**Акчурин Руслан** **Java Developer | Senior Software Engineer, более 10 лет коммерческого опыта** **Локация: Москва**

**Основной стек технологий** **Языки**: Java (Java 8, 11, 17), Kotlin, Groovy, SQL, Python (для вспомогательных задач)  
 **Фреймворки**: Spring Framework (Spring Boot, Spring Security, Spring Data, Spring Cloud), Hibernate, Micronaut  
 **Базы данных**: PostgreSQL, Oracle, MySQL, MongoDB, Redis, Cassandra, Elasticsearch  
 **Сообщения и интеграции**: Kafka, RabbitMQ, ActiveMQ, JMS, gRPC, SOAP, REST API, GraphQL  
 **Инструменты DevOps**: Docker, Kubernetes, Jenkins, GitLab CI/CD, Ansible  
 **Методологии**: Микросервисы, DDD, TDD, BDD, Event Sourcing, Асинхронное программирование, Многопоточность, Reactive Programming  
 **Прочее**: OpenShift, Maven, Gradle, TestContainers, Swagger/OpenAPI, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana), Liquibase, Selenium, RestAssured

**Образование** Московский государственный университет путей сообщения  
 Специальность: Системы автоматизированного проектирования | Инженер | 2005

**Знание английского языка** Английский: B1

**Кратко о себе** Java-разработчик с более чем 10-летним опытом в создании сложных энтерпрайз-решений. Специализируюсь на разработке высоконагруженных микросервисных систем, использовании современных фреймворков (Spring, Spring Boot, Kafka), а также внедрении процессов CI/CD. Обладаю опытом работы с многопоточностью, асинхронными вычислениями и оптимизацией производительности.

**Опыт работы**

**GetMeGit (проект Сбер) (Сентябрь 2023 – настоящее время)** **Руководитель направления** Сфера: Fintech  
 **Описание проекта** Giga IDE — корпоративная среда разработки для внутренних пользователей Сбербанка, позволяющая вести совместную работу над кодом, управлять зависимостями и автоматизировать тестирование. Система консолидирует все необходимые инструменты для разработчиков, снижая издержки на настройку и поддержку окружения. Проект обеспечивает стандартизованный процесс деплоя и тестирования, повышая качество и скорость выпуска продуктов банка.  
 **Стек проекта** Java 17, Spring Boot, Kafka, Gradle, Groovy, Jenkins, Ansible, OpenShift, IntelliJ Platform SDK, Bash  
 **Состав команды проекта** 6 разработчиков, 2 DevOps-инженера, 1 аналитик  
 **Задачи**

* Руководство разработкой и интеграция Giga IDE с корпоративными сервисами
* Управление архитектурными решениями и стандартами кодирования
* Настройка и оптимизация сложных CI/CD пайплайнов с помощью Jenkins и Ansible
* Разработка сценариев асинхронного взаимодействия между компонентами через Kafka  
   **Достижения/результаты**
* Внедрил расширенные возможности IDE для создания и тестирования микросервисов
* Сократил время деплоя за счет оптимизации Ansible-скриптов
* Обеспечил консолидацию внутренних библиотек и шаблонов в единой среде разработки

**NDA (Банк) (Сентябрь 2022 – Сентябрь 2023)** **Главный инженер по разработке (тимлид)** Сфера: Fintech  
 **Описание проекта** АС «Карта жителя инфообмен» — это комплексная система, обеспечивающая цифровое управление картами через мобильное приложение и веб-сервисы. Первый модуль системы отвечает за обработку транзакций через ЕФС 19.5, а второй — за взаимодействие с внешними сервисами через ППРБ. Проект ориентирован на микросервисную архитектуру, позволяющую адаптироваться к постоянно растущему числу пользователей и интеграций.  
 **Стек проекта** Java 17, Spring Boot, Spring Security, REST, SOAP, gRPC, Kafka, PostgreSQL, Swagger, Maven, Liquibase, Docker, Kubernetes  
 **Состав команды проекта** 8 разработчиков, 2 тестировщика, 1 аналитик, 1 DevOps-инженер  
 **Задачи**

* Разработка микросервисной архитектуры для обеспечения гибкости и масштабируемости
* Построение интеграционных решений через REST, SOAP и gRPC
* Настройка Docker/Kubernetes-окружения для автоматизированного развертывания
* Управление производительностью с помощью многопоточности и оптимизированных запросов к PostgreSQL
* Контроль качества кода, проведение код-ревью и наставничество команды  
   **Достижения/результаты**
* Успешно перевел систему с монолита на микросервисы, сохранив стабильность работы
* Организовал централизованную документацию REST API с помощью Swagger
* Укрепил безопасность сервисов и оптимизировал конфигурацию Kafka для обработки событий

**Инфосистемы Джет (Май 2019 – Сентябрь 2022)** **Java Developer** Сфера: Fintech  
 **Описание проекта** Сбербанк Бизнес Онлайн (СБОЛ) — масштабная система дистанционного банковского обслуживания для корпоративных клиентов, предоставляющая инструменты для управления платежами, уведомлениями и отчетностью. Микросервисная архитектура обеспечивает высокую доступность и быстрый выпуск новых функций. В проекте уделялось повышенное внимание информационной безопасности и оптимизации производительности при пиковых нагрузках.  
 **Стек проекта** Java 11, Spring Boot, PostgreSQL, Kafka, Jenkins, Groovy, Liquibase, RabbitMQ, JUnit, Mockito, Selenium, Ansible  
 **Состав команды проекта** 10 разработчиков, 3 тестировщика, 2 аналитика  
 **Задачи**

