

Nghiên cứu công nghệ lập trình fron-end React JS

Trình bày: Nguyễn Thị Sim

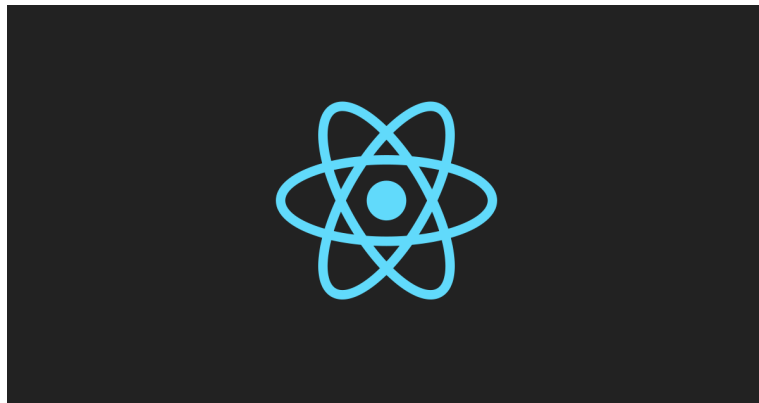
Time: 2019 October

Table of Contents

Giới thiệu về React JS	3
Core feature của React JS	4
Tìm hiểu Component trong React JS	4
React Router	11
Ưu điểm và nhược điểm của React JS	12
Lợi ích của React JS	13
Tại sao chọn React JS để phát triển.....	13
Các trang web tốt nhất để học ReactJS.....	15
So sánh công nghệ React JS và Angular JS	16
Thực hành tạo React App và cài đặt thư viện.....	18
Tổng kết	23

Giới thiệu về React JS

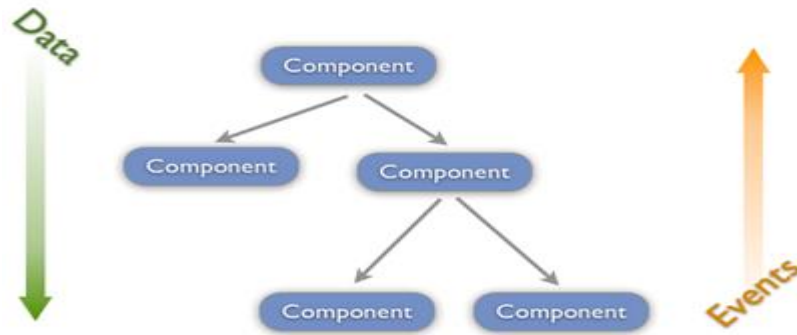
React JS là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. React là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.



Lịch sử React.JS, phiên bản hiện tại của React.JS là V16.8.6 (tháng 3 năm 2019). Phát hành lần đầu ra công chúng (V0.3.0) là vào tháng 7 năm 2013. React.JS được sử dụng lần đầu tiên vào năm 2011 cho tính năng Newsfeed của Facebook. Kỹ sư phần mềm Facebook, Jordan Walke, đã tạo ra nó. Gói phiên bản ứng dụng tạo phản ứng 2.0 được phát hành vào tháng 10 năm 2018. Tạo-Reac-app phiên bản 2.0 hỗ trợ Babel 7, webpack 4 và Jest23.

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và www.instagram.com được viết hoàn toàn trên React.

React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi.



React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện.

Core feature của React JS

Dưới đây là các tính năng cốt lõi minh họa tại sao ReactJS là một thư viện JavaScript hoàn toàn khác với các thư viện khác và các thành phần giúp nâng cao các hoạt động đa dạng của nó theo cấp độ:

JSX - Thay vì JavaScript thông thường, React sử dụng JSX để tạo template. Vì JSX cho phép trích dẫn HTML dễ dàng và sử dụng cú pháp HTML, nên nó giúp hiển thị các thành phần con(subcomponent).

React Native - Vào năm 2015, Facebook đã công bố một số thư viện native(gốc) của React, để các nhà phát triển có thể phát triển các ứng dụng đa dạng với các chức năng front-end mong muốn. Nó có một bộ các component UI có thể tái sử dụng cho cả nền tảng Android và iOS. Sử dụng React native, bạn có thể phát triển bất kỳ ứng dụng di động nào với giao diện gốc.

