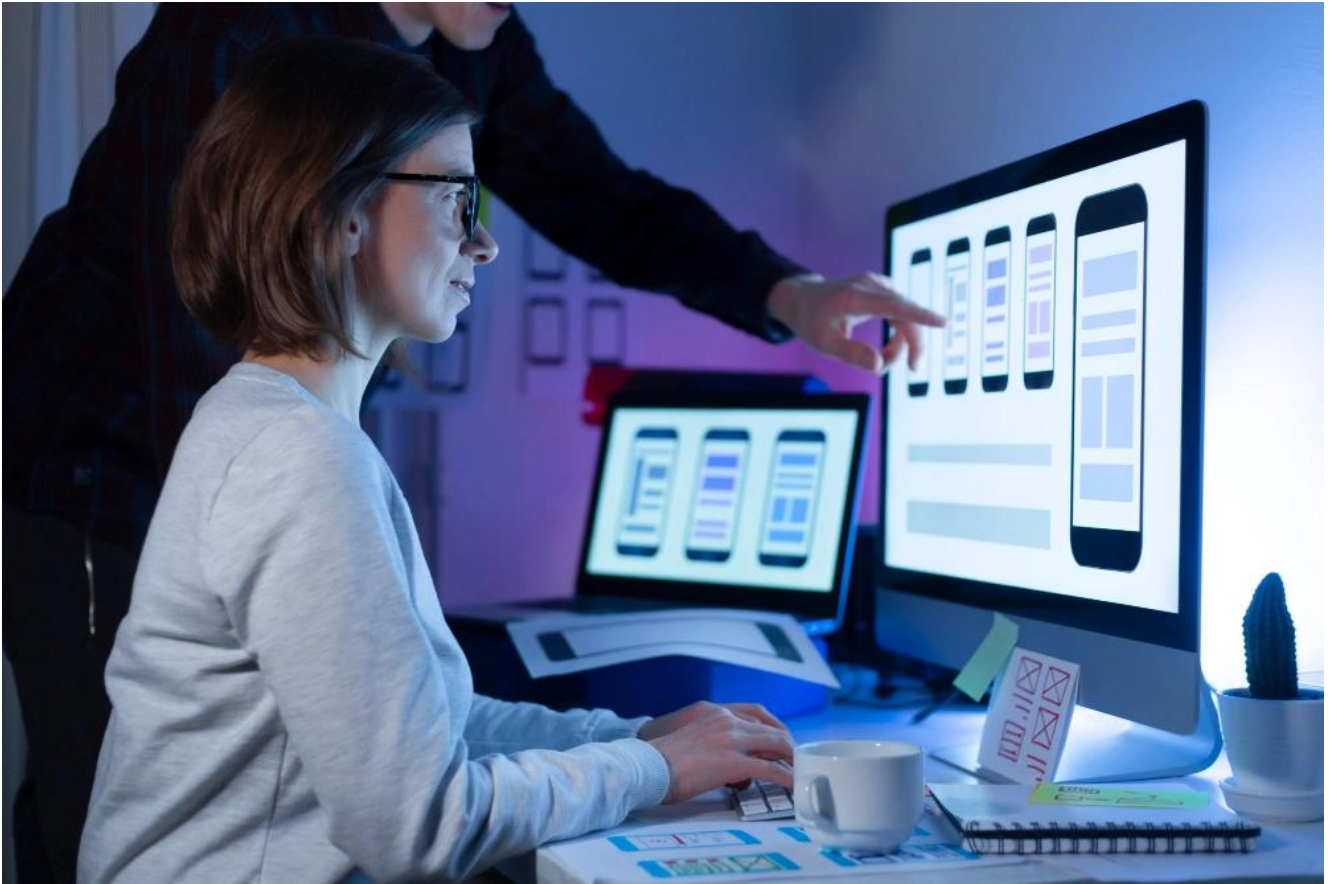


Flutter vs. React Native: Какой фреймворк будет лучше работать в 2023 году?

