

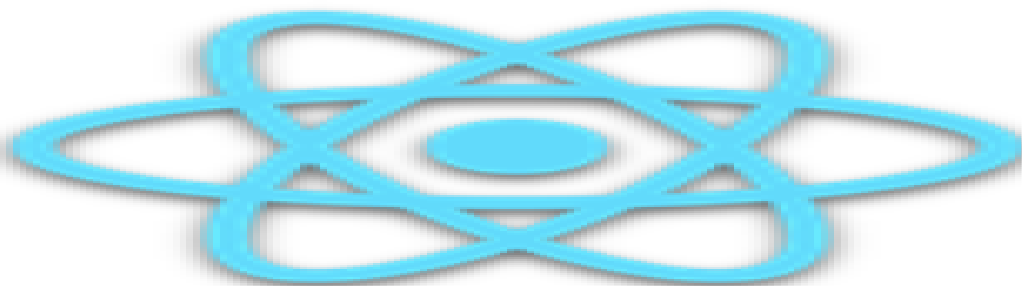


Flutter vs. React Native: Что выбрать в 2023 году?

Описание

React Native и Flutter – это технологии разработки мобильных приложений. Они помогают создавать интерактивные приложения для Android и iOS. Поскольку они обе служат одной цели, очень важно понимать различия между ними и выбрать подходящую для вас. Эта статья о Flutter vs. React Native поможет вам в этом. Теперь, когда вы знаете, что вас ждет, давайте начнем.

Что такое React Native?



React Native – это фреймворк, который объединяет лучшие части нативной разработки с React для создания пользовательских интерфейсов. Он позволяет создавать мобильные приложения, используя только JavaScript. Эти мобильные приложения можно загрузить в Google Play Store и Apple Store.

Что такое Flutter?



Flutter – это кроссплатформенная платформа для разработки мобильных приложений, разработанная компанией Google. Он предоставляет разработчикам простой способ создания и развертывания визуально привлекательных, нативно скомпилированных приложений для мобильных (iOS, Android), веб и настольных компьютеров, используя единую базу кода. Теперь, когда вы знаете, что представляют собой эти два фреймворка, рассмотрим некоторые различия между ними.

История Flutter и React Native

Flutter и React Native – самые популярные среди разработчиков кроссплатформенные фреймворки для создания мобильных приложений. Оба фреймворка произвели революцию в индустрии разработки мобильных приложений, позволив разработчикам создавать высокопроизводительные, визуально привлекательные и полнофункциональные приложения, которые легко работают на различных платформах.

- Google разработал Flutter в 2017 году, и с тех пор он стал популярным выбором среди разработчиков благодаря высокой производительности, простоте использования и обширной библиотеке готовых виджетов. Flutter использует

язык программирования Dart, разработанный для простоты изучения и использования. Кроме того, функция горячей перезагрузки Flutter позволяет разработчикам вносить изменения в код и сразу же видеть изменения в приложении, что делает процесс разработки более быстрым и эффективным.

- С другой стороны, React Native был разработан Facebook в 2015 году и широко используется разработчиками для создания кроссплатформенных мобильных приложений. React Native использует JavaScript и позволяет разработчикам создавать приложения, работающие как на устройствах iOS, так и на Android. Популярность React Native объясняется его способностью предоставлять пользователям нативный опыт, несмотря на то, что приложение создано на JavaScript.

Что касается того, какой фреймворк лучше, то это в конечном итоге зависит от конкретных потребностей проекта. Например, Flutter известен своей высокой производительностью и простотой использования, что делает его отличным выбором для создания сложных приложений. React Native, с другой стороны, обеспечивает более нативный опыт для пользователей и является идеальным выбором для создания приложений, требующих большого количества анимации и сложных взаимодействий. И Flutter, и React Native обладают уникальными достоинствами и сыграли важную роль в эволюции разработки мобильных приложений. По мере развития технологий будет интересно посмотреть, как эти фреймворки будут продолжать внедрять инновации и расширять границы возможного в разработке мобильных приложений.

Как работают кросс-платформенные фреймворки?

Фреймворки для кросс-платформенной разработки – это программные инструменты, позволяющие разработчикам создавать приложения, которые могут беспрепятственно работать на различных операционных системах и платформах.