* Создание и поддержка микросервисов на Java 11 с использованием Spring Boot
* Разработка системы уведомлений через RabbitMQ и JMS
* Настройка Kafka-кластеров для асинхронной обработки событий
* Автоматизация тестирования с помощью JUnit, Mockito и Selenium
* Подготовка GitLab CI/CD пайплайна для регулярных деплоев  
   **Достижения/результаты**
* Внедрил дополнительные очереди в RabbitMQ для повышения отказоустойчивости
* Улучшил производительность PostgreSQL за счет оптимизации сложных запросов
* Настроил систему мониторинга микросервисов для оперативного выявления проблем
* Сократил время ручного тестирования, расширив покрытие автотестами

**Световые Технологии (Декабрь 2018 – Апрель 2019)  
Java Developer** Сфера: Industrial  
 **Описание проекта** Проект охватывал разработку и внедрение систем управления промышленным освещением, включая сбор телеметрии с IoT-устройств и анализ больших массивов данных. Система обеспечивала централизованный контроль за параметрами освещения и позволяла в реальном времени изменять настройки для повышения энергоэффективности. Важная часть проекта — надежная и быстрая передача сообщений от большого числа сенсоров.  
 **Стек проекта** Java, Spring Boot, JMS, JSON, Kafka  
 **Состав команды проекта** 3 разработчика, 1 DevOps-инженер  
 **Задачи**

* Разработка серверного приложения для управления IoT-устройствами на Spring Boot
* Настройка Kafka и JMS для асинхронной обработки крупных потоков данных
* Интеграция с облачными сервисами для аналитики собранной статистики
* Создание механизмов реактивного управления освещением в режиме реального времени  
   **Достижения/результаты**
* Разработал архитектуру, позволяющую автоматически обрабатывать поступающие события от сотен сенсоров
* Упростил мониторинг системы за счет детальной логики распределения сообщений
* Ввел дополнительный уровень кеширования для ускорения анализа данных

**Сбербанк-Технологии (Декабрь 2017 – Май 2018)  
Java Developer** Сфера: Fintech  
 **Описание проекта** Платформа для обработки кредитных заявок (Pega BPM) автоматизирует весь цикл рассмотрения кредитов, от подачи до одобрения. Важной частью было глубокое взаимодействие с внешними сервисами, позволяющее быстро проверять заемщиков и оценивать риски. Использование Pega BPM и GridGain обеспечивало высокую производительность при обработке большого объема параллельных заявок.  
 **Стек проекта** Java, Spring, Pega BPM, BPMN, JPA, Git, Maven, Jenkins, GridGain  
 **Состав команды проекта** 5 разработчиков, 2 аналитика  
 **Задачи**

* Разработка бизнес-логики для кредитных сценариев в Pega BPM
* Интеграция с внешними SOAP и REST сервисами для проверки клиентов
* Настройка GridGain для ускорения обработки больших потоков данных
* Поддержка CI/CD процессов через Jenkins и Maven  
   **Достижения/результаты**
* Снизил время задержки при обработке заявок благодаря оптимизации GridGain
* Реализовал стабильную интеграцию с сервисами проверки заемщиков
* Улучшил общую архитектуру за счет более гибкой конфигурации процессов BPM

**Леруа Мерлен (Май 2015 – Ноябрь 2017)** **Java Developer** Сфера: Retail  
 **Описание проекта** Проект по внедрению микросервисной архитектуры для мобильных и веб-приложений, позволяющей клиентам просматривать ассортимент, проверять наличие товаров и управлять заказами. Важным направлением была интеграция с внутренними системами складского учета, чтобы обеспечить точность данных о наличии товара в режиме реального времени. Система также включала инструменты персонализации и систему рекомендаций для улучшения пользовательского опыта.  
 **Стек проекта** Spring Cloud, Spring Boot, REST, MongoDB, JPA, Docker, Maven, Jenkins  
 **Состав команды проекта** 6 разработчиков, 2 тестировщика, 1 DevOps-инженер  
 **Задачи**

* Разработка микросервисов для мобильных и веб-приложений
* Настройка REST API для интеграции с системами управления складом
* Автоматизация сборки и развертывания через Docker и Maven
* Создание модулей личного кабинета с учетом персонализации  
   **Достижения/результаты**
* Оптимизировал скорость обновления каталога благодаря асинхронной обработке запросов
* Упростил процедуру релизов за счет внедрения пайплайнов в Jenkins
* Обеспечил корректную работу сервисов в условиях высокой нагрузки (акции и распродажи)

**WINLINE (Май 2013 – Апрель 2015)  
Java Developer** Сфера: Gambling/Game Development

**Описание проекта**WINLINE — онлайн-платформа для спортивных ставок, обеспечивающая прием и обработку бетов в режиме реального времени. В рамках проекта интегрированы различные платежные системы, а также внешние сервисы для получения спортивных данных и коэффициентов. Основная цель — поддерживать высокую надежность и производительность платформы во время пиковых нагрузок, связанных с крупными спортивными событиями  
 **Стек проекта** Java 8, Java Core, Java FX 8, JavaScript (Nashorn)  
 **Состав команды проекта** 4 разработчика, 1 DevOps-инженер  
**Задачи**

* Разработка и поддержка кассового модуля для приема и учета ставок
* Интеграция с внешними провайдерами спортивных данных для обновления коэффициентов
* Настройка Java FX 8 для реализации клиентского интерфейса внутреннего терминала
* Оптимизация кода, включая поиск и устранение узких мест в логике приема платежей
* Настройка Nashorn для серверного выполнения JavaScript-скриптов при обработке специальных правил ставок  
   **Достижения/результаты**
* Ускорил обработку ставок за счет оптимизации взаимодействия с базой данных
* Повысил стабильность системы во время пиковых спортивных событий, внедрив мониторинг и алертинг
* Обеспечил гибкую конфигурацию для подключения новых платежных сервисов и обновления спортивных коэффициентов