

Бонкуров Магомед Русланович

Senior Java Developer / Ведущий Java-разработчик

Батуми, Грузия · удалённо, готов к релокации · +7 938 902-06-52 · antichit95@gmail.com · Telegram: @mnstrvu · bonkurov.ru

Senior Java-разработчик, 6 лет 5 месяцев коммерческой разработки на Java, почти 4 года из них — в грейде Senior. Работаю в EdTech и вообще с пользовательскими сервисами, где нагрузка приходит резкими пиками. 3 года 5 месяцев из этого — образовательная платформа «Сберкласс»: пришёл на неё со стороны подрядчика, потом перешёл в штат «Сберобразования».

Основной стек: Java 17, Spring Boot, PostgreSQL, Apache Kafka, Redis, Docker, Kubernetes, GitLab CI/CD.

Из того, что запомнилось лучше всего: выдачу задания ученику ускорил с 1,9 секунды до 260 мс по p95 — убрал N+1 и добавил составной индекс. Сервис обычно веду от начала до конца: схема данных, интеграции, выкатка, дежурства на проде. Плюс менторю джунов и мидлов и провожу технические собеседования.

КЛЮЧЕВЫЕ НАВЫКИ

Языки и платформы: Java 17 (records, sealed-классы, текстовые блоки), опыт с Java 8 и Java 11, включая миграцию 11 → 17; SQL, Linux.

Backend: Spring Boot, Spring Framework (Core, MVC, Data JPA, Security), Hibernate/JPA, JdbcTemplate, REST API, GraphQL (в том числе DataLoader), OpenAPI/Swagger, Keycloak, OAuth 2.0 / OIDC / JWT, Resilience4j.

Данные и интеграции: PostgreSQL — EXPLAIN (ANALYZE, BUFFERS), pg_stat_statements, составные индексы, устранение N+1, батчинг, декларативное партиционирование, materialized view; HikariCP, Flyway; Redis — cache-aside, TTL и инвалидация по событию; Apache Kafka — transactional outbox, at-least-once, идемпотентные консьюмеры.

Инфраструктура и тестирование: Docker, Docker Compose, Kubernetes — эксплуатация сервисов (deployment, configmap, secrets, Helm-чарты, kubectl), кластер администрирует платформенная команда; GitLab CI/CD, SonarQube, Maven, Gradle, Git; JUnit 5, Mockito, Testcontainers, JMeter; Grafana, Kibana/ELK, SLF4J + Logback. Практики: микросервисная архитектура, идемпотентность и обработка дублей, стратегии кэширования, работа с legacy, code review, менторство, Scrum.

ОПЫТ РАБОТЫ

Март 2025 — настоящее время (1 год 6 месяцев)

«Куб», Грозный (удалённо)

Ведущий Java-разработчик (Senior Java Developer), backend

«Куб» — разработка ПО и системная интеграция. Внутренняя платформа обработки заявок и начислений для корпоративных клиентов: приём заявки, маршрутизация по бизнес-правилам, расчёт начислений и обмен данными с внешними системами. В личной зоне ответственности — backend трёх сервисов с пиковой нагрузкой в рабочие часы.