- Эти фреймворки позволяют разработчикам написать код один раз и развернуть его на различных платформах, таких как iOS, Android и Windows. Такой подход экономит время и ресурсы, избавляя от необходимости писать другие коды для каждой платформы.
- Кроме того, эти фреймворки работают с помощью набора библиотек и инструментов, которые абстрагируют различия между платформами. Эти библиотеки позволяют разработчикам писать платформу-агностический код,

то есть он может работать на любой платформе без изменений.

Одно из главных преимуществ кроссплатформенных систем разработки заключается в том, что они обеспечивают единообразный пользовательский опыт на различных устройствах и платформах. Кроме того, они облегчают разработчикам поддержку и обновление приложений, поскольку им нужно вносить изменения только в одну кодовую базу. Существует несколько кроссплатформенных фреймворков для разработки, включая React Native, Xamarin, Flutter и Ionic. Каждый фреймворк имеет свои особенности и преимущества, и разработчики должны выбрать тот, который лучше всего соответствует их потребностям. Итак, кроссплатформенные платформы разработки – это ценный инструмент для разработчиков, желающих создавать приложения, которые могут работать на разных платформах. Они обеспечивают экономичный и эффективный способ разработки приложений, сокращают время разработки и улучшают пользовательский опыт.

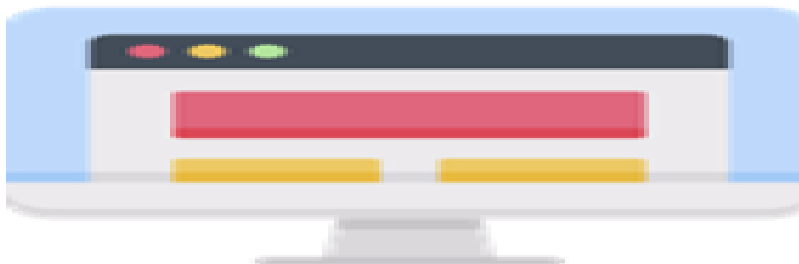
Flutter против React Native

Язык



- **React Native:** React Native полностью написан на JavaScript с использованием React. Это преимущество для React Native, поскольку JavaScript является одним из самых популярных языков в современном мире.
- **Flutter:** Flutter написан с использованием языка под названием Dart. Dart – это язык программирования, представленный Google в 2011 году и редко используемый разработчиками.

Пользовательский интерфейс



- **React Native:** React Native в большей степени основан на нативных компонентах для устройств Android и iOS. Он предлагает обширную коллекцию внешних наборов UI, которые помогают в создании красивых пользовательских интерфейсов для ваших приложений.
- **Flutter:** Flutter использует собственные визуальные, структурные, платформенные и интерактивные виджеты. Эти виджеты представляют собой встроенные компоненты пользовательского интерфейса, которые заменяют родные компоненты платформы.

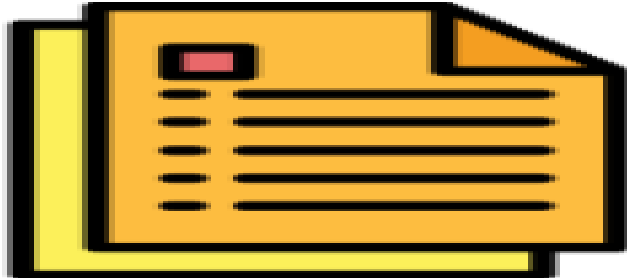
Производительность



- **React Native:** React использует JavaScript для подключения к нативным компонентам через мост. В результате скорость разработки и время выполнения медленнее, чем у Flutter.
- **Flutter:** Нет соединительного моста для инициирования взаимодействия с нативными компонентами устройства. Следовательно, взаимодействие с родными компонентами происходит быстрее, что повышает общую

