

Flutter vs. React Native: Что лучше в 2023 году

16.03.2023

По мере роста популярности разработки мобильных приложений компании ищут способы создания кроссплатформенных приложений, которые можно использовать на различных устройствах. Когда мы говорим “кроссплатформенные”, мы, конечно, имеем в виду Android и iOS. По данным Statista:

Android сохранил свою позицию ведущей мобильной операционной системы во всем мире в июне 2021 года, контролируя рынок мобильных ОС с долей почти 73 процента. Android от Google и iOS от Apple совместно занимают более 99 процентов мирового рынка. В этой статье мы сравним два популярных фреймворка для кроссплатформенной разработки: Flutter и React Native. Мы рассмотрим плюсы и минусы каждого фреймворка и обсудим, какой из них лучше подходит для использования в 2023 году.

Почему разработка мобильных приложений так популярна?

Разработка мобильных приложений – это постоянно растущая бизнес-ниша. Практически у всех людей на планете есть мобильные телефоны, что означает практически неограниченное количество потенциальных пользователей. Следовательно, в наше время существуют приложения практически для всего. Вы можете выбрать множество способов разработки и создания приложения. Вы можете использовать нативные методы, например, Swift и Objective-C для создания приложений для iOS и Java для приложений для Android. Это официальные языки программирования Apple / Google соответственно, которые обеспечивают поддержку и часто обновляемые функции.

В качестве альтернативы можно использовать кроссплатформенные

фреймворки, такие как Flutter или React Native.

Что такое кроссплатформенная разработка приложений?

Прежде чем начать, давайте дадим определение термину “кроссплатформенная разработка приложений” и разделим его на две категории: Гибридная разработка и Нативная разработка.

Кроссплатформенные приложения – это приложения, которые могут быть разработаны с использованием единой кодовой базы и функционируют практически одинаково как на операционных системах iOS, так и Android. (В этой статье мы сосредоточимся на разработке мобильных приложений; о веб-приложениях и приложениях для настольных компьютеров мы поговорим здесь).

Гибридная разработка

Гибридные приложения разрабатываются с использованием комбинации веб-технологий, таких как HTML5, CSS и JavaScript. Это означает, что гибридные приложения используют общий код для разных платформ (например, код HTML/CSS/JS), и этот общий код запускается в веб-вью на целевой платформе.

Приложения WebView – это гибридные приложения, которые используют встроенные веб-просмотры для отображения пользовательского интерфейса, внутри которого вы можете использовать HTML5, CSS и JavaScript для настройки.

Приложения WebView имеют некоторые ограничения в доступе к API устройства из коробки, что требует дополнительных усилий для достижения некоторых из тех же функций, что и в “родных” приложениях. Компромиссом является то, что эти приложения кроссплатформенные, что может значительно сэкономить время.

Гибридные приложения могут выглядеть одинаково на обеих платформах, но вести себя по-разному, в зависимости от доступных им API для конкретной платформы. Например, погодное

приложение будет проверять API погодной службы текущего местоположения на обеих платформах и возвращать различные данные в зависимости от того, что доступно на каждой платформе.

Нативная разработка

Нативные приложения разрабатываются с использованием нативных SDK для целевых платформ (например, Android или iOS). Это означает, что они не имеют общего кода для разных платформ, и этот общий код написан только для целевой платформы, в то время как пользовательский интерфейс реализован с помощью виджетов и библиотек, специфичных для конкретной платформы.

Нативные приложения обеспечивают лучший пользовательский опыт, чем гибридные приложения, а также выглядят более естественно на каждой платформе, но они дороже в разработке и требуют больше времени для выпуска новых функций из-за времени, необходимого разработчикам для изучения API целевых платформ.

В целом, идеальным вариантом является разработка приложения с помощью нативного инструмента разработки для целевых платформ (например, Android Studio или Xcode).

Flutter и React Native: Кросс-платформенные фреймворки

Flutter и React Native являются одними из лучших кроссплатформенных фреймворков для разработки на сегодняшний день. Они оба используют нативные виджеты для создания высоконастраиваемого, отзывчивого пользовательского интерфейса и совместного использования кода на разных платформах.

