Engineer

Побудувала нативний клієнт для macOS на Swift 6

Побудувала нативний клієнт обміну повідомленнями для macOS на Swift 6 і SwiftUI — 11 141 рядок у 53 файлах — поверх C-інтерфейсу ядра на Rust, злінкованого як статичний архів із зафіксованої ревізії.

серпень 2026

11,141

рядків Swift над ядром на Rust

Ситуація. Протокол обміну повідомленнями мав добре протестоване ядро на Rust і клієнти для кількох платформ, але настільна версія для macOS була кросплатформною оболонкою: вона не мала вигляду застосунку для Mac, не поводилася як такий застосунок і тягнула за собою середовище виконання, яке нативному застосунку не потрібне. Ядро надає свої можливості через C-інтерфейс, тобто будувати на ньому може будь-яка мова, здатна викликати C, — а Swift здатна.

Завдання. Нативний клієнт треба було створити безпосередньо на цьому C-інтерфейсі, актуальною мовою платформи та на її актуальному фреймворку для інтерфейсу користувача, причому ядро мало бути скомпоноване із застосунком, а не постачатися поруч.

Дія. Застосунок — це 11 141 рядок Swift у п’ятдесяти трьох файлах цілі застосунку; він побудований на SwiftUI, а ядро скомпоновано з ним як статичний архів, зібраний із закріпленої ревізії основного репозиторію. Саме закріплення ревізії замість стеження за гілкою робить збірку відтворюваною: C-інтерфейс — це контракт, а ядро, яке постійно змінюється, змінює контракт, не змінюючи жодного рядка Swift. Шари побудовано так, щоб небезпечна поверхня лишалася малою: тонка обгортка володіє кожним вказівником і кожним рядком, що перетинає межу, моделі над нею — звичайні значення Swift, а шар інтерфейсу користувача ніколи не бачить сирого вказівника. Для всього, що повертає C-інтерфейс, діє правило

володіння, і помилка в будь-якому з них призводить або до витоку пам'яті, або до аварійного завершення, причому компілятор ні про що не попереджає, тож саме в обгортці зосереджена вся дисципліна роботи з пам'яттю в проєкті. Збіркою керує інструмент запуску завдань із п'ятдесятьма шістьма цілями, які охоплюють компіляцію ядра, збірку Swift, набори тестів, лінтери та пакування.

Результат. Нативний клієнт, який запускається як застосунок для Mac, використовує власні матеріали й елементи керування платформи та не містить вбудованого браузера. Варто прямо сказати, чого не зроблено: це не випущений реліз. Немає ні розповсюдження нотаризованих збірок, ні каналу оновлень, а інтерфейс доступний лише однією мовою, тож це робочий застосунок, а не продукт, яким користується хтось іще.

Застосуйте цей досвід у своєму проєкті

Використайте «Побудувала нативний клієнт для macOS на Swift 6» як відправну точку для схожого завдання у вашій організації. Послуги, що стоять за цією роботою, доступні й для вашого проєкту. Структура системи, спершу продумана на папері, щоб частини пасували одна до одної, а вартість була відома до першого рядка коду.

Переглянути наші послуги Зв'язатися з нами →

Дізнайтеся, що привело до результату

Прочитайте про ситуацію, рішення та результат цього прикладу. Ми можемо обговорити, що відповідає вашим умовам, а що потребує іншого підходу. Побудувала нативний клієнт для macOS на Swift 6 — 11,141 рядків Swift над ядром на Rust

Переглянути наші послуги Зв'язатися з нами →

Потрібен справжній нативний застосунок, а не обгорнута вебсторінка?

Обговоріть свій проєкт з інженером, відповідальним за виконання. Обсяг, основу розрахунку вартості та передачу результатів узгоджуємо письмово до початку робіт.

Роль у цій роботі: Інженер застосунків macOS. Тут задіяні Архітектура рішень, Frontend-розробка, Full-Stack-розробка, API та інтеграції, Дизайн-системи та UI та UX / UI-дизайн. І, до речі, ми добре вміємо Архітектура платформи та рішень, Frontend-розробка, Full-Stack-розробка продуктів та UI/UX-дизайн та дизайн-системи.

Зв'язатися з нами Ознайомтеся з нашими послугами

Одна частина **довшого переліку** — і він увесь на цьому сайті.

Кожен запис написано однаково: ситуація, завдання, що ми зробили і що змінилося.

Переглянути весь перелік

Визначте потрібне покращення

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Потрібен справжній нативний застосунок, а не обгорнута вебсторінка?

Запитайте про рішення в роботі «Побудувала нативний клієнт для macOS на Swift 6». Ви говоритимете з людьми, які можуть пояснити роботу й оцінити ваш проєкт.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/portfolio/built-a-native-macos-client-in-swift-6/