



Павел Прохоренко

✉ job@mankey.ru

🔗 linkedin.com/in/pavel-prokhorenko

Старший/Ведущий инженер-программист, более 10 лет в высоконагруженных/высокодоступных корпоративных системах.

Технические навыки

- **Фронтенд:** Vue.js v3 и v2 (с 2016), Vuex, Vue Router, TypeScript, SCSS, Less, TipTap, ProseMirror, D3.js, Chart.js, Vuetify, Quasar
- **Бэкенд:** Node.js (с 2017), PostgreSQL/Oracle, Elasticsearch, Kafka, Redis, Express.js/Nest.js
- **Десктоп:** Electron (с 2017)
- **Инфраструктура и доставка:** Vite, Rollup, Webpack, Docker, GitLab CI/CD, GitHub Actions
- **Тестирование:** Jest, Cypress (модульное, компонентное, E2E);
- **Прочее:** Системы на основе метаданных, микрофронтенды, публикация NPM пакетов, модернизация легаси, инструменты ИИ и интеграция LLM, миграции на TypeScript;

Опыт работы

Прошел путь от младшего до старшего/ведущего разработчика в ОАО «РЖД» (RZD.ru), ноябрь 2011 – апрель 2026

- Фронтенд
 - ▶ Высоконагруженная платформа продажи билетов
 - ▶ UARM-FE: Панель администратора РЖД
 - ▶ wFlow: Приложение для управления проектами в стиле Канбан
 - ▶ МСС: менеджер развертывания/доставки
- Бэкенд
 - ▶ Адаптер API CSM
 - ▶ Информеры версий (IVI + RVI)
- Архитектура
- Инфраструктура
- Тестирование
- NPM пакеты
- Покрытие TypeScript
- Десктоп
- Использование ИИ
- Образование
- Языковые навыки
- Местоположение
- Мои приоритеты

Фронтенд

Создал более 30 фронтенд-приложений исключительно на современном стеке:

- Vue.js v2 и v3, Vuex, Vue Router;
- Typescript;
- D3.js, Chart.js;
- Vuetify, Quasar;
- Scss, Less;

Миграция высоконагруженной платформы продажи билетов

Масштаб: более 20 миллионов посещений в месяц, ~150 тыс. *оплаченных* билетов в день.

Критически важная миграция фронтенда с собственного фреймворка на основе jQuery на Vue.js

- Результат: одно SPA вместо множества страниц. Примерно в 3 раза меньше трафика и нагрузки на сервер, улучшение производительности, стабильности, UX, удобства сопровождения, скорости поставки, а также современный технологический стек на будущее
- Обработанная сложность:
 - Бизнес-логика реализована на фронтенде с помощью JS и шаблонов сервера на основе [метаданных](#);
 - Тысячи станций в 11 часовых поясах; междугородние, международные и пригородные поезда и автобусы;
 - Десятки дополнительных сервисов; поддержка 3 языков;
 - Сотни сценариев;

UARM-FE: Фронтенд-приложение панели администратора РЖД

Современная замена устаревшей панели администратора

- До: JSP-приложение с состоянием на IBM WebSphere (создано Java-разработчиками);
- После: Современное асинхронное приложение на Vue.js;
- Архитектура: Динамический рендеринг компонентов на основе декларативных [метаданных](#);
- Результат: Значительно улучшена производительность и опыт разработки
- Внедрение: Более 1000 пользователей;
- Сложный [конвейер сборки](#);

wFlow: Приложение для управления проектами в стиле Канбан

Фронтенд для управления проектами корпоративного уровня

Внедрение и масштаб:

- Принято в масштабах компании: более 70 пользователей перешли с Jira;
- Развернуто в Российском университете транспорта: более 200 пользователей;

Возможности:

- Канбан-доски с перетаскиванием;
- Настраиваемые конфигурации рабочих процессов;
- Настраиваемые поля задач;
- Связывание задач и вложения файлов;
- Собственный WYSIWYG JSON-редактор (на базе TipTap/ProseMirror);

[Узнать больше о wFlow \[RU\]](#)

МСС: менеджер развертывания/доставки

МСС расшифровывается как Центр управления полетами (Mission Control Center)

Внутреннее приложение, отвечающее за CI/CD: развертывание образов контейнеров, запуск и проверку результатов автоматизированных тестов, выпуск дистрибутивов ПО, проверку и утверждение запросов на слияние (MR), а также генерацию документации по выпускам совместно с двумя [сервисами Node.js](#), упомянутыми ниже. Это критически важная часть процесса доставки ПО компании, используемая несколькими командами в организации.