03.05.2023

Вы являетесь пользователем Android или iOS? Использует ли ваш ноутбук операционные системы на базе Windows, macOS или Linux? Возможно, вы хотите охватить каждого потенциального клиента, но на современном рынке есть устройства с разными операционными системами. Наличие платформы, которую можно использовать для разработки кросс-платформенных приложений, может значительно сэкономить время и ресурсы на разработку. Flutter и React Native – одни из самых больших имен на рынке кроссплатформенной разработки. Однако, если вам предложат эти два варианта, вы можете не знать, какой из них выбрать.

Почему стоит использовать кроссплатформенные решения вместо нативных?



Экономия времени

Создание функционального приложения может отнимать много времени. Если вы хотите создать приложение для пользователей iOS и Android, вам не придется создавать отдельную кодовую базу для каждого из них.

Экономия затрат на разработку и обслуживание

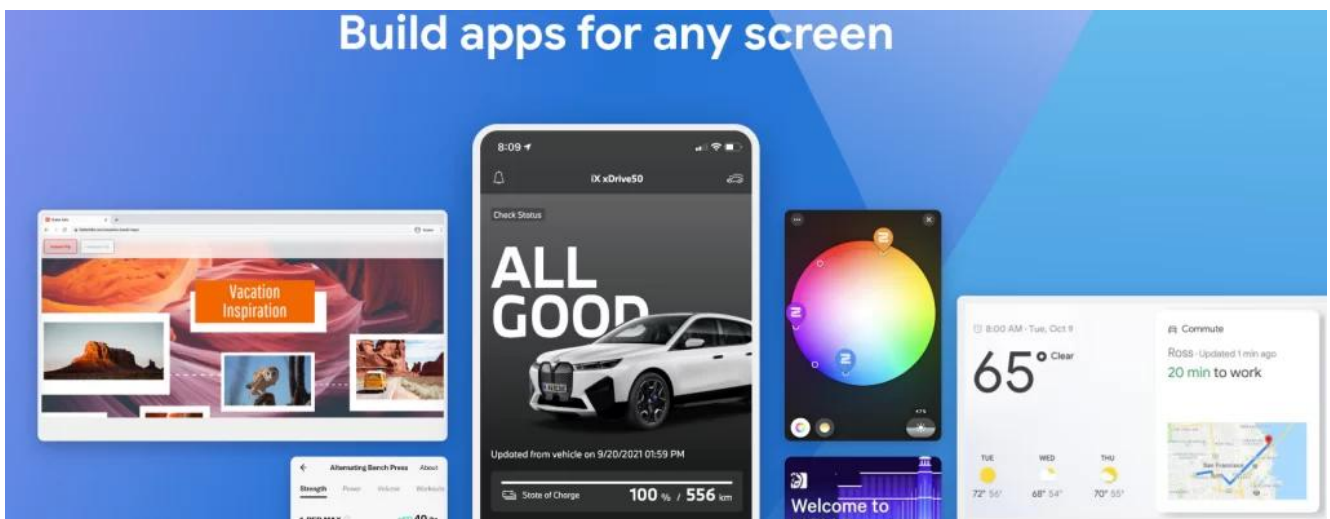
Платить разработчикам за создание разных приложений для разных пользователей может быть дорого. Стоимость эксплуатации этих приложений также высока, в зависимости от количества пользователей. Кросс-платформенная разработка позволяет использовать одних и тех же разработчиков для создания кода для разных приложений. Единственное различие возникает при доставке. Обслуживание также становится дешевле, так как вам нужно обновить только одну кодовую базу, а изменения будут действовать на всех платформах.

Производительность, близкая к нативной

Нативные приложения создаются специально для определенной операционной системы. Нативные приложения известны своей высокой производительностью. Однако некоторые кроссплатформенные решения, такие как React Native и Flutter, создают приложения, производительность которых близка к производительности нативных приложений, так что рядовые пользователи могут даже не заметить разницы. В этой статье Flutter против React Native мы рассмотрим их особенности, различия, сходства и производительность, чтобы помочь вам принять обоснованное решение.

Что такое Flutter?

Flutter – это фреймворк Dart с открытым исходным кодом, созданный компанией Google. Flutter позволяет разработчикам использовать одну и ту же кодовую базу для создания приложений для Android, iOS, настольных и веб-версий.