- Заменил синхронный REST-обмен между сервисами на асинхронный через Apache Kafka: transactional outbox с поллером на стороне продюсера, идемпотентный консьюмер с дедупликацией по ключу события в таблице processed_events (запись в одной транзакции с бизнес-операцией, retention — удаление записей старше месяца). События перестали теряться при недоступности смежного сервиса; сознательно разменял строгую консистентность на eventual consistency, вариант с распределённой транзакцией отклонил из-за связности и стоимости поддержки.
- Перевёл таблицу событий, выросшую до сотен миллионов строк, на декларативное партиционирование по месяцам: исторические данные переносил батчами и подключал через ATTACH PARTITION без даунтайма, создание новых партиций и удаление старых автоматизировал. Аналитические выборки перестали конкурировать с OLTP.
- Вынес синхронный вызов внешней системы за Resilience4j (таймаут и circuit breaker): деградация смежного сервиса перестала занимать пул потоков и ронять эндпоинт целиком.
- Ввёл обязательное ревью двумя разработчиками для core-модулей и quality gate в GitLab CI/CD: сборка падает на критичных issue SonarQube и при снижении покрытия ниже порога, интеграционные тесты гоняются на Testcontainers.
- Менторю разработчиков уровня junior/middle: еженедельные встречи один на один (1:1), планы роста, разбор замечаний с ревью. Провожу технические интервью, написал чек-лист секции по Java и SQL.
- Эксплуатирую сервисы в Kubernetes (deployment, configmap, secrets, Helm-чарты, диагностика подов через kubectl), выкатываю dev/staging/prod в GitLab CI/CD; участвую в дежурствах по продакшену — после инцидента с исчерпанием пула соединений на пике добавил алерты на насыщение пула и лаг Kafka-консьюмера, написал постмортем, повторов не было.

Стек: Java 17, Spring Boot 3, Spring Data JPA, PostgreSQL, Apache Kafka, Redis, Kubernetes, Helm, Docker, GitLab CI/CD, SonarQube, REST API, Resilience4j, Flyway, JUnit 5, Mockito, Testcontainers, Grafana, Kibana, OpenAPI/Swagger, Gradle, SLF4J + Logback.

Октябрь 2022 — февраль 2025 (2 года 5 месяцев)

«Сберобразование» — продукт «Сберкласс» (удалённо)

Старший Java-разработчик (Senior Java Developer), backend

Перешёл из компании-подрядчика Naiku dev в продуктовую команду заказчика: тот же продукт, расширенная зона ответственности. Образовательная платформа для школьников с выраженным пиком в первые минуты урока. Владел модулем заданий и успеваемости целиком — от постановки с продуктом до выката и поддержки на проде.

- Карточка задания собиралась 300+ запросами (N+1 по вложенным сущностям); я заменил сборку на batch-выборку с join fetch и добавил составной индекс (student_id, task_id) — p95 снизился с 1,9 секунды до 260 мс. Замерял в JMeter, подтвердил в Grafana на проде в пиковые минуты.
- Провёл нагрузочное тестирование эндпоинтов выдачи заданий в JMeter: первым исчерпывался пул HikariCP на 10 соединений, следом упирались в синхронный вызов сервиса контента. Рассчитал размер пула от реального параллелизма с оглядкой на max_connections базы, вызов перевёл в асинхронный — пропускная способность выросла примерно вдвое при p95 в пределах SLA.
- Закэшировал профили и справочники в Redis (cache-aside; TTL как страховка на случай пропущенного события инвалидации, разброс TTL против одновременного истечения ключей на пике) — снял с PostgreSQL основную долю повторяющихся чтений в учебные часы.
- Разработал GraphQL-слой поверх Spring Boot: устранил N+1 в резолверах батчингом через DataLoader — число обращений к БД на сборку карточки упало с десятков до двух-трёх; ограничил глубину и сложность запроса, чтобы один клиентский запрос не мог положить сервис.
- Вынес агрегаты по успеваемости в materialized view с ночным REFRESH CONCURRENTLY (под него — уникальный индекс): чтение отчёта по классу сократилось с полутора минут до секунд, отчётность перестала конкурировать с OLTP.
- Реализовал REST API модулей заданий, успеваемости и результатов тестов, авторизацию вынес на Keycloak (OAuth 2.0 / OIDC / JWT) с ролевой моделью ученик/учитель/администратор, контракты описал в OpenAPI. Покрыл модуль интеграционными тестами на Testcontainers (реальные PostgreSQL и Kafka), вёл code review и онбординг двух разработчиков в модуль.

