

## Розробка кроссплатформених додатків

Олександр Безверхий<sup>1</sup>, Олександр Куценко<sup>2</sup>

Національний транспортний університет  
вул. Михайла Омеляновича-Павленка 1, Київ, 01010  
<sup>1</sup>[o\\_bezver@ukr.net](mailto:o_bezver@ukr.net), <https://orcid.org/0000-0002-0834-6335>  
<sup>2</sup>[alexkutsenko95@gmail.com](mailto:alexkutsenko95@gmail.com)

Отримано 25.04.2021, прийнято 19.05.2021  
<https://doi.org/10.32347/tit2141.0307>

### ВСТУП

Робота присвячена дослідженню кроссплатформеної розробки додатків, елементів архітектури, інструментів, мов програмування та фреймворків. Під час роботи були розглянуті існуючі засоби розробки, детально досліджено один фреймворк який наразі є єдиним на сьогодні в своєму роді. Розроблені рекомендації що до створення додатків.

### МЕТА І МЕТОДИ ДОСЛІДЖЕННЯ

Сьогодні технології вже мають багато платформ, бібліотек та фреймворків. Open-source відкриває для програмування безмежні можливості. Розробка додатків та сайтів стає дуже простою і привабливою не тільки для розробників але і для звичайних користувачів. Однак постає проблема в кроссплатформеності та в універсальності коду. А саме використання одного й того ж коду для всіх платформ (web, ios, android). Кожна з платформ використовує власні мови, бібліотеки і навіть не завжди вони стандартизовані і ідеально працюють. Тому для вирішення цієї проблеми було створено фреймворк та платформу на мові програмування Javascript під назвою Expo. Цей фреймворк дозволяє створювати універсальні додатки для всіх платформ одночасно. Він компілює Javascript код в код який необхідний для кожної з платформ. Фундаментально це заощаджує дуже багато ресурсів на розробку продукту. З точки зору бізнесу це дуже вигідно. Javascript є однією з найпопулярніших мов, тому знайти розробників буде не важко. Швидкість цього додат-

ку буде високою через використання бібліотеки React для відображення даних. Також є підтримка SSR що є необхідним для пошукової оптимізації додатка та покращення продуктивності. Цю технологію вже використовують Twitter, Glovo, Uber Eats проте вона є ще досить новою та її екосистема не така досконала як наприклад в React, тому саме ви можете докласти зусиль на її розвиток.

### ОГЛЯД ОСНОВНИХ ТЕХНОЛОГІЙ, ЯКІ ВИКОРИСТОВУЄ EXPO

Javascript – динамічна, об'єктно-орієнтовна, прототипна мова програмування. Може працювати як на стороні сервера (express js , koa js) так і на стороні клієнта . JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування ( імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу. Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності створення односторінкових та прогесивних веб-застосунків (React, AngularJS, Vue.js);
- програмування на боці сервера (Node.js(Express.js ) );
- стаціонарних застосунків (Electron, NW.js);

- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладних програмах (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є абсолютно різними, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme. React – відкрита JavaScript-бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці одно сторінкових застосунків. Розробляється Facebook, Instagram і спільнотою індивідуальних розробників. React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає виду в шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови веб-застосунків. В даний час React використовують Khan Academy, Netflix, Yahoo, Airbnb, Sony, Atlassian та інші. React підтримує віртуальну реальність у браузері. Віртуальна реальність (англ., Virtual reality) – різновид реальності в формі тотожності матеріального й ідеального, що створюється та існує завдяки іншій реальності. В вужчому розумінні — ілюзія дійсності, створювана за допомогою комп'ютерних систем, які забезпечують зорові, звукові та інші відчуття. React-native-web – "React Native for Web" дозволяє запускати компоненти та API React Native в Інтернеті за допомогою React DOM.

Якісні веб-інтерфейси: спрощує створення швидких адаптивних веб-інтерфейсів у JavaScript. Він забезпечує взаємодію рідної якості, підтримку декількох режимів введення (дотик, миша, клавіатура), оптимізовані стилі з префіксом постачальника, вбудовану підтримку макета RTL, вбудовану доступність та інтеграцію з React Dev Tools.