Вот некоторые преимущества использования Flutter:

- **Единая кодовая база для всех платформ:** Вы можете выпустить Android, iOS, настольную и веб-версии вашего приложения из одной кодовой базы.
- **Основан на компилируемом языке:** Flutter – это фреймворк Dart. Dart – это компилируемый язык, который перед

выполнением преобразует свой код в машиночитаемый, что делает Flutter быстрым.

- **Производительность, подобная нативной:** Flutter не полагается на промежуточные представления кода или интерпретаторы, поскольку использует движок Skia. Эта особенность позволяет приложениям Flutter иметь производительность, близкую к нативной.

Что такое React Native?

React Native – это JavaScript-фреймворк, созданный компанией Meta (ранее Facebook) для создания кросс-платформенных приложений. С помощью этой платформы вы можете создавать почти нативные приложения для Android и iOS.



React Native любим разработчиками по следующим причинам:

- **Возможность повторного использования кода:** React Native имеет архитектуру, основанную на компонентах. Таким образом, вы можете повторно использовать блоки кода в вашем приложении и сократить время разработки.
- **Доступность сторонних библиотек и фреймворков:** React Native имеет большое сообщество, библиотеки и фреймворки. Например, вы можете использовать такие библиотеки, как Redux, для управления состояниями в вашем приложении.
- **Живая перезагрузка:** Вы можете видеть изменения в вашем приложении по мере его создания. Вы также можете выбрать перезагрузку только определенной части кода и сэкономить

время компиляции.

- **Ощущение нативности:** React Native использует базовые компоненты операционных систем (iOS и Android), что придает приложениям нативный характер.

Flutter против React Native: Краткое сравнение

Особенности	React Native	Flutter
Язык	JavaScript	Dart
Создатель	Meta (бывший Facebook)	Google
С открытым исходным кодом	Да	Да
Сообщество	Большое и активное	Небольшое, но растущее
Примеры приложений	Wix, Soundcloud Pulse, Facebook и Facebook Ads	Alibaba, eBay, BMW, Crowdsourcing
Библиотеки сторонних разработчиков	Да	Немногочисленные, но растущие
Управление состоянием	Через Context API и различные библиотеки	Через различные пакеты/библиотеки
Горячая перезагрузка	Да	Да
Рендеринг	React Native Библиотека рендеринга	Skia

Flutter vs. React Native: Глубокое сравнение

Несмотря на то, что Flutter и React Native используются для создания кроссплатформенных приложений, они различаются по разным признакам.

Язык

Flutter и React Native – это фреймворки для разных языков программирования.

React Native

Вы можете использовать React Native с JavaScript или TypeScript. По результатам опроса Stack Overflow 2022 года JavaScript стал самым любимым языком программирования, за него проголосовали 65,36% респондентов.



TypeScript, супермножество JavaScript, занял четвертое место в том же опросе, его полюбили 34,83% респондентов.

Flutter

Flutter – это фреймворк Dart. Dart является объектно-ориентированным и используется в различных поддерживаемых

Google продуктах, таких как Flutter Engine, Flutter framework и AngularDart. В том же опросе только 6,54% респондентов предпочли работать с Dart.



Победитель

В данном случае трудно определить победителя, поскольку языки, лежащие в основе JavaScript/ TypeScript и Dart, имеют сильные и слабые стороны. JavaScript/ TypeScript имеет большое сообщество, что означает наличие большого количества библиотек. С другой стороны, Dart – компилируемый язык, то есть перед выполнением он преобразует свой код в машинный код. Эта особенность делает Dart быстрее, чем приложения на JavaScript/ TypeScript.