Virtual Document Object-Model - Mô hình Virtual Document Object-Model của React cho phép tạo cấu trúc dữ liệu cho các hoạt động chức năng khác nhau. Nó giúp thay đổi computation cũng như cập nhật trình duyệt. Điều này tạo điều kiện cho các lập trình viên mã hóa các ứng dụng với các logic mong muốn. Thư viện React kết xuất mỗi trang cho mỗi thay đổi. Trong khi về mặt logic, các thành phần được thay đổi chỉ được hiển thị.

Tìm hiểu Component trong React JS

Trong React JS, mỗi đoạn code sẽ được phân chia thành những component không lệ thuộc lẫn nhau và có thể tái sử dụng khi cần thiết.

Để khởi tạo Component chúng ta sẽ có nhiều cách như:

- Dùng let, const trong JavaScript ES6 hoặc var ở các phiên bản thấp hơn của JavaScript.
- Dùng Function
- Dùng Class trong JavaScript ES6.

Ví dụ:

```
import React, { Component } from 'react';

let Tab_sim={()=>{
  return(
    <div>Khởi tạo bằng let</div>
  )
};

const Tab_sim={()=>{
  return(
    <div>Khởi tạo bằng const</div>
  )
};

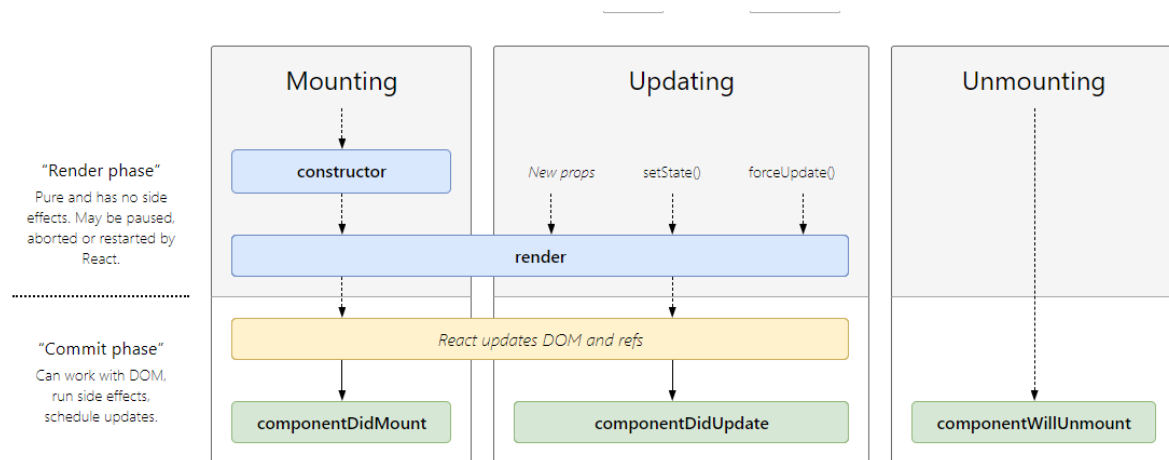
function Tab_function(){
  return(
    <div>Khởi tạo bằng Function</div>
  )
}

class Tab_class extends Component{
  render(){
    return(
      <div>Khởi tạo bằng Class</div>
    )
  }
}
```

Từ ví dụ trên, ta thấy khi dùng let hoặc const, chúng ta cần dùng Arrow Function hoặc Anonymous Function để gán cho biến hoặc hằng. Function có cách dùng tương tự cách định nghĩa biến và hằng. Class phải kế thừa từ Class Component trong React JS để có thể chạy được.

Các phương thức trong React Component Lifecycle

Các React Lifecycle Methods quan trọng các phương thức chính cps thể chia làm 3 pha chính là: Mounting, Updating và Unmounting.



