9. Настроил мониторинг системы с Prometheus метриками и Grafana дашбордами для отслеживания производительности
10. Создал систему тегов и фильтров для категоризации отслеживаемых ссылок с поддержкой JPA связей
11. Настроил базу данных PostgreSQL с Liquibase миграциями и оптимизированным Hikari connection pool
12. Приложение полностью контейнеризовано с помощью Docker Compose с оркестрацией всех сервисов и зависимостей

- **Secured Forum API**

Стек: Java 21, Spring Boot 3.5.3, Spring Security, Spring Data JPA, PostgreSQL, JWT (JSON Web Tokens), OAuth2, Liquibase, Docker, TestContainers, MapStruct, Lombok

Описание: защищенный REST API сервис для форума с многоуровневой системой аутентификации и авторизации, поддержкой ролевой модели и интеграцией с внешними OAuth2 провайдерами.

1. Реализовал JWT-based аутентификацию - создал JWTService для генерации и валидации токенов с временем жизни 1 час
2. Для авторизации пользователей использовал Spring Security с кастомным JwtFilter и UserDetailsService
3. Реализовал сервис UserService - для регистрации, аутентификации и управления пользователями с ролевой системой
4. Создал ролевую модель доступа с сущностями User, Role и системой прав (USER, ADMIN)
5. Реализовал OAuth2 интеграцию для входа через внешних провайдеров (Google, GitHub)
6. Настроил защищенные эндпоинты для управления постами и комментариями с проверкой прав доступа
7. Внедрил MapStruct маппинг для преобразования между DTO и сущностями
8. Реализовал глобальную обработку исключений через GlobalExceptionHandler с информативными ошибками

9. Настроил базу данных PostgreSQL с Liquibase миграциями для управления схемой
10. Создал систему постов и комментариев с поддержкой CRUD операций и авторизацией
11. Внедрил TestContainers для интеграционного тестирования с изолированной БД
12. Приложение контейнеризовано с Docker Compose для простого развертывания

- **AI-Insight Dashboard**

Стек: Java 17, Spring Boot 3.x, Spring Security, Spring MVC, Thymeleaf, Bootstrap, JavaScript, Ollama API, Prometheus, OpenAPI/Swagger, Lombok, Maven

Описание: веб-приложение для генерации аналитических данных с использованием локальной нейросети, построенное на многослойной архитектуре с интеграцией AI-сервисов.

1. Реализовал многослойную архитектуру - разделение на Controller слой (InsightController), Service слой (InsightService, Neural ApiService) и Config слой (SecurityConfig, OpenApiConfig)
2. Для генерации аналитики использовал Ollama API с локальной нейросетью llama2 на порту 11434
3. Создал RESTful API с endpoints для получения аналитических данных по темам и проверки статуса сервиса
4. Реализовал веб-интерфейс с использованием Thymeleaf для серверного рендеринга и Bootstrap для адаптивного дизайна
5. Внедрил Spring Security для обеспечения безопасности API endpoints с защищенным доступом к метрикам
6. Настроил мониторинг системы через Prometheus с endpoint /actuator/prometheus для сбора метрик приложения
7. Создал OpenAPI документацию с автогенерируемой Swagger спецификацией для всех API endpoints
8. Реализовал обработку HTTP-запросов через Spring MVC с поддержкой JSON для обмена данными
9. Использовал @ConfigurationProperties для гибкой настройки подключения к Ollama API через application.yml
10. Применил Lombok для уменьшения шаблонного кода и повышения читаемости
11. Создал интерактивный frontend с JavaScript компонентами (InsightForm.js, HomePage.js) для динамического взаимодействия
12. Настроил Maven сборку с поддержкой Maven Wrapper для упрощения развертывания

- Другие проекты:

- 1) [Fractal flame](#) – Java-приложение для генерации фрактальных изображений с использованием алгоритма итерационных функциональных систем (IFS) с поддержкой многопоточности и различных типов трансформаций.
- 2) [Maze](#) – Java-консольное приложение для генерации лабиринтов (алгоритмы Прима и Краскала) и поиска пути в них (BFS, DFS, A\*) с отрисовкой в консоли.