Стек: Java 11 → 17 (участвовал в миграции), Spring Boot 2.x, Spring Data JPA, JdbcTemplate, PostgreSQL, Apache Kafka, Redis, Keycloak (OAuth 2.0 / OIDC / JWT), GraphQL, REST API, Docker, Bitbucket, Bitbucket Pipelines, Maven, Gradle, Flyway, JUnit 5, Mockito, Testcontainers, JMeter, Grafana, Kibana, OpenAPI/Swagger.

Октябрь 2021 — сентябрь 2022 (1 год)

Naiku dev, Санкт-Петербург (удалённо)

Java-разработчик (Middle Java Developer), backend

Подрядная разработка на продукте «Сберкласс»: аутстафф-команда в модуле контента и заданий. По итогам года перешёл в штат заказчика на тот же продукт с расширением ответственности.

- Нашёл утечку памяти: самописный кэш на ConcurrentHashMap без ограничения размера и TTL удерживал объекты пользовательских сессий. Снял дамп по HeapDumpOnOutOfMemoryError, разобрал в Eclipse MAT — доминатором оказался этот кэш; заменил на Caffeine с maxSize и expireAfterWrite, добавил алерт на заполнение old gen. Сервис перестал уходить в OOM, регулярные перезапуски по расписанию отменили.
- Реализовал фичи модуля заданий целиком: REST-контроллеры и GraphQL-резолверы, миграции схемы, unit- и интеграционные тесты, выкатка на dev/staging, приёмка с аналитиком; вычистил нестабильные (flaky) тесты — прогон в CI перестал падать без изменений в коде.
- Разбирал дефекты с прода по логам в Kibana. Самый тяжёлый — двойная отправка ответа учеником создавала дубль результата: закрыл уникальным индексом по бизнес-ключу и идиоматичным ключом запроса, повторную отправку API возвращает как конфликт с актуальным состоянием.

Стек: Java 8/11, Spring Boot 2.x, Spring Data JPA, JDBC, PostgreSQL, GraphQL, REST API, Caffeine, Eclipse MAT, Maven, Bitbucket, JUnit 5, Mockito, Kibana, SLF4J + Logback.

Ноябрь 2019 — апрель 2021 (1 год 6 месяцев)

ChemCool, Грозный · штат (удалённо)

Java-разработчик (Junior → Middle), backend

Образовательный портал для учащихся 8–11 классов: тесты и проверка знаний. Небольшая команда, продукт работал на проде.

- Реализовал backend клиентского сервиса на Spring Boot: каталог тем, прохождение теста, выдача результатов; доступ к PostgreSQL через Spring Data JPA и Hibernate, контракты описал в OpenAPI/Swagger.
- Настроил обмен событиями между сервисами через Apache Kafka: результаты теста уходили в сервис статистики асинхронно — пересчёт статистики перестал блокировать ответ пользователю.
- Настроил аутентификацию и авторизацию на Spring Security: вход по JWT, роли ученик/преподаватель, запрет доступа к чужим результатам. Покрыл свои модули unit- и интеграционными тестами, вёл README по модулям; правил клиентскую часть на React в объёме сквозной отладки своих фич.

Стек: Java 8, Spring Boot 2.x, Spring MVC, Spring Data JPA, Spring Security, Hibernate/JPA, PostgreSQL, Apache Kafka, REST API, Maven, Git, JUnit 5, Mockito, OpenAPI/Swagger, SLF4J + Logback; базовый фронт — React.

ОБРАЗОВАНИЕ И РАЗВИТИЕ

Среднее профессиональное, 2017 — Чеченский индустриальный техникум, Грозный. Специальность: оператор ЭВМ.
2017–2019 — работал и параллельно учился на программиста, проходил стажировку разработчиком; самостоятельно изучал Java Core, Spring и SQL.

ЯЗЫКИ

Русский — родной. Английский — уверенный B1: говорю и пишу, техническую документацию и исходники читаю без словаря, веду рабочую переписку и code review на английском, разбираю англоязычные логи и стектрейсы.