1) Mounting

Mounting là giai đoạn khi React Component được tạo ra và render lên trên DOM tree. Các methods được gọi trong giai đoạn này lần lượt là: `constructor()`, `static getDerivedStateFromProps()`, `render()`, `componentDidMount()`. Khi Component được khởi tạo và chèn vào DOM thì React sẽ follow theo trình tự như sau :

1. Khởi tạo class đã kế thừa từ Component
2. Khởi tạo giá trị mặc định cho Props
3. Khởi tạo giá trị mặc định cho State
4. Gọi hàm `componentWillMount()`
5. Gọi hàm `render()`
6. Gọi hàm `componentDidMount()`

Nói chung, **`constructor()`** luôn là phương thức được gọi đến đầu tiên mỗi khi khởi tạo. Mục đích sử dụng phương thức này là khởi tạo state cho React Component và Bind method(xử lý event, sử dụng trong `setTimeout` hoặc `setInterval`) với **`this`**.

```
constructor(props){
  super(props);
  this.state={snackbaropen:false,snackbarmsg:''};
  this.handleChange = this.handleChange.bind(this);
  this.handleSubmit=this.handleSubmit.bind(this);
}
```

Trong đó:

`Super(props)` dùng để gọi đến hàm khởi tạo của cha `React.Component` nếu component khai báo được kế thừa từ `React.Component`. Nếu thiếu thì `this.props` sẽ là `undefined`.

`This.state={...}` : khởi tạo biến thuộc state, ví dụ biến `snackbaropen= false`.

Phương thức **`render()`** là phương thức duy nhất bắt buộc phải có đối với `React Component` và có cấu trúc như sau:

```
render() {  
    return (  
        /* Định nghĩa cấu trúc Component tại đây */  
    )  
}
```

Phương thức này dùng để miêu tả cấu trúc của Component sau khi nó được chèn vào DOM tree. Nó bắt buộc được gọi lần đầu tiên để chèn component vào HTML, và có thể được gọi lại để cập nhật giao diện mỗi khi state của Component thay đổi.

Phương thức **`componentDidMount()`** được gọi một lần duy nhất ngay sau khi Component được render xong. Trong phương thức này có thể lấy dữ liệu từ server để cập nhật state cho Component, định nghĩa interval thông qua `setInterval` để thực hiện một số nhiệm vụ lặp lại, đăng kí sự kiện.

2) Updation

Updating là giai đoạn khi React Component cần cập nhật giao diện mỗi khi **props** hoặc **state** của nó thay đổi. Một bản cập nhật có thể được thực hiện bởi những thay đổi về props hoặc state. Các phương thức này được gọi theo thứ tự sau khi một thành phần đang được kết xuất lại:

- `static getDerivedStateFromProps()`
- `shouldComponentUpdate()`
- `render()`
- `getSnapshotBeforeUpdate()`
- `componentDidUpdate()`