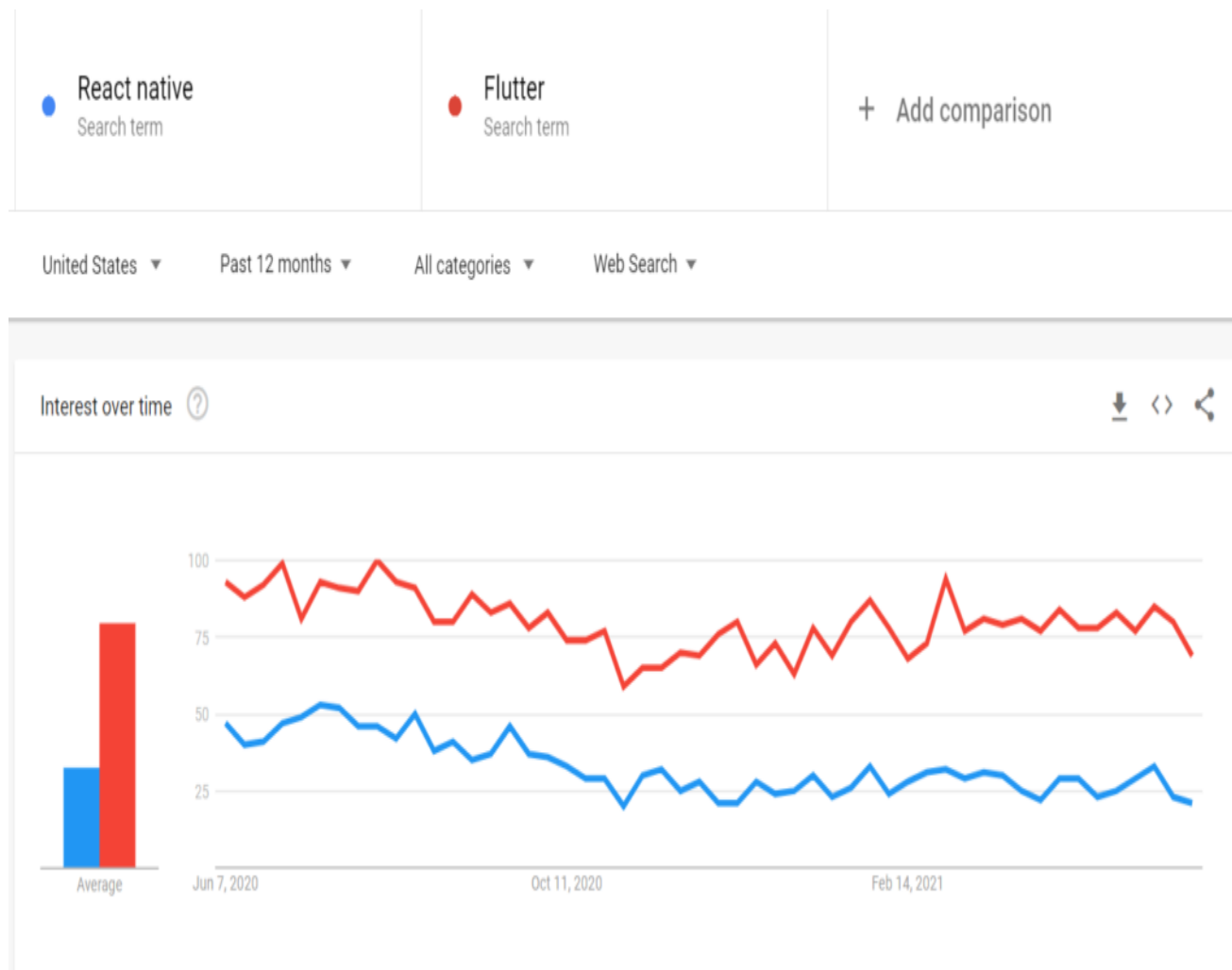
производительность приложения.

Документация



- React Native: React Native предлагает общую документацию. Фреймворк в значительной степени зависит от внешних наборов разработчиков. Документация не такая понятная, как у Flutter.
- Flutter: Он предоставляет документацию, которую легко читать, благодаря ее формату и подробному содержанию. Документация также более упорядоченная и понятная.

Популярность



- **React Native:** React Native широко распространен и является более популярным фреймворком для разработки приложений, чем Flutter. Разработчики React могут быстро создавать нативные мобильные приложения, что побуждает больше разработчиков использовать этот фреймворк.
- **Flutter:** Flutter – относительно новый фреймворк, и хотя в последнее время он привлек к себе некоторое внимание, он все еще сильно отстает по количеству разработчиков. Поскольку Dart является новым языком, уровень его принятия среди разработчиков немного ниже.

Поддержка сообщества



- React Native: React Native, выпущенный с открытым исходным кодом на GitHub в 2015 году, является самым популярным фреймворком на Stack Overflow и поддерживается обширным сообществом с более чем 310 507 отмеченными вопросами.
- Flutter: Flutter – это относительно новый фреймворк, который имеет меньшее сообщество. Тем не менее, все больше и больше разработчиков постепенно принимают его. На Stack Overflow ему задано более 89 638 вопросов.

