

Гильмуллин Руслан

+7 *** ** * | mellonis@yandex.ru | linkedin.com/in/mellonis | github.com/mellonis | mellonis.ru | Москва, Россия

ПРОФЕССИОНАЛЬНОЕ РЕЗЮМЕ

Senior Frontend Software Engineer с фулстек-опытом, фокус на TypeScript и интерактивный фронтенд, ~10 лет в банке/финтехе (платформы интернет-банкинга, KYC, платёжные флоу, финансовые калькуляторы). Клиент - React, Next.js. Сервер - Fastify, NestJS, Apollo GraphQL. Auth - WebAuthn passkeys, JWT. Данные - MySQL, Meilisearch. Инфра - self-hosted на одном VPS: Docker, nginx, GitHub Actions. Базируется в Москве. Сейчас - Wildberries, Senior Frontend Developer: разрабатывает одно из приложений портала продавцов на React. До этого - Яндекс Пэй (Яндекс Финтех): владел фронтендом многоуровневого KYC, поднял Apollo Server BFF для KYC/регистрации/OTP-авторизации, разработал подход интеграции в вебью с safe-area на уровне дизайн-системы. Ранее - Студия Артемия Лебедева: финансовые калькуляторы (сам написал движки аннуитетной и дифференцированной амортизации с полным покрытием юнит-тестами), CMS-архитектура страниц-блоков, анимационно-насыщенные продуктовые сайты. Ещё раньше - 6+ лет в Compass Plus на платформах интернет-банкинга TranzAxis/TranzWare: миграция ядра клиента с ES3 на ES6 и Java/Spring на бэке. Опыт legacy-миграций и performance-работы - ленивая загрузка в масштабе, обход утечки памяти Chromium. Поддерживает open-source TypeScript-библиотеки (эмуляторы машин Тьюринга и Поста с интерактивным демо). Ведёт задачи от архитектуры до прода.

ОПЫТ РАБОТЫ

Wildberries

июнь 2026 - настоящее время

Senior Frontend Developer · Москва, Россия (гибрид/удалённо)

Крупнейший e-commerce-маркетплейс России и СНГ.

- Разрабатываю одно из приложений портала продавцов на React
- **Stack:** [React](#) · [Angular](#)

Яндекс Финтех (Яндекс Пэй)

апрель 2022 - ноябрь 2025

Senior Frontend Engineer (incl. BFF/SSR) · Москва, Россия (гибрид/удалённо)

Платёжная платформа и цифровой кошелек Яндекс Финтеха - веб-приложение и нативные вебью внутри суперапа Яндекса.

- Сделал флоу регистрации - точку входа в продукт, через которую проходит каждый дальнейший сценарий
- Владел фронтендом многоуровневого KYC - воронки, через которую каждый пользователь кошелька получает доступ к полноценному банковскому продукту. Три уровня: базовый, mobileID + Госуслуги, полная идентификация - либо через курьера на Яндекс Картах с полигонами зон доставки для выбора доступного места встречи, либо через онлайн-флоу с переходом в нативное приложение по диплинку
- Реализовал BFF-слой (Apollo Server / GraphQL) для регистрации, KYC и OTP-авторизации - собирал вызовы внутренних сервисов в единый граф для фронтенда
- Разработал подход интеграции в вебью с учётом safe-area на уровне дизайн-системы (iPhone notch, Dynamic Island) - продуктовые команды перестали дублировать мобильные хаки в каждом проекте
- Вёл интеграцию тёмной темы по веб-поверхностям
- Спроектировал пайплайн генерации CSS из Figma-токенов - токены попадают в код без ручного переноса
- **Stack:** [AbortController](#) · [Next.js](#) · [React](#) · [Apollo GraphQL](#) · [React Hook Form](#) · [Express](#) · [NestJS](#) · [TypeScript](#) · [Zod](#)

Фронтенд-разработчик · Москва, Россия

Одна из самых известных российских дизайн-студий - несколько клиентских продуктов.