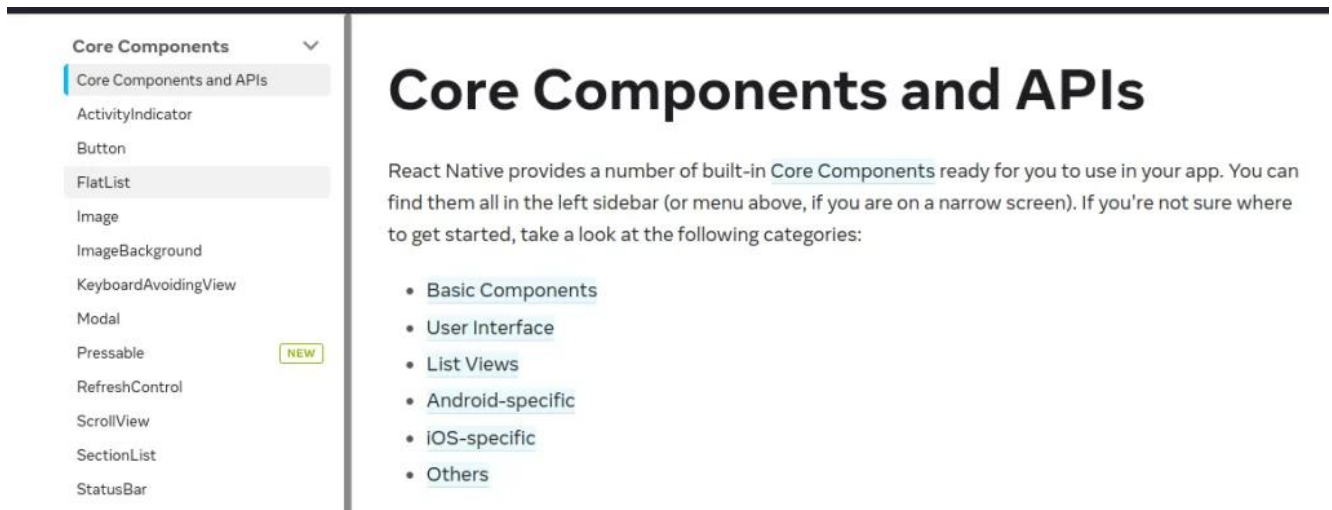
Компоненты и рендеринг

Способ рендеринга фреймворка, его компоненты влияют на производительность приложений.

React Native

Компоненты пользовательского интерфейса React Native создаются из компонентов нативной платформы (Android и iOS). Как результат, эти компоненты отзывчивы и быстры. React Native

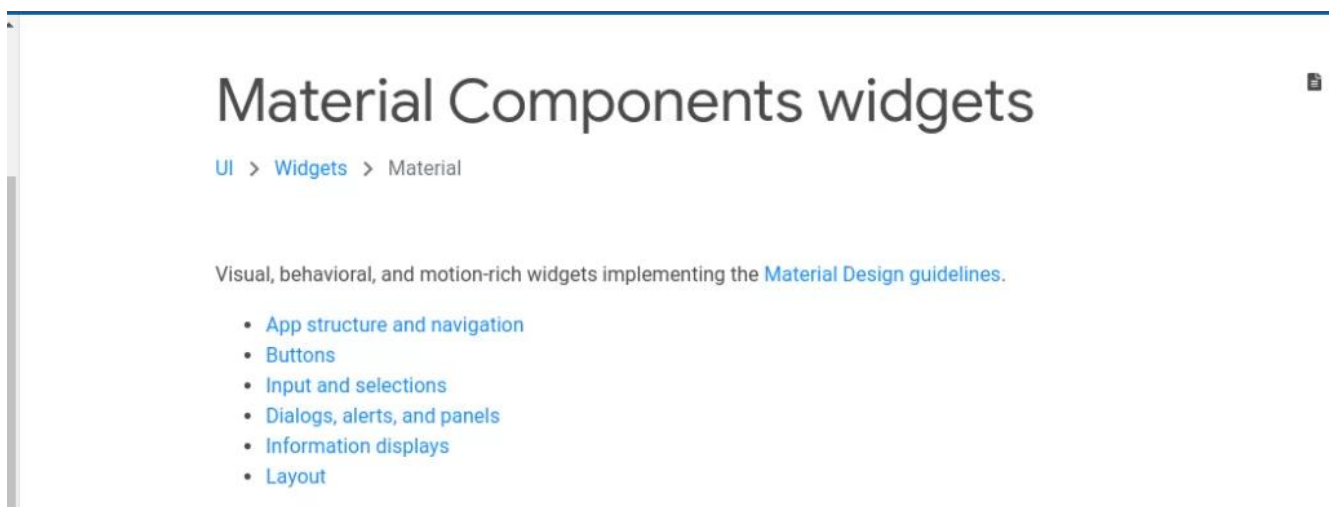
имеет различные компоненты, такие как “View”, “Image”, “Text”, “ScrollView”, “Touchable” и “TextInput”.