Тенденции отрасли

- React Native: Согласно недавнему опросу StackOverflow за 2019 год, 62,5% разработчиков любят React Native. Поскольку он существует уже некоторое время и использует React и JavaScript, это приводит к лучшим возможностям трудоустройства.
- Flutter: Flutter занял более высокое место с 65,4%. Flutter сравнительно новый и используется компаниями реже. Это приводит к меньшему количеству вакансий. Однако через несколько лет это может измениться.

Плюсы и минусы Flutter и React Native Apps

Производительность нативных приложений

Плюсы:

Flutter создан с использованием нативных языков C/C++ и Dart, что делает его подходящим для создания высокопроизводительных приложений. Flutter также поддерживает множество функций аппаратного ускорения, таких как Skia, которые повышают производительность приложения. Приложения React Native обладают нативной производительностью, они очень быстрые и отзывчивые. React Native

использует нативные компоненты (UI) для создания пользовательского интерфейса приложения, что обеспечивает большую скорость работы. И этот пользовательский интерфейс обеспечивает плавную и быструю работу приложения.

Минусы:

Flutter все еще относительно новый, и его производительность не так хороша, как у других нативных платформ. Как и в любом кроссплатформенном приложении, производительность может быть не на одном уровне с полностью нативным приложением. Хотя приложения React Native быстры и отзывчивы, они все же не настолько быстры и отзывчивы, как нативные приложения. Это связано с тем, что React Native не всегда использует нативные компоненты, поэтому они не могут полностью использовать аппаратные возможности устройства.

Размер приложения**Плюсы:**

Приложения Flutter относительно небольшие по сравнению с другими кроссплатформенными фреймворками. Кроме того, приложения Flutter используют единую кодовую базу, что экономит место и упрощает сопровождение. Приложения React Native обычно намного меньше, чем нативные приложения. Эта уникальность объясняется тем, что в приложениях React Native используется технология JavaScript, которая легче нативных языков, таких как Java и Swift.

Минусы:

Приложения Flutter все еще относительно велики по сравнению с полностью нативными приложениями. Хотя приложения React Native обычно намного меньше нативных приложений, они все равно могут быть заметными.

Минимальная требуемая версия SDK:**Плюсы:**

Flutter имеет минимально необходимую версию SDK 16 и может быть скомпилирован в Android 4.1, что делает его совместимым со многими устройствами. Для приложений React Native часто требуется более низкая минимальная версия SDK, чем для других. В результате приложения на базе реакт-натив обычно более легковесны и требуют меньше кода, чем другие.

Минусы:

Минимальная версия SDK для flutter ниже, чем для других платформ мобильной разработки, что делает его менее подходящим для старых или низкопроизводительных устройств. React Native требует минимальную версию SDK 21, что означает, что он не будет работать на старых версиях Android.

Разработка пользовательского интерфейса

Плюсы:

Flutter основан на парадигме реактивного программирования, которая проще и эффективнее традиционных подходов. Кроме того, Flutter предоставляет различные компоненты и инструменты пользовательского интерфейса, что упрощает создание красивых, интерактивных и динамичных пользовательских интерфейсов. React Native часто позволяет разрабатывать пользовательский интерфейс лучше, чем другие типы приложений. Лучшая разработка пользовательского интерфейса объясняется тем, что React Native использует декларативный стиль программирования, что упрощает создание отзывчивых пользовательских интерфейсов, которые отлично выглядят на различных устройствах.

Минусы:

Компоненты пользовательского интерфейса Flutter должны иметь более подробную документацию и поддержку, чем другие платформы мобильной разработки, что затрудняет создание сложных или пользовательских элементов пользовательского интерфейса. Разработка пользовательского интерфейса в React Native может быть более сложной и трудоемкой, чем в других фреймворках, поскольку требует знания React и нативных компонентов пользовательского интерфейса.

Отладка

Плюсы:

Flutter поддерживает функцию горячей перезагрузки, которая позволяет разработчикам быстро итерировать свой код, что значительно упрощает отладку. Кроме того, Flutter имеет встроенную среду отладки, что упрощает выявление и устранение проблем с производительностью. Приложения React Native более доступны, чем традиционные нативные приложения, поскольку код более нагляден.

Минусы:

К среде отладки Flutter может быть сложно привыкнуть, поскольку она может сильно отличаться от других платформ. React Native может быть сложнее отлаживать, чем другие приложения. Кроме того, реакт натив может работать медленнее, чем нативные приложения, а некоторые функции могут быть доступны не на всех платформах.