Пишіть один раз, робіть де завгодно: взаємодіє з існуючими компонентами React DOM і сумісний з більшістю API React Native. Ви можете розробляти нові компоненти для власної та веб-мережі, не переписуючи існуючий код. React Native for Web може також відображати HTML та критичний CSS на сервері за допомогою Node.js. SSR (Server-Side Rendering) генерує повний HTML для сторінки на сервері у відповідь на запит (перехід по посиланню). Це дозволяє уникнути додаткових запитів даних, наповнення шаблонів на стороні клієнта, оскільки вони обробляються до того, як браузер отримує відповідь. Загалом, SSR створює швидкі FP (First Paint) та FCP (First Content Paint). Запуск логіки сторінок і рендеринг на сервері дають можливість уникнути надсилання клієнту великої кількості JavaScript, що допомагає досягти швидкого TTI (Time To Interactive). Це має сенс, оскільки при SSR ви просто надсилаєте текст і посилання браузеру користувача. Такий підхід добре працює для широкого спектра пристроїв і якості мереж, а також дає можливість для цікавої оптимізації браузера, як, наприклад, потоковий аналіз документів. Зі SSR користувачам навряд чи доведеться чекати на обробку пов'язаного CPU bound JavaScript перш ніж вони зможуть використовувати ваш сайт. Навіть коли не можна уникнути сторонніх JS, використання SSR для зменшення JS-навантаження може дати вам більше "бюджету" для інших процесів. Однак, існує один основний недолік такого підходу: генерування сторінок на сервері вимагає часу, що часто може призвести до повільнішого TTFB (Time to First Byte).

## ПОРІВНЯННЯ З ТРАДИЦІЙНИМИ ТЕХНОЛОГІЯМИ

Android (вимов. Андроїд) — операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance (ОНА). Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java. У 84 % смартфонів, проданих у 3-му кварталі 2014 року, була встановлена операційна система Android[9]. У березні 2017 року ОС Android стала найпопулярнішою ОС, з якої виходили в інтернет. Так, 37,93 % користувачів заходили в інтернет із Android'a, а з Windows – 37,91 % користувачів. В Азії показники ще вищі – 52,2 % і 29,2 % відповідно.

### Переваги

Деякі користувачі відзначають, що Android проявляє себе краще одного зі своїх конкурентів, Apple iOS, в ряді особливостей, таких як веб-серфінг, інтеграція з сервісами Google і інших. Також Android, на відміну від iOS, є відкритою платформою, що дозволяє реалізувати на ній більше функцій. Незважаючи на початкову заборону на установку програм з «неперевіраних джерел» (наприклад, з карти пам'яті), це обмеження відключається штатними засобами в налаштуваннях пристрою, що дозволяє встановлювати програми на телефони та планшети без інтернет-підключення (наприклад, користувачам, які не мають Wi-Fi-точки доступу і не бажають витратити гроші на мобільний інтернет, який зазвичай коштує дорого), а також дозволяє будь-кому безкоштовно писати програми для Android і тестувати на своєму пристрої. Android доступний для різних апаратних платформ, таких як ARM, MIPS, x86. Існують альтернативні Google Play магазини додатків: Amazon Appstore, Opera Mobile Store, GetUpps!, F-Droid. У версії 4.3 з'явилась підтримка багатокористувачького режиму.

### Критика

Платформа базується на Java (спеціальна реалізація Dalvik), тому переваги і можливості операційної системи Linux на цій платформі практично не використовуються. Наприклад, не використовується жоден з популярних графічних тулкітів і бібліотек (наприклад Qt або GTK), що робить малоймовірною появу значної кількості застосунків, портованих з повноцінного десктопного варіанту Linux на цю платформу через відсутність поза вибором X-сервера і поширених графічних бібліотек. З'явилась інформація про те, що Google на свій розсуд видалятиме застосунки на телефонах користувачів, якщо порушуються умови їх використання. Конкуренти Android виступили з критикою платформи, звинувачуючи її в надмірній фрагментації, що створює перешкоди розробникам. Google спростувала всі звинувачення, заявивши, що ніяких подібних проблем немає. Щоб користувачі телефонів мали доступ до Google Play та інших сервісів від Google, виробники цих телефонів мають укласти контракт з Google на використання відповідного пропріетарного програмного забезпечення.