Бэкенд

Имею сильный опыт в бэкенд-разработке и управлении [инфраструктурой](#).

Разработал более 7 сервисов на Node.js с использованием:

- Elasticsearch (индексация, поиск, логирование);
- Apache Kafka;
- PostgreSQL;
- Redis;
- Express.js / Axios;
- Стратегии "повтор/откат";
- Аутентификация JWT и LTPA;
- Логирование;
- Генерация DOCx и XLSx;
- Интеграции с различными сторонними API: JSON и XML;
- Все компоненты [докеризированы](#)

Адаптер API CSM

Критически важный бэкенд-сервис на Node.js для РЖД (Центр содействия мобильности пассажиров). Хаб для обработки четырех различных типов заявок от пассажиров с ограниченными возможностями со сложными моделями данных; интегрирует следующие системы:

- Сайт РЖД <https://www.rzd.ru/ru/9290> — Kafka
- Административные интерфейсы РЖД — Kafka
- АСУ ЕКМП (Единое клиентское мобильное приложение) — JSON REST API
- АСУ ТТК (ТрансТелеКом) — JSON REST API
- АСУ СИОП ФПК (Система идентификации и обслуживания пассажиров, Федеральная пассажирская компания) — JSON REST API

Двунаправленная синхронизация на протяжении всего жизненного цикла приложения, уведомления пользователей и администраторов и многое другое.

Информеры версий (IVI + RVI)

Два бэкенд-сервиса: IVI (на стороне компании) и RVI (развернут на сервере заказчика — РЖД). Сервисы полностью автоматизируют жизненный цикл выпуска ПО — от принятия запроса на

слияние до публикации релиза и установки по цепочке серверных сред (от тестовых до промышленной).

Ключевые функции и обязанности:

- Разработал и поддерживал интеграции с: Wflow (трекер задач), MCC (менеджер установки), RVI, PostgreSQL, FTP, SVN, GitLab, Mattermost и Kibana.
- Реализовал структурированное логирование в Elasticsearch с мониторингом в реальном времени.
- Настроил интеллектуальные уведомления в Mattermost по нескольким каналам, включая ссылки на формы повторного запуска, логи Kibana и сведения об ошибках.
- Обеспечил высокое покрытие тестами с помощью комплексных end-to-end и модульных тестов (Jest).
- Обработчик принятия MR: Проверяет поля задачи в трекере задач, ищет дубликаты, собирает данные контракта/эпика, определяет измененные репозитории, записывает данные в базу данных IVI и передает их в RVI через REST API.
- Обработчик публикации релиза: Обнаруживает "потерянные" задачи, загружает и распаковывает дистрибутив, генерирует артефакты для заказчика (6 типов документов) и JSON состава релиза для надежной передачи данных в RVI (запасной вариант при недоступности REST API).
- Обработчик событий установки: Фиксирует успешные установки/откаты на сервере, выполняет массовое обновление статусов релизов и связанных задач через Bulker API, добавляет комментарии об откате.

Архитектура

Разработал различные решения на многодоменной основе: бэкенд (уровень приложений + уровень БД) и фронтенд.

Капитальный редизайн национального портала RZD.RU (РЖД)

Руководил полным технологическим редизайном и миграцией национального портала RZD.RU в рамках масштабного перехода с устаревшего стека Oracle + IBM WebSphere на современную архитектуру на базе Postgres, Kubernetes и Docker.

Масштаб проекта: ~1000 пользовательских страниц на ~50 поддоменах и ~500 редакторских CRUD-интерфейсов / рабочих мест администратора.