Фреймворк Flutter разработан компанией Google, а фреймворк React Native – компанией Facebook, поэтому эти технологические гиганты имеют очень большие команды, занимающиеся всем, начиная от SDK платформы и заканчивая документацией, поддержкой и т. д.

Хотя приложения Flutter и React Native являются нативными, они выгодно отличаются от традиционной разработки нативных приложений тем, что могут разделять значительную часть своей кодовой базы между платформами.

По данным Instagram, объем кода, разделяемого между iOS и Android с помощью функций React Native, составил более 90%.

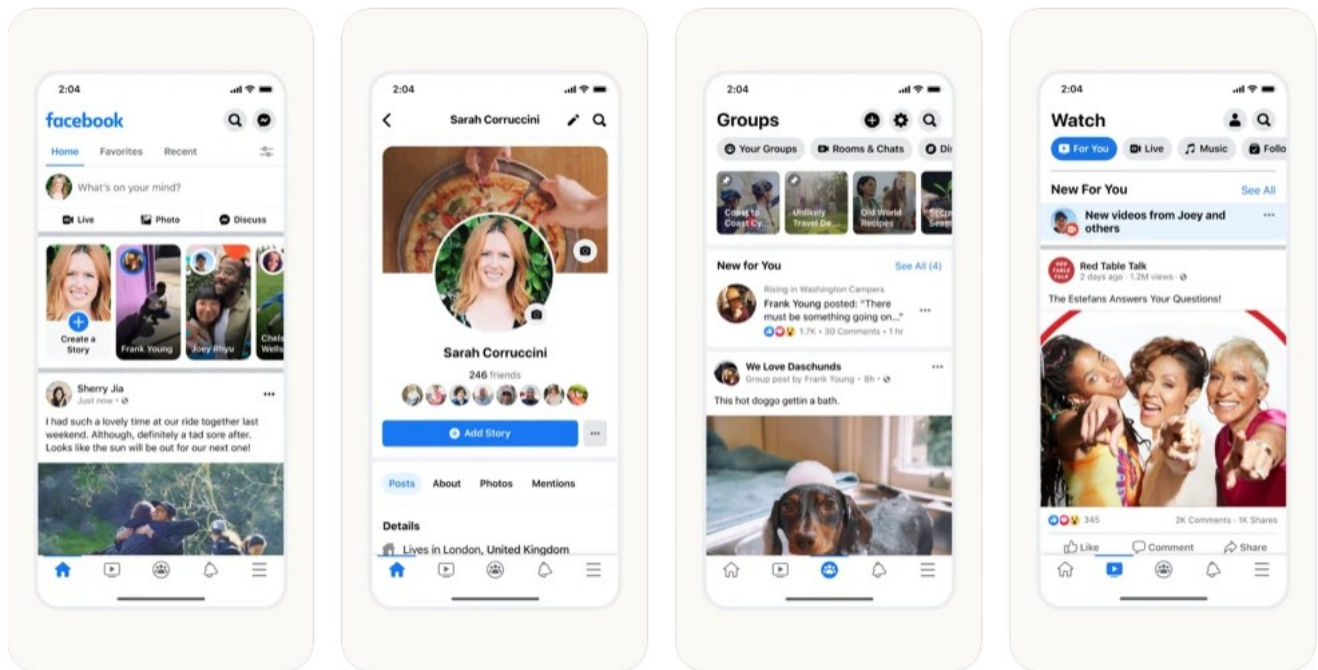
История Flutter и React Native

Flutter был анонсирован на саммите разработчиков Dart в октябре 2016 года. Основная идея этого кроссплатформенного фреймворка для разработки мобильных приложений – дать разработчикам инструменты для создания нативных приложений для iOS и Android с помощью единой кодовой базы, написанной на собственном языке программирования Dart от Google.

Первый стабильный релиз (1.0) Flutter вышел 16 февраля 2018 года.

React Native стартовал немного раньше, чем Flutter: его первая бета-версия была выпущена в марте 2015 года, однако выйти из стадии бета-версии ему удалось только в марте 2017 года. 5 сентября 2017 года React Native достиг версии 1.0 при огромной поддержке со стороны технологического гиганта Facebook.