Приложения React Native на разных платформах могут иметь разный внешний вид, поскольку они используют компоненты пользовательского интерфейса базовой ОС.

Flutter

Flutter использует библиотеку компонентов пользовательского интерфейса, которую поддерживает Google. Эти компоненты созданы с использованием графического движка Skia, что делает их быстрыми и гибкими. Некоторые популярные компоненты пользовательского интерфейса Flutter – это виджеты Купертино и виджеты Material Design.



Пользователи также могут создавать пользовательские виджеты, переписывая существующие или создавая их с нуля. Приложения, созданные с помощью Flutter, имеют единообразный UI/UX,

независимо от операционной системы.

Победитель

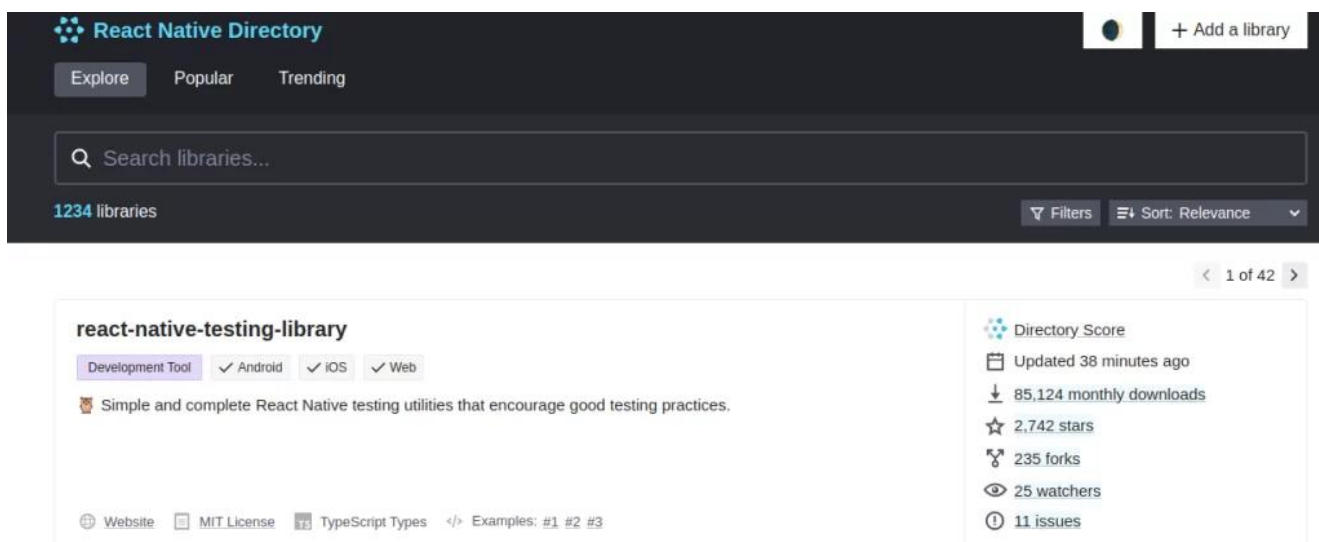
Оба фреймворка проделали хорошую работу по созданию компонентов пользовательского интерфейса. Выбор между ними зависит от опыта, вкуса и предпочтений разработчика.

Библиотеки и поддержка сообщества

Наличие библиотек и поддержка сообщества – это факторы, которые нельзя упускать из виду в сфере разработки.