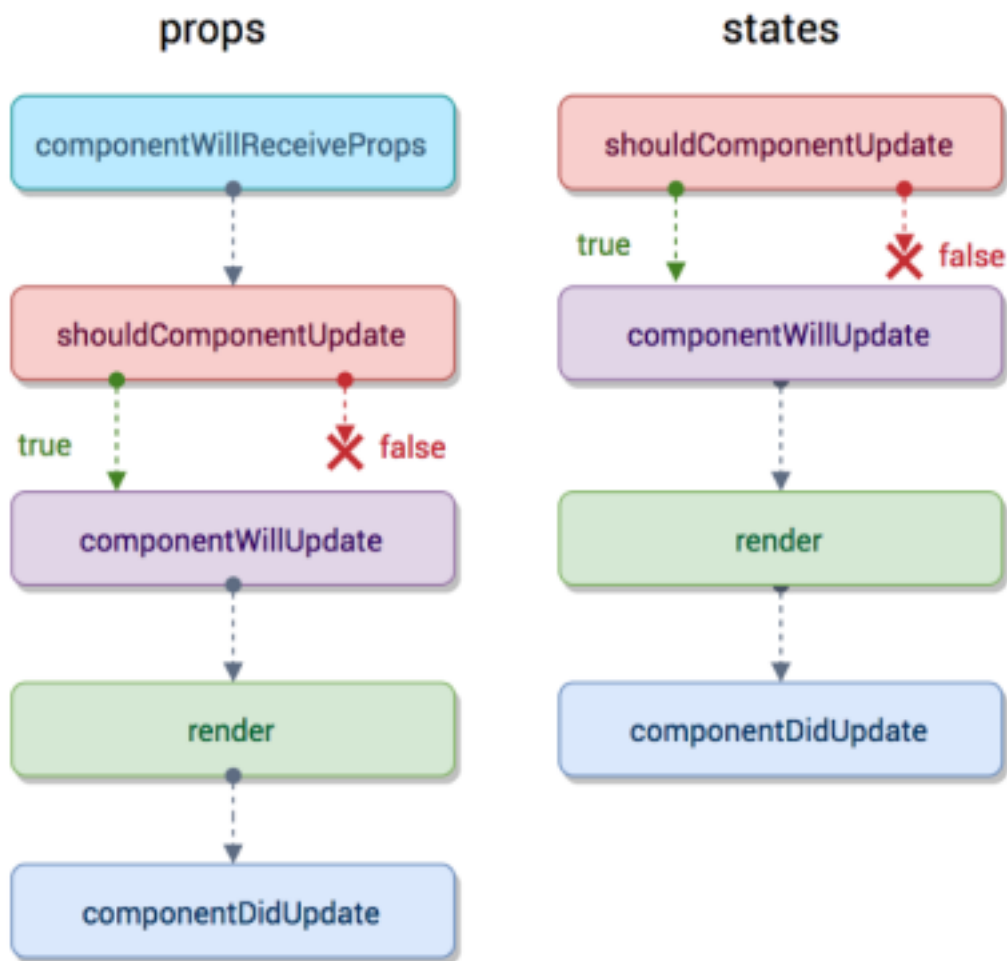
Phương thức **`render()`** trong pha Updating có thể được gọi hoặc không, phụ thuộc vào phương thức `shouldComponentUpdate()`. Mặc định phương thức `shouldComponentUpdate()` sẽ trả về **true**.

Phương thức **componentDidUpdate()** được gọi sau khi việc update kết thúc- component với những dữ liệu mới đã được cập nhật xong lên giao diện. Ví dụ, khi thêm mới dữ liệu bằng modal bạn muốn danh sách đối tượng luôn luôn được update khi có bất kì sự thay đổi nào thì dùng componentDidUpdate().

```
componentDidUpdate(){  
    this.refreshList();  
}
```

Trong đó: refreshList() là hàm gọi dữ liệu từ server.

Updation



Props giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến. Khi `Props()` thay đổi:

- `componentWillReceiveProps()`: Chạy khi component con nhận props từ component cha. Sau khi nhận được props mới từ component cha rồi có thì component con có thể set lại state.

- `shouldComponentUpdate()`: Hàm này có thể nói là nó tăng hiệu năng của React lên. Nếu như `return false` thì các phương thức `componentWillUpdate`, `render`, `componentDidUpdate` sẽ không được chạy nữa (vì mặc định nó `return` về `true` để chạy được 3 hàm tiếp theo, nhiều trường hợp mình không cần chạy 3 hàm tiếp theo).
- `componentWillUpdate()`: Hàm này cũng giống như hàm `componentWillMount()` trước khi re-render ra Component. Nhưng chúng mình hầu hết không tương tác gì nhiều lắm trong hàm này, hàm `setState` hầu hết chúng mình sẽ sử dụng trong hàm `componentWillReceiveProps`
- `componentDidUpdate()`: hàm này được gọi đến sau khi đã re-render lại hay React đã cập nhật lại UI, nếu mà chúng ta muốn chạy animation thì đây chính là lúc chúng ta nên gọi trong hàm này.

State thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI. Khi State thay đổi:

1. Cập nhật giá trị cho state
2. Gọi hàm `shouldComponentUpdate()`
3. Gọi hàm `componentWillUpdate()` – với điều kiện hàm trên `return true`
4. Gọi hàm `render()`
5. Gọi hàm `componentDidUpdate()`

3) Unmounting

Quá trình unmounting xảy ra khi component bị remove khỏi DOM tree, hay nói một cách khác là hàm `componentWillUnmount()` sẽ được gọi khi render ra không có component nào hoặc người dùng chuyển hướng trang web. Trong giai đoạn này, chỉ có một phương thức được gọi duy nhất là `componentWillUnmount()`.