Как видите, обе технологии относительно новые, но не позволяйте этому обмануть вас. React Native уже используется такими крупными игроками, как Facebook, Instagram, Airbnb и Uber.



Flutter не может похвастаться такими же громкими именами, но его уже приняли BMW, Toyota, eBay и, конечно, Google Pay.

Чем отличаются Flutter и React Native

Основное различие между Flutter и React Native заключается в том, что React Native не компилируется в нативный мобильный язык (Java, Swift, Objective-C), а просто выполняет код JavaScript. Flutter, с другой стороны, компилирует свой язык Dart в нативный, что может повлиять на производительность (об этом позже).



JavaScript

Dart

Еще одно большое отличие заключается в том, что для написания компонентов на React Native используется ванильный JavaScript (помимо JSX). Разработчики Facebook рекомендуют использовать Flow или TypeScript при работе с React из-за его динамической природы. Для Flutter компания Google рекомендует использовать Dart для написания кода также из-за его статической системы типизации.

На наш взгляд, решение о том, какую из этих двух технологий выбрать, должно основываться скорее на ваших предпочтениях, чем на их реальных функциях и возможностях.

Конечно, полезно знать все парадигмы программирования, чтобы вы могли легко освоить новые языки и фреймворки, даже если они написаны не на знакомом вам языке.

Программистам, которые уже знакомы с JavaScript (ES2015+) или TypeScript/Flow, будет гораздо проще начать работу с React Native. Это особенно актуально для разработчиков, которые использовали React для веб, так как между React и React Native есть много общего.

JavaScript по-прежнему занимает значительную долю рынка в сфере мобильной разработки, а поскольку и React Native, и Flutter позволяют выбрать любимый язык программирования, сложно сказать, какой из них будет более популярным в 2023

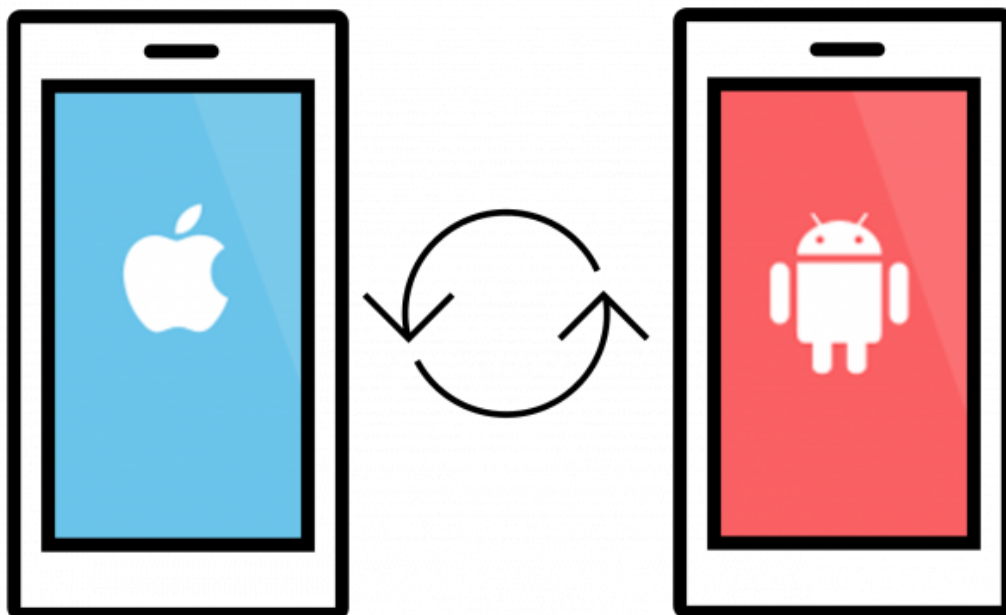
году.

Существуют и другие важные факторы, которые могут повлиять на это решение, например:

- предпочтение компаний к определенному технологическому стеку
- Знакомство разработчиков с тем или иным языком/фреймворком
- Наличие разработчиков с навыками, необходимыми для использования конкретной технологии.

Как мы уже отметили, и React Native, и Flutter будут значимыми игроками в 2023 году, поэтому вам решать, какой вариант лучше для ваших долгосрочных целей.