**Kotlin** (Котлін) – статично типізована мова програмування, що працює поверх JVM і розробляється компанією JetBrains. Також компілюється в JavaScript. Мову названо на честь острова Котлін у Фінській затоці, на якому розміщена частина Кронштадту. Автори ставили перед собою ціль створити лаконічнішу та типобезпечнішу мову, ніж Java, і простішу, ніж Scala. Наслідками спрощення, порівняно з Scala стали також швидша компіляція та краща підтримка IDE. Мова розробляється з 2010 року, публічно представлена в липні 2011. Сирцевий код було відкрито в лютому 2012. В лютому було випущено milestone 1, який містив плагін для IDEA. У червні – milestone 2 з підтримкою Android. У грудні 2012 року вийшов milestone 4 та забезпечив підтримку Java 7. Станом на листопад 2015 року основні можливості мови стабілізовані, готується реліз версії 1.0. В грудні 2015 року з'явився реліз-кандидат версії 1.0, а 15 лютого 2016 року відбувся реліз версії 1.0. З 17 травня 2017 року входить в список

офіційно підтримуваних мов для розробки застосунків для платформи Android. Із 7 травня 2019 року є рекомендованою мовою для розробки Android застосунків

**Swift** – багатопарадигмова компільована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C і бути стійкішою до помилкового коду. Swift була представлена на конференції розробників WWDC 2014. Мова побудована з LLVM компілятором, включеного у Xcode 6 beta. Безкоштовний посібник мови програмування Swift доступний для завантаження у магазині iBooks. Компілятор Swift побудований з використанням технологій вільного проекту LLVM. Swift успадковує найкращі елементи мов C і Objective-C, тому синтаксис звичний для знайомих з ними розробників, але водночас відрізняється використанням засобів автоматичного розподілу пам'яті і контролю переповнення змінних і масивів, що значно збільшує надійність і безпеку коду. При цьому Swift-програми компілюються у машинний код, що дозволяє забезпечити високу швидкодію. За заявою Apple, код Swift виконується в 1.3 рази швидше коду на Objective-C. Замість збирача сміття Objective-C в Swift використовуються засоби підрахунку посилань на об'єкти, а також надані у LLVM оптимізації, такі як автовекторизація. Мова також пропонує безліч сучасних методів програмування, таких як замикання, узагальнене програмування, лямбда-вирази, кортежі і словникові типи, швидкі операції над колекціями, елементи функційного програмування. Основним застосуванням Swift є розробка користувачьких застосунків для macOS, iOS, tvOS, watchOS з використанням тулкіта Cocoa і Cocoa Touch. При цьому Swift надає об'єкт-

ну модель, сумісну з Objective-C. Сирцевий код мовою Swift може змішуватися з кодом на C і Objective-C в одному проекті. Swift щільно інтегровано до власницького середовища розробки Xcode, проте може бути викликано з терміналу, що уможливує її використання на операційних системах, відмінних від macOS, наприклад, на Linux. Окремо варто відзначити, що Swift від компанії Apple не варто плутати з досить давно розроблюваною скриптовою мовою Swift, націленої на багатонитеве програмування і поставленого під вільною ліцензією Apache.

### ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

В роботі, було детально досліджено інноваційний на сьогодні інструмент для розробки додатку для всіх платформ та його ключові елементи. Проаналізувавши всі платформи було встановлено, що на сьогодні нативні додатки мають перевагу у продуктивності, проте швидкість розробки і універсальність коду має Expo. Автори вважають що у майбутньому продуктивність цих додатків буде однаковою та процес створення додатків буде швидший. Досліджені всі переваги Expo. Встановлено, що для створення додатку для всіх платформ необхідно знати лише мову програмування Javascript в той час для розробки додатків без Expo необхідно знати як мінімум 3 мови (Kotlin/Java, Swift, Javascript). Та витратити мінімум у 3 рази більше часу та в 3 рази більше репозитаріїв для зберігання коду. Підсумовуючи можна сказати що Expo вперше дає нам можливість створювати додатки на всіх платформах і не витрачати на це багато часу.