Повторное использование кода между мобильными платформами

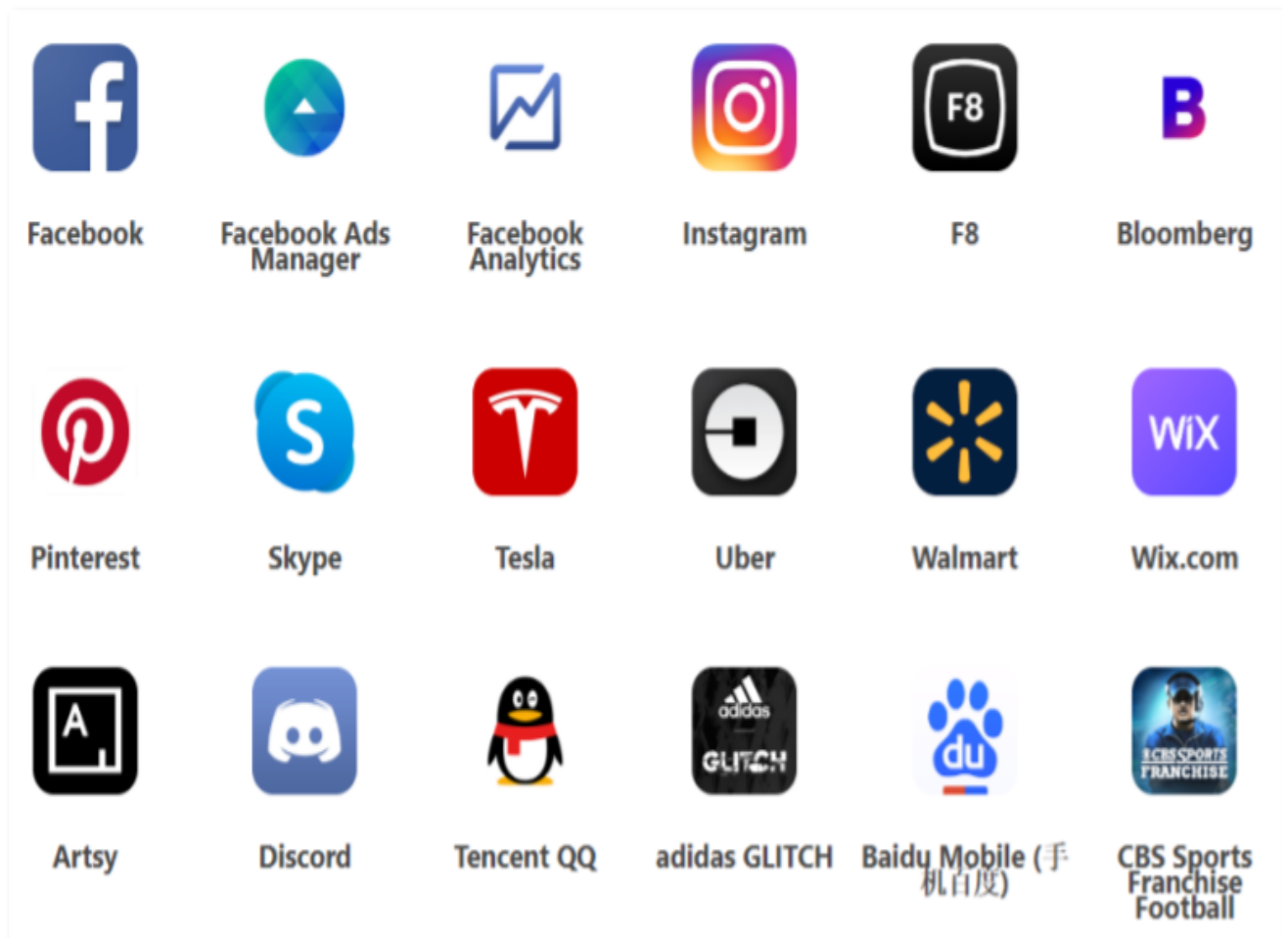
Плюсы:

Flutter позволяет разработчикам писать код на одном языке (Dart), а затем компилировать его в нативные приложения для Android и iOS. Это делает повторное использование кода между двумя платформами гораздо более доступным, что экономит время и ресурсы на разработку. React Native позволяет повторно использовать код между платформами, экономя время и усилия в процессе разработки.