Как работают кроссплатформенные фреймворки для разработки



Хотя кроссплатформенные платформы для разработки мобильных устройств имеют много общих концепций и функций, каждая из них создавалась с разными целями.

React Native был разработан для обеспечения производительности нативного кода в сочетании с простотой разработки, которую привносит React web. Идея заключается в том, чтобы не использовать единый набор общих компонентов для iOS и Android, а вместо этого иметь полностью отдельные пользовательские интерфейсы, обернутые в единый пучок JavaScript, что позволяет отправлять почти половину кода приложения в одно место.

Flutter был создан в основном для того, чтобы удовлетворить потребности Google... Google. Другими словами, это еще одна попытка объединить быстрые циклы разработки с производительностью нативного кода и созданием многократно используемых компонентов пользовательского интерфейса, которые можно использовать совместно в приложениях для iOS и Android. Именно поэтому Flutter от Google намного быстрее, чем React Native.

Flutter также был разработан с идеей сделать разработку приложений более простой и доступной, поскольку он позволяет писать код с помощью Dart – языка, который можно выучить за выходные и освоить за несколько дней или даже часов в зависимости от навыков разработчика.

Именно поэтому мы считаем, что Flutter станет основой мобильной разработки для компаний, которым необходимо быстро создавать множество нативных мобильных приложений без ущерба для производительности и функциональности.

Создавать мобильные приложения снова интересно

Когда Google анонсировал Flutter, разработчики были ошеломлены тем, насколько хорошо он работает на практике по сравнению с другими технологиями, разработанными специально для создания кроссплатформенных мобильных приложений.

Философия React Native, предусматривающая совместное

использование кода пользовательского интерфейса между iOS и Android, была отличной инициативой, но из-за присущих React Native ограничений получаемые приложения не могут работать так же хорошо, как нативные.

Flutter поставляется с множеством преимуществ, которые вы не найдете ни в одном другом инструменте на сегодняшний день. Dart – это впечатляющий язык, который был создан с нуля для создания мобильных приложений.

На сегодняшний день Dart является самым быстрым языком для создания приложений для Android и iOS, облегчает создание производительных компонентов пользовательского интерфейса, имеет отличную поддержку IDE (интегрированной среды разработки) с мощными функциями автозаполнения, позволяет создавать прототипы в реальном времени без потери состояния приложения и, наконец, принимает объектно-ориентированное программирование, делая его обязательным.

Наличие фреймворка, имеющего собственное мнение, означает, что Google сможет принимать многие важные решения за вас, позволяя сообществу сосредоточиться на том, что действительно важно – создании приложений.

Flutter поставляется с полным набором инструментов и красивым набором виджетов в стиле Material Design, которые разработчики могут повторно использовать в своих приложениях. Google также создал ряд интеграций со сторонними библиотеками, такими как библиотеки обработки изображений для эффективной работы с изображениями, базы данных SQL (доступные через абстракции) и текстовые редакторы.

Все это представлено разработчикам в виде целостного пакета, созданного с учетом скорости, простоты использования, продуктивности и производительности.