Vậy về React Component Lifecycle cần quan tâm là:

- Mounting
 - `Constructor()`
 - `Render()`
 - `componentDidMount()`
- Updating
 - `shouldComponentUpdate()`
 - `render()`
 - `componentDidUpdate()`

- UnMounting
 - componentWillMount()

React Router

React Router là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React Router cho phép bạn định tuyến "luồng dữ liệu" (data flow) trong ứng dụng của bạn một cách rõ ràng.

Ý tưởng của Router hữu ích và React là một thư viện Javascript để lập trình các ứng dụng một trang (Single Page Application). Để phát triển React cần viết nhiều component nhưng lại chỉ cần một tập tin duy nhất để phục vụ người dùng, đó là index.html.

React Router giúp định nghĩa các URL động, và lựa chọn Component phù hợp để hiển thị trên trình duyệt người dùng ứng với từng URL.

React Router cung cấp 2 thành phần chính là **<BrowserRouter>** và **<HashRouter>**. Hai thành phần này khác nhau ở kiểu URL mà chúng sẽ tạo ra và đồng bộ.

// <BrowserRouter>

<http://example.com/about>

// <HashRouter>

<http://example.com/#/about>

Thành phần <Route> định nghĩa một ánh xạ giữa một URL và một Component. Nghĩa là khi người dùng truy cập theo một URL trên trình duyệt, một component tương ứng sẽ được render trên giao diện.

```
<BrowserRouter>
  <Route path="/" component={Home} exact />
  <Route path="/authors" component={Authors} />
  <Route path="/categories" component={Categories} />
</BrowserRouter>
```

Thuộc tính exact được sử dụng để nói rằng <Route> này chỉ được hoạt động nếu URL trên trình duyệt phù hợp tuyệt đối với giá trị của thuộc tính path của nó.

Ưu điểm và nhược điểm của React JS

1. Ưu điểm

- Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
- Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.
- Render tăng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dung sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Vì vậy, Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
- Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
- Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

2. Nhược điểm

- Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax
- Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
- React khá nặng nếu so với các framework khác React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh
- Khó tiếp cận cho người mới học Web

Lợi ích của React JS

React giúp tạo UI tương tác đơn giản. Thiết kế các khung hình đơn giản cho từng trạng thái trong ứng dụng của bạn và React sẽ cập nhật và render đúng các thành phần phù hợp khi dữ liệu của bạn thay đổi.

Xây dựng các component và quản lý các trạng thái của riêng chúng, sau đó kết hợp chúng để tạo các UI phức tạp.

Các câu khai báo làm cho mã của bạn dễ sử dụng hơn và dễ tìm lỗi hơn.

Vì component logic được viết bằng JavaScript thay vì các mẫu, bạn có thể dễ dàng truyền dữ liệu phong phú qua ứng dụng của mình và tránh thao tác với DOM.

Các thành phần React thực hiện một phương thức `render()` lấy dữ liệu đầu vào và trả về những gì sẽ hiển thị. Dữ liệu đầu vào được truyền vào thành phần có thể được truy cập bằng `render ()` qua `this.props`.

Tại sao chọn React JS để phát triển

Dưới đây là những lý do khiến ReactJS ngày càng được sử dụng bởi nhiều công nghệ lớn và những lợi thế nào sẽ mang lại cho ứng dụng front-end:

1. Tập trung chủ yếu vào giao diện người dùng (Primarily Focused on User Interface)

ReactJS tập trung nhiều hơn vào giao diện người dùng, không giống như Meteor JS và AngularJS. Nó cung cấp cho người dùng một giao diện phản hồi cao với sự trợ giúp của các Javascript interaction giữa môi trường Native của thiết bị và ReactJS. Do đó, điều này giúp giảm thời gian tải ứng dụng và giúp ứng dụng hoạt động trơn tru mà không bị gián đoạn.