Минусы:

Flutter – все еще относительно новая технология; в некоторых областях повторное использование кода может быть улучшено. В то время как в случае с React Native не хватает опытных разработчиков и ресурсов поддержки при повторном использовании кода между мобильными платформами.

Вот некоторые компании, использующие React Native.



К числу компаний, использующих Flutter, относятся.



Теперь, когда вы знаете разницу между этими двумя понятиями, для вашего удобства приводим исчерпывающую таблицу.

Характеристика

React Native

Flutter

Язык	JavaScript популярен	Dart – новый язык
Пользовательский интерфейс	Внешние наборы пользовательского интерфейса, больше вариантов пользовательского интерфейса	Готовые виджеты, меньшая настройка пользовательского интерфейса
Производительность	Сравнительно медленнее	Быстрая, поскольку позволяет избежать мостов
Документация	Дезорганизованный	Простота и обтекаемость
Популярность	Более популярны и широко распространены	Менее популярны
Поддержка сообщества	Большой	Относительно небольшой
Тенденции развития отрасли	Лучшие перспективы трудоустройства	Меньшее количество рабочих мест, которое будет увеличиваться в будущем

Что легче изучать: Flutter или React Native?

Не существует простого ответа при решении вопроса о том, что легче изучать – Flutter или React Native. Оба фреймворка имеют уникальные преимущества и недостатки, которые необходимо учитывать при определении. Однако если бы нам пришлось выбирать, мы бы сказали, что Flutter немного проще в освоении, чем React Native. Flutter использует Dart, относительно простой язык для изучения, в то время как React Native использует JavaScript, который может быть немного сложнее.

Бэкенд для Flutter и React Native

Два самых популярных фреймворка для разработки мобильных приложений – Flutter и React Native. Оба фреймворка предлагают разработчикам целый ряд преимуществ, включая быстрое время разработки, кросс-платформенную

совместимость и легкую интеграцию с внутренними системами. Но как насчет самого бэкенда? Хотя оба фреймворка можно использовать с различными технологиями бэкенда, несколько вариантов особенно хорошо подходят для каждого из них.

Бэкенд для Flutter

Для Flutter одним из популярных вариантов бэкенда является Firebase. Firebase – это платформа для разработки мобильных и веб-приложений, которая предоставляет разработчикам различные инструменты и услуги, включая базу данных реального времени, аутентификацию, хостинг и многое другое. С помощью Firebase разработчики могут быстро создавать облачные мобильные и веб-приложения, которые являются масштабируемыми, надежными и безопасными. Среди других вариантов бэкенда для Flutter – AWS Amplify, Google Cloud Platform и Parse Server.

Бэкенд для React Native

Для React Native одним из популярных вариантов бэкенда является Node.js. Node.js – это среда выполнения JavaScript, которая позволяет разработчикам создавать быстрые, масштабируемые, сетевые приложения. С помощью Node.js разработчики могут быстро создавать внутренние системы, способные обрабатывать большие объемы трафика, обрабатывать данные в режиме реального времени и интегрироваться с различными другими технологиями. Среди других вариантов бэкенда для React Native – AWS Lambda, Azure Functions и Google Cloud Functions. В конечном итоге, выбор технологии бэкенда будет зависеть от конкретных потребностей вашего приложения и ваших предпочтений как разработчика.

Конвейеры: Создание и выпуск приложения

Создание и выпуск приложения может быть сложной задачей, но конвейеры могут сделать этот процесс более плавным. Конвейеры – это набор автоматизированных шагов, которые позволяют разработчикам создавать, тестировать и развертывать свой код упорядоченным образом. Разбивая процесс на более мелкие сегменты, конвейеры позволяют командам выявлять ошибки и устранять неполадки до того, как они превратятся в серьезные проблемы. Кроме того, конвейеры помогают обеспечить надлежащее тестирование кода и его соответствие всем требованиям перед выпуском в свет. Такое тестирование экономит время и ресурсы и гарантирует высокое качество конечного продукта. Используя конвейеры,

разработчики могут выпускать приложения быстрее и эффективнее, сохраняя при этом высокое качество и надежность. Таким образом, можно сделать вывод, что в битве Flutter vs. React Native лучше использовать React Native, по крайней мере, на данный момент. Мы надеемся, что эта статья о Flutter vs. React Native внесла некоторую ясность в эти два фреймворка.

Дата Создания

06.06.2023