- **Сайт пластического хирурга Alex Glushko (alexglushko.ru)** - CMS-архитектура страниц-блоков: страницы из типизированных блоков с настраиваемыми позициями изображений и динамической генерацией SVG; контент-зависимые попапы; анимации Lottie + canvas; адаптивные SVG на ResizeObserver; текст по кривой; ленивая загрузка с обходом утечки памяти в Chromium. Анонсирован студией Артемия Лебедева. **Stack:** [React](#) · [Sass](#) · [Redux](#)
- **Сайт «Финансовая культура» - Банк России (fincult.info)** - сам написал движки кредитного и депозитного калькуляторов с полным покрытием юнит-тестами; корректное начисление процентов с внеплановыми погашениями; аннуитет и дифференцированная амортизация; банковское округление; сделал UI инфляционного калькулятора - динамические SVG-графики инфляции (с d3-хелперами) и режим сравнения «суперфудов» (движок калькулятора реализован на бэкенде другой командой); карты Leaflet; кастомные контролы форм. Анонсирован студией Артемия Лебедева. **Stack:** [React](#) · [Redux](#)
- **Сайт Агробанк Узбекистан (agrobank.uz)** - многоязычный банковский сайт с роутингом ru/uz через i18next; поиск отделений на Яндекс Картах; клиентская генерация PDF через @react-pdf/renderer; библиотека компонентов в Storybook. Анонсирован студией Артемия Лебедева. **Stack:** [React](#) · [Sass](#) · [Redux Toolkit](#) · [TypeScript](#)
- **Игротека «Мир Достоевского» (dostoevskyworld.ru)** - сборник мини-игр на Angular + RxJS: чат с «Фёдором Достоевским» с эффектом онлайн общения в чате, сортировка событий «Идиота» в хронологическом порядке, стенографирование отрывков «Игрока» под аудиодорожку с посимвольным диффом против канонического текста, фактчек «верю / не верю» по биографии. Кастомизация флоу игры - порядок мини-игр и наборы вопросов задаются независимо. Анонсирован студией Артемия Лебедева. **Stack:** [Angular](#) · [RxJS](#) · [TypeScript](#)

Compass Plus

июль 2014 - сентябрь 2020

Frontend + Backend Developer · Магнитогорск, Россия

Платформы интернет-банкинга TranzAxis Internet Banking (TXIB) и TranzWare Internet Banking (TWIB) - клиентский веб-доступ к банковским продуктам и смежная серверная логика на Java.

- Спроектировали с коллегами модульную систему клиентских компонентов с ленивой загрузкой и client-side рендером - основу, на которой команда продолжала наращивать функциональность без деградации первичной загрузки
- Реализовал клиентские флоу интернет-банкинга: счета, карты и депозиты, привязка карт сторонних банков, платежи вендорам, история операций
- Перевёл ядро клиентского приложения с ES3 на ES6 и поднял Rollup-сборку под ES-модули; параллельно поддерживал неперенесённый ES3-слой компонентов, удерживая обратную совместимость на протяжении миграции
- Писал backend-логику на Java (Tomcat, Spring, Groovy): серверные компоненты банковских операций и REST API для фронтенда
- **Stack:** [CSS3](#) · [ES3](#) · [ES6](#) · [ESM](#) · [HTML5](#) · [JavaScript](#) · [Groovy](#) · [Java](#) · [REST APIs](#) · [Spring](#) · [Tomcat](#) · [XML](#) · [XSD](#) · [XSLT](#) · [Rollup](#)

ПРОЕКТЫ

Yandex Pay · pay.yandex.ru

Веб-приложение и нативные вебвью платёжного сервиса. Сделал флоу регистрации; владел большей частью фронтенда многоуровневого KYC - базовый, mobileID + Госуслуги, полная идентификация через офлайн-курьера (Яндекс Карты с полигонами зон доставки) или онлайн-селфи. Поднял Apollo Server BFF для регистрации, KYC и OTP-авторизации - собирал вызовы внутренних сервисов в единый граф для фронтенда. Вёл интеграцию тёмной темы и пайплайн генерации CSS из Figma-токенов, поработал над безопасной зоной вебвью на мобильных устройствах (safe-area)

Poetry Site · poetry.mellonis.ru

Личный фулстек-проект - стихи и проза с закладками, комментариями, голосованием и полнотекстовым поиском. - Перенёс долгоживущий PHP-сайт в мульти-стек-архитектуру - Фронтенд: Next.js / React; Бэкенд: Fastify + TypeScript + Zod API с JWT + WebAuthn passkeys, rate-limiting, структурированным логированием - MySQL 8 + Meilisearch для полнотекстового поиска - Локальная разработка на DDEV; журналирование контейнеров через Dozzle - Self-hosted на одном VPS: Docker, nginx, Let's Encrypt, GitHub Actions CI/CD

machines-demo · demo.machines.mellonis.ru

Интерактивный браузерный эмулятор машин Тьюринга и Поста (Svelte 5). Пользовательский код выполняется в Web Worker - unsafe-eval изолирован в воркере, и воркер же служит границей безопасности; между потоками - только plain data через postMessage. Terminate-on-timeout. Главный поток держит UI и зеркальную TuringMachine, которая повторяет команды воркера шаг за шагом, обновляя анимированную ленту. Vite, CodeMirror 6 с Lezer-проверкой синтаксиса до запуска, Playwright e2e.