2. ReactJS mang lại hiệu quả cao(ReactJS Brings Advanced Efficiency)

ReactJS có virtual Document Object Model (DOM) riêng để lưu trữ các thành phần ứng dụng. Nó mang lại cho các nhà phát triển tính linh hoạt cao trong việc tổ chức các operation với hiệu suất cao. Sử dụng DOM, ReactJS tính toán trước những thay đổi mà nhà phát triển cần trong các thành phần và từ đó cập nhật các DOM-tree tương ứng. Sự việc cho phép DOM tránh các hoạt động tiêu tốn thời gian và quản lý chúng một cách hiệu quả.

3. Khả năng tái sử dụng của các React component(Reusability of React Components)

Một lợi thế lớn khác tại sao chọn React cho ứng dụng front-end tiếp theo của bạn là nó cho phép tái sử dụng các thành phần mã bất cứ lúc nào. Việc thực hiện tiết kiệm rất nhiều thời gian ở end-developer, và giảm bớt gánh nặng liên quan đến hoạt động của app. Các component ReactJS bị cô lập, do đó, sự thay đổi trong một component sẽ không ảnh hưởng đến bất kỳ component nào khác. Do đó khả năng sử dụng lại trở nên hiệu quả và làm cho việc lập trình chính xác hơn cho các developer.

4. Simple Framework

ReactJS khá đơn giản để sử dụng so với các front-end framework khác. Nó có workflow được xác định rõ và cách tiếp cận component-based giúp cho một số thao tác dễ dàng hơn nhiều. Nó sử dụng cấu trúc JavaScript đơn giản, đó là lý do tại sao khá dễ hiểu.

ReactJS có một cú pháp duy nhất được gọi là JSX, là sự pha trộn giữa JavaScript và HTML. Vì vậy, bất cứ ai có kiến thức lập trình cơ bản, đều có thể dễ dàng hiểu được JSX và bắt đầu xây dựng ứng dụng di động / web đầy đủ chức năng.

5. Luồng dữ liệu một chiều (One-Way Data Flow)

Bạn có thể đạt được sự ổn định code với ReactJS. Nó cho phép các lập trình viên làm việc trực tiếp với các component và cũng sử dụng kỹ thuật liên kết dữ liệu(data-binding) để đảm bảo các thay đổi diễn ra trong các cấu trúc con không ảnh hưởng đến parents. Nhiều view-models of JS có một nhược điểm, trong đó những thay đổi trong các phần tử con gây ra những thay đổi bên trong parents. Nhưng Facebook đã quan tâm đến vấn đề này với React JS. React JS chỉ sử dụng luồng dữ liệu flow-downward. Trong kiểu cấu trúc này, không có cách nào mà các phần tử con có thể ảnh hưởng đến dữ liệu gốc. Do đó, nếu developer muốn thay đổi bất kỳ đối tượng nào, tất cả những gì developer cần làm là sửa đổi trạng thái của đối tượng đó và cập nhật. Theo đó, chỉ các thành phần được phép sẽ nâng cấp.

6. Thư viện JavaScript

Cú pháp JSX là sự pha trộn tổng hợp của HTML và JavaScript được sử dụng cụ thể trong ReactJS. Nó làm cho toàn bộ quá trình viết các component cho trang web dễ dàng hơn rất nhiều, trong khi phần HTML cho phép các nhà phát triển render các chức năng mà không cần nối các strings. ReactJS sử dụng đúng API native và do đó cho phép stack JavaScript hoạt động trên các nền tảng.

7. Hoạt động hoàn hảo với SEO

Nhược điểm chính của nhiều JavaScript framework là chúng không thân thiện với công cụ tìm kiếm. Mặc dù một số đã có những cải tiến đáng kể trong lĩnh vực này, nhưng không phải là rất hữu ích để bổ sung cho các hoạt động SEO.

Đáng ngạc nhiên, ReactJS hoạt động hoàn hảo với SEO. Bạn có thể chạy ReactJS trên máy chủ và DOM ảo sẽ được hiển thị cho trình duyệt như một trang web thông thường, không cần bất kỳ tác vụ nào khác.

