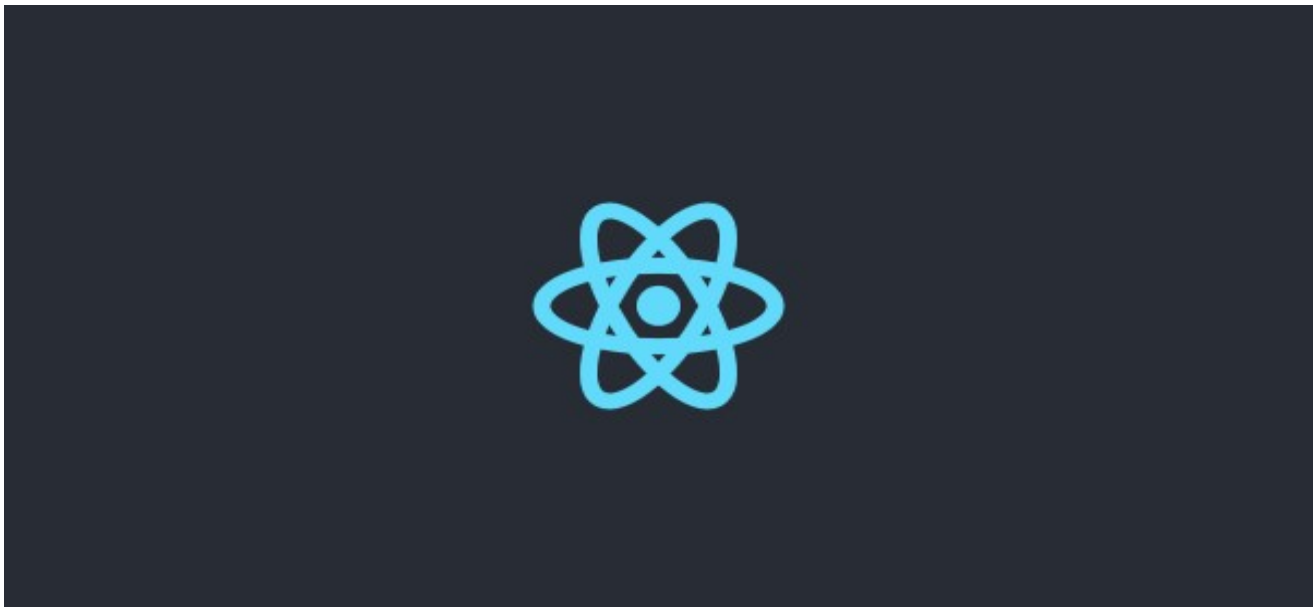
Объяснение разницы между React и React Native

Описание

Мы, люди, путаемся во многих вещах. Путаница может возникать на основе различных характеристик вещей. Самый распространенный способ путать различные вещи – это их названия. Сначала люди знакомятся с названиями новых вещей. Будь то человек, животное, продукт, программное обеспечение и т.д., они будут искать их по названиям, а затем узнают о различных характеристиках, применении, истории и т.д. Зачем все это нужно? Да, люди часто путаются в терминах (названиях) React и React Native.

Если не технари видят названия React и React Native, то в большинстве случаев они воспринимают Reactive Native как расширение React. Даже некоторые технари могут думать так же, если у них 0% знаний в этих областях. Что же это такое на самом деле? Почему люди часто путают именно их, а не что-то другое? React и React Native – это два фреймворка. Названия похожи, но между ними есть лишнее слово. Поэтому люди часто путаются из-за их названий на первый взгляд. Если у вас такая же путаница, то вы находитесь в правильном месте, чтобы раскрыть тайну, скрывающуюся за ними. Давайте разберемся.

React



React – это JavaScript-библиотека, используемая для создания одностраничных веб-приложений. Это одна из самых популярных библиотек для создания пользовательских интерфейсов (frontend) в Интернете. Пожалуй, можно сказать, что на сегодняшний день это самая популярная библиотека. Она создана и поддерживается компанией Facebook. Она также известна как ReactJS. Поскольку у меня есть опыт работы с React, я могу сказать, что она красива и проста в освоении и создании. Это библиотека. Поэтому, используя ее возможности, мы можем создавать все, что захотим и как захотим. При разработке приложений на React нет строгих правил, которым нужно следовать. Таким образом, мы получаем свободу. У React очень много интересных возможностей. Давайте рассмотрим их.

Компоненты

В React все является компонентом. Это как строительный блок веб-приложения. Мы можем формировать большие компоненты, объединяя маленькие. Каждый компонент имеет собственное состояние и управление. Компоненты управляют пользовательским интерфейсом и решают, что показывать пользователям, исходя из своего состояния. Компоненты – это все в React.

И они многократно используются. Напишите один раз и используйте его везде. Мы должны писать компоненты с особой тщательностью. Это облегчает сопровождение при росте приложения. Если мы напишем много кода в одном компоненте, то со временем его поддержка станет для нас непосильной ношей. Компоненты React должны быть маленькими и милыми. Они превращают жизнь

разработчиков как в рай, так и в ад.