React Native

React Native построен на одних из самых популярных языков программирования: поддержка JavaScript составляет 65%, а поддержка TypeScript – 35%. JavaScript существует уже более двух десятилетий и имеет множество библиотек от своего сообщества. Все библиотеки React Native доступны на сайте reactnative.directory.



The screenshot shows the React Native Directory interface. At the top, there's a search bar with the text "Search libraries...". Below the search bar, it indicates "1234 libraries" and provides options for "Filters" and "Sort: Relevance". The main content area displays a search result for "react-native-testing-library". This result includes a "Development Tool" tag, platform compatibility for "Android", "iOS", and "Web", and a description: "Simple and complete React Native testing utilities that encourage good testing practices." To the right of the description is a "Directory Score" section with various metrics: "Updated 38 minutes ago", "85,124 monthly downloads", "2,742 stars", "235 forks", "25 watchers", and "11 issues". At the bottom of the result, there are links for "Website", "MIT License", "TypeScript Types", and "Examples: #1 #2 #3".

Flutter

Flutter основан на Dart, языке программирования, популярность которого составляет менее 10%. Тем не менее, платформа имеет развивающееся сообщество, которое постоянно создает новые библиотеки. Dart был запущен в 2011 году. Все пакеты Dart и

Flutter находятся в репозитории pub.dev.



Победитель

React Native – победитель по наличию библиотек и поддержке сообщества.

Производительность

Современным пользователям очень важна производительность различных приложений.

React Native

React Native – это фреймворк JavaScript (интерпретируемый язык). JavaScript не имеет этапа компиляции, то есть ему нужен браузер, чтобы прочитать его код, интерпретировать каждую строку и запустить его.

Flutter

Flutter – это фреймворк Dart (компилируемый язык). Компилируемый язык перед выполнением преобразует код в машиночитаемый.

Победитель

Flutter выигрывает по производительности, поскольку построен на компилируемом языке. Однако разница в производительности

может быть незаметна в приложениях с минимальным набором функций.

Управление состояниями

Управление состоянием – это шаблоны или техники, которые можно использовать для управления состоянием приложения. Например, когда пользователь входит в приложение и вводит данные, состояние изменяется, и им необходимо управлять.

React Native

Существует несколько подходов к управлению состоянием в приложениях React Native. Первый подход заключается в использовании “Context”, встроенного API, который позволяет обмениваться состояниями между различными компонентами. Context подходит для небольших и средних приложений. Для больших приложений можно использовать библиотеки управления состояниями, такие как MobX и Redux.

Flutter

Flutter использует комбинацию подходов для управления состоянием. Если ваше приложение еще небольшое, для управления состоянием можно использовать такие пакеты, как Riverpod и Provider. Flutter также использует компонент бизнес-логики (паттерн BLoC) для управления состоянием. Этот подход отделяет бизнес-логику от презентационного слоя. В этом подходе для управления состоянием используются потоки и события.

Победитель

И React Native, и Flutter имеют потрясающие подходы к управлению состоянием, и победителя здесь нет. Вы также можете использовать Redux для управления состоянием в обоих случаях.