Ключевой вклад:

- Архитектура: Разработал архитектуру фронтенда и тесно сотрудничал с Java-архитекторами над новыми форматами метаданных, что позволило реализовать современный [headless metadata-driven CMS \[RU\]](#) подход.
- Миграция метаданных: Организовал миграцию более 1500 устаревших XML-файлов (сильно обремененных техническим долгом и обходными решениями) в чистые, поддерживаемые JSON-структуры. Разработал скрипты преобразования и создал [MetaWizard](#) — мощный редактор метаданных, ставший центральным инструментом для всей миграции и текущего обслуживания.
- Пользовательский портал: Реализовал базовые фронтенд-решения, включая новую тему сайта и полный редизайн наиболее критичных и сложных страниц. Эта основа позволила остальной команде фронтенда эффективно создавать оставшийся функционал.

- Редакторские интерфейсы: Разработал и создал новое административное приложение ([UARM-FE](#)), предоставив большинство элементов управления UI, а также несколько наиболее сложных редакторских интерфейсов.

Десктоп

Приложение MetaWizard на Electron

Сложное внутреннее десктопное приложение (Windows/Mac/Linux).

- Разработано и создано в 2017 году;
- Поддерживалось в течение 9 лет, при этом обновлялось до последних версий Electron и Node, начиная с версий 1.x и 0.x соответственно

Ключевые возможности:

- Редакторы и списки [метаданных](#);
- Подключение к базам данных (Oracle, PostgreSQL);
- Операции SSH/SFTP;
- Сложные файловые операции;
- Многорепозиторная структура с интегрированным [приложением панели администратора](#);

Инфраструктура

Разработал и поддерживал сложные полные циклы конвейеров для фронтенда и бэкенда:

- **Сборка:** Vite и Webpack (включая Code-Splitting/Module Federation/DLL), Babel, tsc/tsx, SCSS/Less, ESLint, Prettier, Husky, Lint-Staged и т.д.
- **Доставка:** Docker, GitLab CI/CD, GitHub Actions

Тестирование

Внедрил тестирование для нескольких проектов (фронтенд, бэкенд, десктоп):

- Модульное, компонентное и E2E
- Jest и Cypress

NPM пакеты

- Опубликовал более 10 пользовательских пакетов в частный реестр компании
- Фронтенд и бэкенд, используемые в нескольких проектах компании:
 - Библиотеки для фронтенда/бэкенда для аутентификации в различных системах (JWT, Blitz, LTPA)
 - Унифицированные (стандарт компании) библиотеки для:
 - Обработки ошибок
 - Логирования (ELK)
 - Интеграции с Kafka
 - Уведомлений Mattermost
 - Утилиты для фронтенда и бэкенда для общих задач

- Общие конфигурации для ESLint, Prettier и т.д.
- Библиотеки для фронтенда/бэкенда для интеграции с различными системами
- Обеспечил [различные конвейеры сборки и CI/CD](#)

Покрытие TypeScript

Руководил миграциями на TypeScript для нескольких проектов, улучшая качество кода и удобство сопровождения, сокращая ошибки времени выполнения и улучшая опыт разработки. Обычно поэтапно, с постепенной миграцией и строгим соблюдением типизации.

Использование ИИ

Использую различные LLM с разными целями и подходами.

Поддерживаю баланс между контролем кода и производительностью разработки.

Когда это применимо — практикую агентные рабочие процессы и разработку на основе спецификаций (Openspec).

Для других целей — конечно: предложения, конфигурация, документация, отладка, тестирование, быстрое прототипирование, и т.д.

Образование

Степень бакалавра по инженерной геологии

Российский университет дружбы народов (РУДН), Москва, Россия, 2005 – 2009

Языковые навыки

Свободный английский, родной русский, начальный немецкий, французский и украинский.

Местоположение

На удаленной работе с 2013 года, сейчас из Таиланда. *Очень* гибок в отношении часовых поясов (до GMT+1).

Открыт к гибридной работе при необходимости, а также к переезду/работе в офисе.

Мои приоритеты

- Командная работа — Сотрудничество, постоянное обучение и обмен знаниями с другими в качестве наставничества и лидерства
- Четкий дизайн — Архитектура, задачи, код и т.д.
- Создание решений — Либо поддерживаемые приложения на годы вперед, либо быстрые PoC/MVP
- Баланс — Бизнес-цели (первый приоритет), опыт разработчика и управление техническим долгом