Виртуальная DOM

Вы наверняка видели что-то вроде загрузчика внутри кнопки. А на платформах социальных сетей количество лайков увеличивается сразу после нажатия. В ранние времена существования Интернета нам приходилось перезагружать все, чтобы получить информацию. Но теперь один элемент, который необходимо обновить, обновляется сам, не затрагивая другие элементы. Для чего все это нужно? Как мы уже видели, в React все является компонентом. Браузеры поддерживают структуру DOM для элементов веб-приложения.

Когда часть веб-приложения нуждается в обновлении, мы должны обновить ее с помощью манипуляций с DOM. React делает то же самое эффективно. React создает виртуальный DOM (копию DOM) для всех своих компонентов. Чтобы обновить что-либо в веб-приложении, React сравнивает реальный DOM с виртуальным DOM. Если есть какие-либо изменения, то React запускает обновление компонента.

Односторонний поток данных

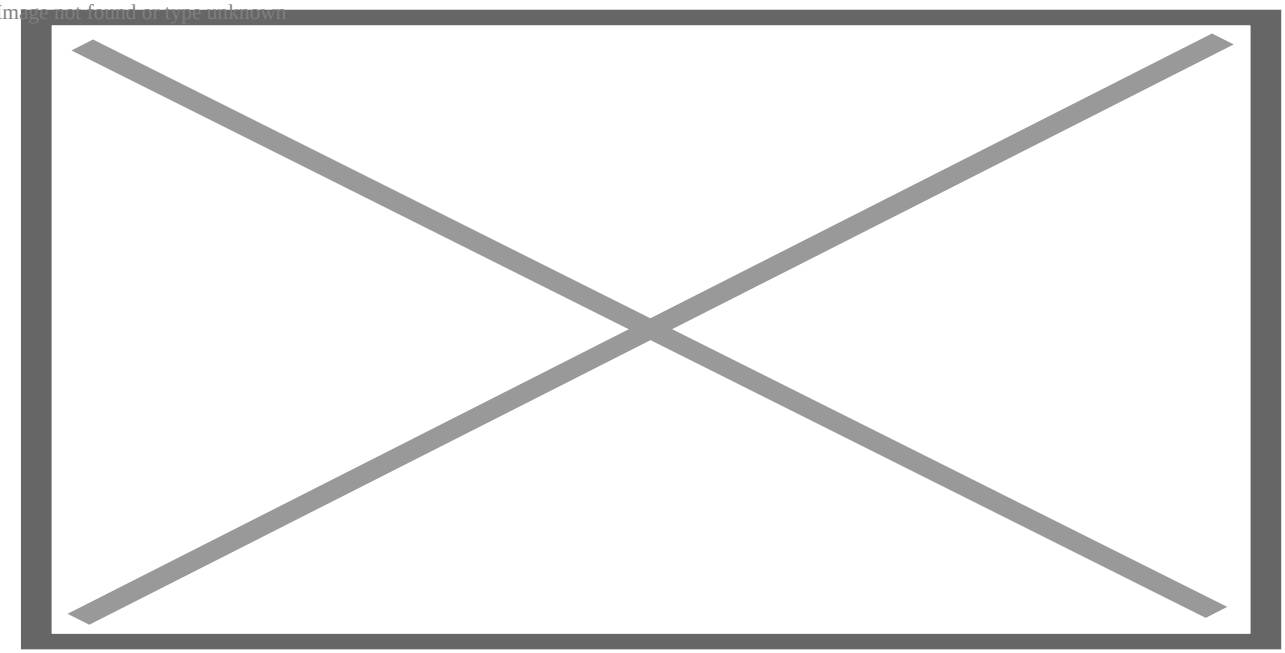
Мы не можем разделить большой набор компонентов на более мелкие компоненты без потока данных. Должен быть какой-то способ передачи данных между компонентами. React позволяет передавать данные от одного компонента к другому компоненту в одном направлении. Данные передаются от родительских компонентов к дочерним компонентам. При этом дочерние компоненты не могут обновлять данные. Нет возможности отправить данные обратно в родительский компонент, поскольку поток данных однонаправленный. Сначала можно подумать, что это не разнонаправленный поток данных. Но однонаправленный поток данных дает нам больше контроля над многонаправленным потоком данных.

Обзор

Существует множество других возможностей, таких как JSX, условный рендеринг и т.д.; они вторичны. Мы рассмотрели основные возможности библиотеки React. Если говорить о применении React, то с ее помощью можно создать практически любой тип веб-приложения. Сообщество разработчиков React очень велико. Вы можете найти множество пакетов для работы с React.

React Native

Image not found or type unknown



React Native – это JavaScript-фреймворк, используемый для разработки кроссплатформенных мобильных приложений. Он также создан и поддерживается компанией Facebook. Большинство из вас будут удивлены приведенным выше утверждением. Неужели мы можем создавать мобильные приложения для Android и IOS с помощью одного фреймворка? Если вы не следите за обновлениями в мире технологий, то вряд ли вам это известно. Да, мы можем создавать кроссплатформенные (как для Android, так и для IOS) приложения с помощью React Native.