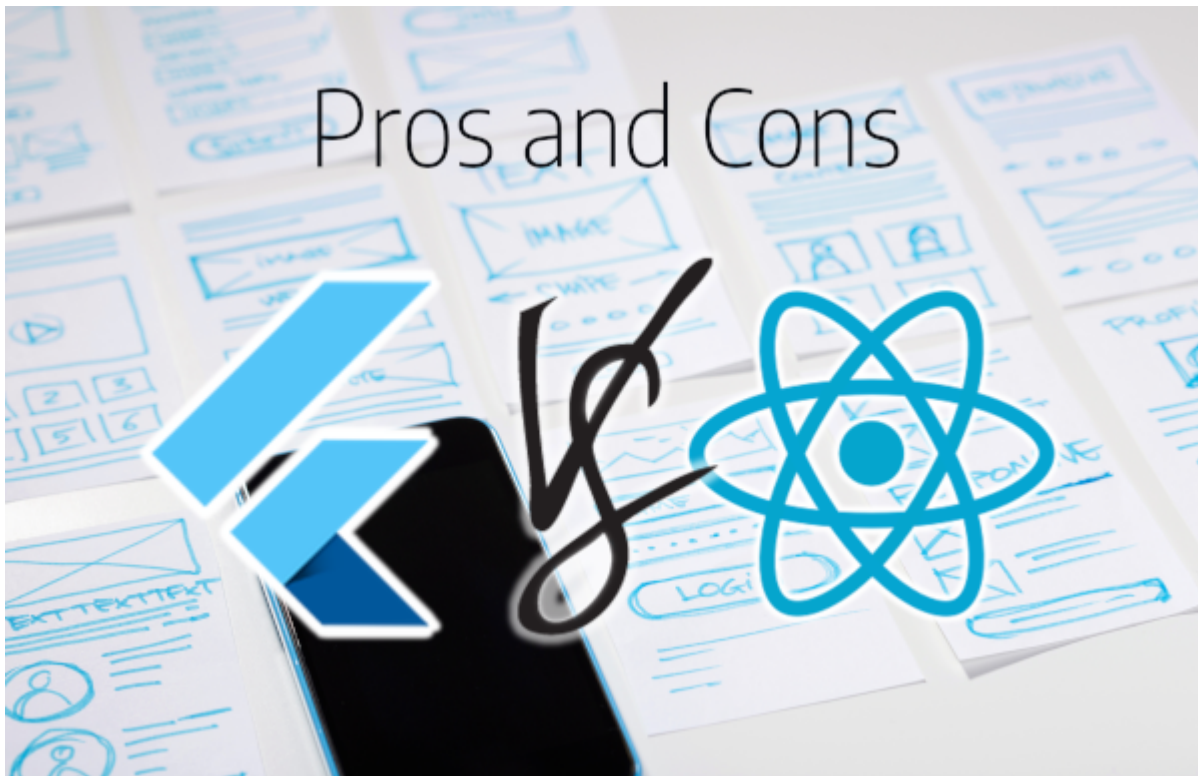
Плюсы и минусы Flutter и React Native Apps

Приложения, созданные с помощью Flutter, неотличимы от нативных. Они имеют ту же производительность и тот же внешний вид (за исключением некоторых стилистических аспектов, характерных для конкретной платформы).

Основные проблемы, на которые люди обычно жалуются при создании приложений с использованием React Native, связаны с его средой выполнения, которая тяжелее, чем управление отдельными процессами для каждой архитектуры. Это означает, что вы не сможете добиться чисто нативной производительности приложения с помощью React Native, хотя и можете приблизиться к этому.

Flutter не обладает теми же преимуществами, что и React Native, в плане поддержки существующих кодовых баз JavaScript и возможности повторного использования некоторых компонентов, общих для приложений для iOS и Android.

Теперь давайте немного углубимся в технические плюсы и минусы этих двух фреймворков.



Плюсы и минусы с точки зрения производительности нативного языка

React Native поставляется с улучшенной виртуальной машиной JavaScript, которая быстрее, чем V8, благодаря JIT-компилятору. Кроме того, он является опережающим компилируемым фреймворком, что означает, что вы можете поставлять любую кодовую базу, которая вам нужна, поскольку она будет скомпилирована в нативный исполняемый файл.

На практике React Native работает так же быстро, как и чистые нативные приложения, поскольку он может достичь такой же производительности, как и приложение для iOS, не требуя изменений в настройках сборки iOS.

Flutter поставляется с собственным компилятором Ahead-of-time, который после создания проекта выдает оптимизированный код для iOS и Android. Вы получаете нативную производительность без необходимости поставлять всю кодовую базу в бинарном файле приложения, как в случае с React Native.

Плюсы и минусы с точки зрения размера приложения

Приложения React Native обычно поставляются с JavaScript runtime, который весит около 300 кб в сжатом виде, хотя это число можно уменьшить, настроив некоторые опции, такие как Bypass filling (что заставит React Native пропустить процесс заполнения своего виртуального DOM с последующим сравнением его с родным UI), а также установив useDeveloperMode в true (что изменит размер изображений в памяти и снизит их качество).

Flutter поставляется с опережающим компилятором, который позволяет разработчикам поставлять только кодовую базу, необходимую для приложения, которое они создают, без необходимости поставлять что-либо в комплекте.