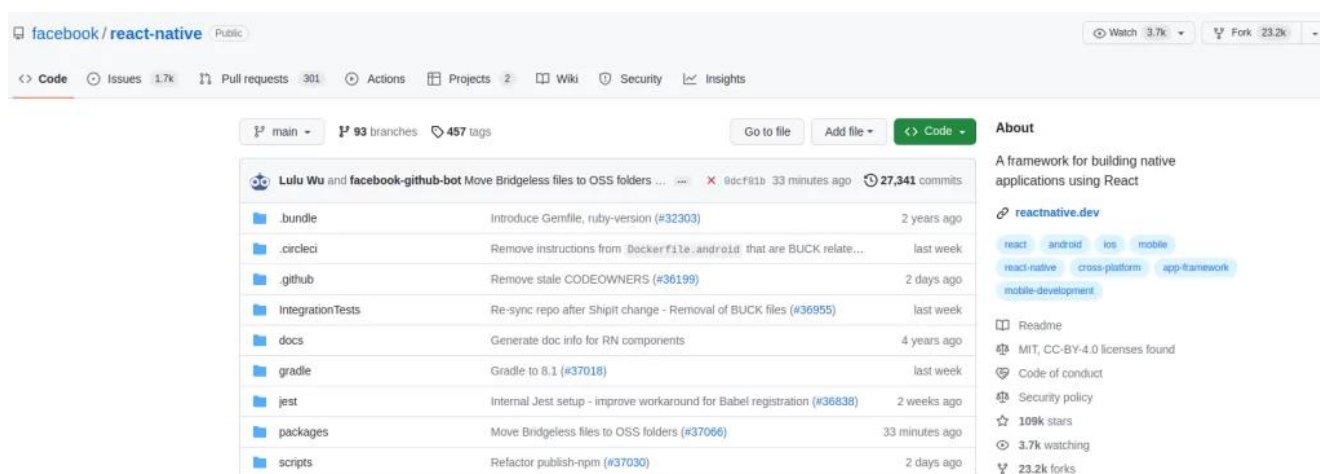
Кривая обучения

Знание того, насколько легко или сколько времени вы потратите на изучение нового фреймворка, является важным фактором для

начинающих и опытных разработчиков. Несмотря на то, что способности к обучению у всех разные, некоторые языки/фреймворки изучаются легче, чем другие.

React Native

React Native использует JavaScript, который имеет огромное количество поклонников. Этот фреймворк был создан в 2015 году и завоевал огромную популярность. У платформы 23k+ форков и 109k звезд на GitHub.



Если вы уже владеете JavaScript, изучение React Native не должно быть сложным. Ресурсы по JavaScript и React Native легко доступны, и вы можете позаимствовать у них некоторые идеи.

Flutter

Flutter использует Dart. Flutter был запущен в 2017 году и не так популярен. Несмотря на то, что изучить Dart/Flutter легко, вы можете не встретить много ресурсов по Dart, поскольку это нишевый язык. Flutter имеет более 25k форков на GitHub.

Победитель

В этой категории трудно назвать однозначного победителя. Человек, уже знакомый с JavaScript, может склониться к React Native. С другой стороны, разработчик, знакомый с Dart, скорее всего, выберет Flutter, а не React Native. Если разработчик не владеет JavaScript или Dart, то выбор кроссплатформенного фреймворка будет зависеть от личных предпочтений.

Известные бренды, использующие Flutter

Flutter использовался для создания различных мобильных приложений Google. Этот фреймворк также используется другими брендами, такими как:

- Alibaba Group
- Abbey Road Studios
- Google Play
- eBay
- CrowdSource
- 4 Pics 1 Word

Бренды, использующие React Native

React Native используется многими крупными брендами для своих приложений для Android и iOS. Вот некоторые из них:

- Facebook
- Facebook Ads Manager
- Oculus
- приложения Microsoft (Microsoft Office, Skype, Microsoft Teams и Xbox Game Pass)
- Shopify, Shopify Inbox и Shopify Point of Sale

Когда следует избегать Flutter и React Native

Кроссплатформенные платформы разработки, такие как Flutter и React Native, могут сэкономить много ресурсов и времени на разработку. Однако эти платформы не идеально подходят для всех приложений. Вот некоторые неблагоприятные случаи использования этих двух платформ;

- **Приложению необходимо использовать некоторые функции базовой операционной системы:** Вашему приложению может понадобиться использовать такие функции, как микрофон, присущие определенной операционной системе.
- **Вы хотите получить высокопроизводительное приложение:** Решение для кросс-платформенной разработки может оказаться не самым лучшим вариантом, если вам нужно приложение с высокой скоростью отклика и производительностью.

Заключение

Если вы хотите создать быстрое приложение, Flutter будет вашим лучшим выбором. Однако если вы хотите создать приложение на базе языка с большим сообществом, то вам лучше выбрать React Native. Использовать ли Flutter или React Native для своей следующей кросс-платформы, зависит от вашего понимания основного языка, типа приложения, которое вы хотите создать, вашего вкуса и предпочтений. Разработчики JavaScript, скорее всего, будут использовать React Native, в то время как программисты Dart предпочтут Flutter.