Существуют и другие фреймворки для разработки кроссплатформенных приложений. React Native – один из самых популярных в своем роде. Не самый популярный из-за ограничений JavaScript в нативных приложениях. Но он блистает в своих областях разработки. Его используют даже такие крупные компании, как Facebook, Instagram, Flipkart и т.д. Это не значит, что вы должны использовать его. Это значит, что мы можем создавать кроссплатформенные приложения производственного уровня с помощью React Native.

В приведенном выше абзаце я использовал группу слов под названием “нативные приложения”. Что они собой представляют? Это не какой-то новый тип приложений. Нативное приложение создается специально для конкретной

платформы. Android-приложения для android-мобильников, IOS-приложения для iPhone-мобильников, Windows-приложения для Windows и т.д.., Что происходит с нативным приложением в React Native?

В React Native создается нативное приложение, которое подходит как для Android, так и для IOS, исходя из наших пожеланий. Приложения, разработанные с помощью React Native, являются нативными, как и Android Studio для Android и аналогично для IOS. Возможно, из-за этого создатели назвали его React Native. Не факт. Если говорить о возможностях React Native, то здесь нас ждет целая куча возможностей. Давайте рассмотрим некоторые из них.

Кроссплатформенность

Мы можем создавать мобильные приложения для Android и IOS одновременно с единой кодовой базой. Это значительно экономит время и деньги компаний.

Горячая или живая перезагрузка

Если у вас есть опыт работы с приложениями на React или React Native, то вы наверняка знаете об этой функции. Эта функция перезагружает все приложение с новыми обновлениями, когда мы меняем код. Нам не нужно нажимать кнопку перезагрузки каждый раз, когда мы меняем код. Обновите код и посмотрите на изменения. Вот и все. Нам не нужно ничего ждать, если только не возникнет ошибка. Вам может показаться, что это второстепенная функция. Но если вы занимаетесь андроидной разработкой без какого-либо фреймворка, вы поймете ценность этой возможности в React Native.

Библиотеки пользовательского интерфейса и сообщество

В React Native есть множество встроенных нативных компонентов. Мы можем использовать их напрямую, без дополнительной настройки или установки. Нативные компоненты выглядят как родные для соответствующих платформ. Пользовательский интерфейс приложений React Native совпадает как с родным пользовательским интерфейсом IOS, так и с родным пользовательским интерфейсом Android. В React Native есть компоненты, аналогичные React. Что касается сообщества. Оно большое и продолжает расти. Вы можете без труда получить помощь от сообщества, если застряли в нем.

Обзор

В Интернете можно найти множество других возможностей React Native. Изучите их, если вы собираетесь разрабатывать мобильные приложения. Фронтенд-разработчик также может разрабатывать нативные приложения с использованием React Native. Это упрощает разработку кроссплатформенных мобильных приложений.

React и React Native

Между React и React Native существуют некоторые сходства и различия. Давайте рассмотрим их. Если говорить о приложениях React и React Native, то они полностью отличаются друг от друга. Но если говорить о принципах, то они похожи. И в React, и в React Native есть компоненты. И при разработке они руководствуются одними и теми же принципами. Оба они используют для разработки язык JavaScript. Давайте посмотрим на простое приложение Hello, World в обоих вариантах.

React

```
import React, { Component } from 'react';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="container">
        <h1>Hello, Wolrd!</h1>
      </div>
    );
  }
}

export default App;
```

React Native

```
import React from 'react';
import { Text, View } from 'react-native';

const App = () => {
  return (
    <View
      style={{
        flex: 1,
        justifyContent: "center",
```

```
        alignItems: "center"
      }}>
      <Text>Hello, world!</Text>
    </View>
  )
}

export default App;
```

Как видно, оба они используют пакет React. Синтаксис в обоих приложениях выглядит одинаково, поскольку они используют специальную разметку JSX. Но когда дело доходит до рендеринга, оба приложения используют разные вещи. React использует виртуальный DOM, а React Native – Native API для рендеринга пользовательского интерфейса. Для управления утверждениями в React-приложениях существуют некоторые внешние пакеты, такие как Redux, MobX и т.д.. Эти же пакеты могут быть использованы и в приложениях React Native. И React, и React Native используют JavaScript. Таким образом, мы можем использовать практически все пакеты JavaScript в обоих приложениях. Это добавляет множество пакетов в обе библиотеки пакетов. React и React Native связаны друг с другом. Но используются они для разных целей.

Заключение

React и React Native отличаются по конечному продукту и платформам приложений. Но они следуют схожим принципам при разработке соответствующего приложения. Если вы сможете освоить один из этих двух фреймворков, React или React Native, вы сможете ускорить изучение другого. Однако для разработки приложений React Native необходимо знание React. Но для этого его недостаточно. Нужно больше знать о разработке нативных приложений, так как в React Native поддержка ограничена.

Будем надеяться, что в будущем она разовьется до полной поддержки. Если вы хотите начать веб-разработку или мобильных приложений, то выбор React или React Native, безусловно, принесет вам пользу в будущем. Однако это не является обязательным. Изучение концепций React не составит труда, если вы знаете JavaScript. Официальная документация станет для вас отличным источником информации для начала работы с React или React Native.

Дата Создания
31.07.2023