При желании Flutter можно запустить внутри существующей JavaScript VM, что позволит вам сэкономить место, необходимое для вашего приложения.

Плюсы и минусы с точки зрения минимально необходимой версии SDK

React Native обычно может быть создан на базе любого SDK iOS 9+ или Android 5.0+ без каких-либо проблем, но само собой разумеется, что для достижения наилучшей производительности вы должны использовать последние версии SDK, доступные на момент выпуска вашего приложения.

На практике приложения React Native можно создавать на более старых SDK для iOS и Android с ограниченным набором функций, доступных во время выполнения, но для получения всех возможностей все же следует ориентироваться на последние версии SDK.

Приложения Flutter можно создавать на Android версии 21 (Lollipop) и новее, хотя для лучшей производительности

рекомендуется использовать последние версии SDK.

Flutter можно запускать на iOS 8 или новее, но вызов некоторых API может привести к сбоям во время выполнения, поскольку Apple устарела в большинстве API, которые использует Flutter.

Плюсы и минусы с точки зрения разработки пользовательского интерфейса

Flutter поставляется с собственным набором виджетов для визуализации пользовательского интерфейса, что означает, что вы можете повторно использовать существующий код iOS или Android при создании приложений Flutter.

Доступны некоторые сторонние библиотеки для упрощения повторного использования существующих нативных компонентов, хотя эта работа еще не завершена, поскольку нелегко сопоставить виджеты Flutter с существующими компонентами пользовательского интерфейса iOS и Android.

React Native поставляется с мостом, который позволяет вам повторно использовать существующий код iOS и Android в качестве модулей JavaScript, а также предоставляет некоторые API для ручного создания моста между вашими нативными компонентами пользовательского интерфейса и кодом JavaScript, который будет обрабатывать их рендеринг.

Плюсы и минусы с точки зрения отладки

React Native поставляется с собственным отладчиком, который можно прикрепить к запущенному приложению на iOS и Android. Он предоставляет разработчикам предварительный просмотр текущего состояния виртуальной машины JavaScript, а также различные инструменты для проверки использования памяти или настройки некоторых параметров на лету.

Flutter также поставляется с собственным отладчиком, который может быть подключен к вашему запущенному приложению на iOS и

Android, предоставляя разработчикам предварительный просмотр текущего состояния движка рендеринга, а также различные инструменты для проверки использования памяти или настройки некоторых параметров на лету.

Плюсы и минусы в плане повторного использования кода между мобильными платформами

React Native поставляется с собственным набором API, которые можно использовать при разработке как для iOS, так и для Android. Хотя большинство компаний, использующих React Native, сначала разрабатывают свои приложения на одной платформе (обычно iOS), а затем переносят их на другую платформу, при желании можно писать общие компоненты для приложений iOS и Android.

Приложения Flutter создаются с использованием кода, специфичного для конкретной платформы, поэтому обмен кодом между приложениями для iOS и Android невозможен. Однако для упрощения повторного использования существующих “родных” компонентов можно воспользоваться библиотеками сторонних разработчиков.

Flutter или React Native легче освоить?

И React Native, и Flutter одинаково просты в освоении (с точки зрения API), хотя это зависит от опыта разработчика. Оба имеют большое и активное сообщество разработчиков, которое может помочь начинающим разработчикам и постоянно создает новые инструменты и компоненты.

Для начинающего разработчика с небольшим опытом кодирования или вообще без него мы бы, вероятно, рекомендовали начать с React Native, поскольку он поставляется с набором

предопределенных компонентов, которые можно использовать для создания приложений для iOS и Android, что означает, что вы можете изучать что-то одно за раз, не беспокоясь об изучении всех API, используемых для рендеринга представлений.

Тем не менее, мы бы, вероятно, рекомендовали выбрать Flutter вместо React Native разработчикам с некоторым опытом кодирования, поскольку API, предлагаемые Flutter, ближе к тем, которые можно найти в iOS и Android.

Кроме того, команда, стоящая за Flutter, уделяет большое внимание тому, чтобы опыт разработки, предлагаемый Flutter, мог конкурировать с опытом разработки, предлагаемым другими SDK (включая React Native).