@turing-machine-js/machine · github.com/mellonis/turing-machine-js

npm-монорепозиторий: ядро (State, Tape, TapeBlock, TuringMachine) с round-trip-сериализацией состояний через Mermaid. Две библиотеки двоичной арифметики делят операции - 5-символьный маркер-вариант vs 3-символьный bare - наглядно показывая trade-off размера алфавита и числа состояний.

@post-machine-js/machine · github.com/mellonis/post-machine-js

Конструктор машин Поста, компилирующий инструкции в State-граф @turing-machine-js/machine; обе библиотеки делят одно ядро через peer-dependency. Поддерживает подпрограммы (call), инлайн-группы и ветвление через check.

15-puzzle · 15-puzzle.mellonis.ru

SPA-«пятнашки» на ванильном JS + Vite в паре с Node-сервисом без рантайм-зависимостей. Уровни образуют цепочку: доски генерируются детерминированно (mulberry32 + Фишер-Йейтс + проверка решаемости по чётности), сервер переигрывает присланные ходы перед подписью Ed25519. Каждый подписанный блоб указывает только на следующий уровень, а URL уровней - нечитаемые 16-символьные хеш-хеши, так что цепочка обеспечивается и на уровне URL.

Minesweeper · mines.mellonis.ru

Сапёр на ванильном TypeScript. Игровое ядро без фреймворков, DOM и canvas; отдельный canvas-рендерер с оффскрин-спрайтами и risk-канвасом для определения попаданий по цвету. С учётом devicePixelRatio, тесты на Jest.

TXIB - TranzAxis Internet Banking

Веб-платформа интернет-банкинга Compass Plus для розничных и SMB-клиентов (2014-2020) - счета, карты, депозиты, привязка карт сторонних банков, платежи вендорам, история операций. Модульная система клиентских компонентов с ленивой загрузкой и client-side рендером; интеграция с Java/Spring-бэкендом через REST API.

TWIB / IB6 - TranzWare Internet Banking

Платформа интернет-банкинга Compass Plus следующего поколения (2014-2020) - новая архитектура относительно TWIB 5. Построена на мигрированной ES6 + ES-модульной основе.

СЕРТИФИКАТЫ

React-разработчик - Яндекс Практикум (АНО ДПО «ШАД») · React, Redux/RTK, React Router, TypeScript, WebSocket, юнит- и E2E-тесты, дипломный проект undefined

TypeScript Intensive - Илья Кантор / javascript.ru · дженерики, utility-типы, типо-ориентированный дизайн, интеграция со сборщиками undefined

RxJS Intensive - Илья Кантор / javascript.ru · hot/cold observables, шедулеры, композиция операторов, обработка ошибок undefined

Погружение в Python - МФТИ, Mail.Ru Group и ФРОО / Coursera · ООП, async/await, многопоточность undefined

НАВЫКИ

Фронтенд: React, Next.js, Svelte, TypeScript, JavaScript, Redux Toolkit, RxJS, Apollo GraphQL (клиент), i18next, Storybook, Sass

Анимации и UI: Lottie, canvas, генерация SVG, Swiper

Формы и документы: React Hook Form, Zod, Inputmask, @react-pdf/renderer

Design-system: Figma-token CSS pipeline, safe-area-aware вебвью-интеграция

Browser APIs: Web Workers, ResizeObserver, IntersectionObserver, AbortSignal

Карты: Яндекс Карты, Leaflet, Google Maps

Бэкенд: Node.js, Fastify, NestJS, Express, GraphQL/Apollo Server, REST API, rate-limiting, структурированное логирование

Security/Auth: WebAuthn passkeys, JWT, OTP, Ed25519, Web Worker sandboxing

JVM (исторически): Java, Spring, Tomcat, Groovy

Данные: MySQL 8, Meilisearch

Сборка: Vite, Rollup, Webpack

DevOps: Docker, DDEV, Dozzle, nginx, Let's Encrypt, GitHub Actions CI/CD, self-hosted VPS, публикация пртм-монорепозитория

Тестирование: Jest, Playwright

Прочее: Python (ООП, asyncio, многопоточность - сертифицирован)

Языки: русский (родной), английский (B1 - Pearson)