Лучший бэкенд для Flutter

Может показаться, что Backend as a Service (BaaS), такой как Firebase от Google, будет лучшим выбором для поддержки приложения Flutter, но на самом деле все обстоит совсем иначе.



Backendless предлагает гораздо более полное и доступное решение для бэкенда приложений Flutter. Backendless Flutter SDK позволяет взаимодействовать не только с мобильными приложениями Flutter, но и с Flutter Web, предназначенным для веб-приложений и приложений для настольных компьютеров.

Backendless включает в себя все элементы, необходимые для полноценного и масштабируемого бэкенда. К ним относятся:

- Защищенная база данных реального времени
- Аутентификация пользователей и управление пользователями
- Бизнес-логика на стороне сервера (Cloud Code), включая таймеры и обработчики событий
- Возможности обмена сообщениями Pub-sub
- Push-уведомления для Android и iOS

Лучший бэкенд для React Native

В отличие от Flutter, для React Native не существует “фирменного” бэкенда. Однако, как и в случае с Flutter, приложения React Native очень эффективно интегрируются с BaaS-платформами, такими как Backendless.

Те же функции, которые Backendless предоставляет разработчикам Flutter, доступны и разработчикам React Native. Для разработчиков React Native, стремящихся максимально повысить эффективность разработки, решения BaaS могут снять с вас значительную часть задач по разработке бэкенда, позволяя сосредоточиться на создании наилучшего пользовательского интерфейса и UX.

Flutter против React Native в 2023 году

React Native появился в 2015 году, и с тех пор его используют многие компании. Мир JavaScript меняется очень быстро, поэтому

React Native со временем также эволюционировал и включал новые функции благодаря вкладу сообщества разработчиков открытого кода.

Flutter – это гораздо более новая технология, которая может показаться довольно чуждой, если вы пришли из мира Android или iOS. Google приложил много усилий, чтобы сделать ее очень простой в освоении, поэтому большинство людей, знакомых с разработкой под iOS или Android, смогут освоить ее в кратчайшие сроки.

Закрытие

Flutter и React Native являются отличными вариантами для разработки кроссплатформенных приложений. Хотя они имеют некоторые общие черты, есть и некоторые ключевые различия, о которых вы должны знать, прежде чем решить, какой из них использовать.

Выбор правильного кроссплатформенного фреймворка для вашего бизнес-приложения или стартап-приложения во многом зависит от вашего опыта разработки, команды разработчиков, а также от того, какие нативные элементы должны быть доступны вашему проекту.

Мы надеемся, что эта статья поможет вам принять обоснованное решение о том, какой фреймворк подходит для вашего следующего проекта.

Часто задаваемые вопросы

В чем разница между веб-разработкой и мобильной разработкой?

Веб-разработка и мобильная разработка создают онлайн и офлайн приложения, но делают это по-разному. Для работы мобильных приложений требуется мобильная операционная система, такая как

iOS или Android, в то время как веб-приложения могут работать на любом устройстве с веб-браузером.

Что такое Flutter?

Flutter – это фреймворк для разработки мобильных приложений с открытым исходным кодом, созданный компанией Google. Он позволяет разработчикам создавать нативные мобильные приложения для Android и iOS из единой кодовой базы.

Что такое React Native?

React Native – это фреймворк для разработки мобильных приложений с открытым исходным кодом, созданный компанией Facebook. Он позволяет разработчикам создавать нативные мобильные приложения для Android и iOS на основе единого кода.

Можно ли использовать React Native для разработки приложений для Интернета?

Нет, React Native – это фреймворк только для создания нативных мобильных приложений. Однако React Native – это мобильный аналог ReactJS, который является популярным фреймворком для создания веб-приложений.

Можно ли использовать Flutter для разработки приложений для Интернета?

Да. Flutter поддерживает использование основанных на стандартах веб-технологий, таких как HTML, CSS и JavaScript, для создания веб-контента. Благодаря поддержке веб-технологий вы можете скомпилировать существующий код Flutter, написанный на Dart, в клиентское приложение для браузера, которое размещается на любом сайте и развертывается на любом веб-сервере.