8. Dễ học và hiểu cho người mới

So với các front-end framework khác, ReactJS khá dễ học. Những người mới bắt đầu với kiến thức lập trình cơ bản có thể dễ dàng học ReactJS vì cấu trúc mã và cú pháp dễ dàng. Điều duy nhất bạn cần trong khi thành thạo ReactJS là kiến thức sâu về HTML và CSS. Đối với Angular hơi khó học hơn so với ReactJS. Bởi vì nếu bạn không biết về Typescript, bạn không thể thành thạo Angular ngay lập tức. Bạn cần học Typescript trước khi chuyển sang Angular, bản thân nó đòi hỏi một khoảng thời gian lớn.

Các trang web tốt nhất để học ReactJS

Để học React, trước hết, bạn cần có tài liệu khóa học trực tuyến phù hợp để nắm vững các nguyên tắc cốt lõi của framework. Mặc dù sau một vài ví dụ thực tế sau đó sẽ có ích để có được kinh nghiệm làm việc thực tế về React. Dưới đây là một số tài nguyên trực tuyến nơi bạn có thể tìm thấy các khóa học phù hợp để khởi động React một cách nhanh chóng.

Facebook React Docs: [Facebook's React documentation](#) chắc chắn là nguồn tốt nhất để học khung React. Thoạt nhìn có vẻ khó khăn với bạn, nhưng tài liệu này rất toàn diện và có những hiểu biết sâu sắc về chất lượng để tìm hiểu React framework. Bằng cách đọc tài liệu này, bạn có thể hiểu nó một cách dễ dàng.

TutsPlus React Tutorials: TutsPlus có một đồng hướng dẫn React lớn cho cả người mới bắt đầu và nhà phát triển chuyên nghiệp. Nó bao gồm nhiều chủ đề khác nhau - từ cơ bản đến nâng cao - và có hàng tá lời khuyên hữu ích. Đối với người mới, tôi sẽ khuyên bạn nên bắt đầu với TutsPlus React tutorial, vì đây là những hướng dẫn rất dứt khoát và cung cấp trải nghiệm học tập kỹ lưỡng về tất cả các quy trình công việc của ReactJS.

React Training: Trang web React Training là một tài nguyên khác để tìm hiểu React framework rất chi tiết. Các khóa học trên trang web này rất toàn diện về bản chất. Cho đến nay, React Fundamentals là khóa học được đăng ký nhiều nhất của trang web vì nó phác thảo các nguyên tắc cốt lõi của React. Tất cả các khóa học trên trang web đều nhận được cập nhật thường xuyên với các kỹ thuật và thực hành mới, vì vậy, nó đáng để dùng trang web thường xuyên để tìm hiểu các mẹo hữu ích mới.

Pluralsight: Các khóa học theo cấp độ dành cho người mới bắt đầu về Pluralsight giúp bạn có được kiến thức sâu về các nguyên tắc cơ bản của React bao gồm các component states, Props object và JSX. Khi bạn đã hoàn thành khóa học, bạn có thể chuyển sang đào tạo cấp độ intermediate của React bao gồm nhiều chi tiết hơn về framework. Nó bao gồm các chủ đề như Flux, Redux và làm việc với các element phức tạp.

So sánh công nghệ React JS và Angular JS

Technology	Angular JS	React JS
Developer	Google	Facebook
Type of technology	Full-fledged MVC framework written in JavaScript	Open Source JavaScript library
Concept	Brings JavaScript into HTML, works with real DOM Client-side rendering	Brings HTML into JavaScript, works with the virtual DOM server-side rendering
Data-Binding	Two-way data binding	One-way data binding
Dependencies	Manages dependencies automatically	Requires additional tools to manage dependencies
Language	JavaScript + HTML	JavaScript + JSX
Best for	SPAs that update single view at a time	SPAs that update many views at a time
Rendering	Client-side	Server-side
App architecture	MVC	None, combined with Flux

Hiệu suất: Dữ liệu ràng buộc trong Angular phản ánh bất kỳ sự thay đổi dữ liệu nào trong mô hình cũng như trên chế độ xem. Và những dữ liệu thay đổi trong chế độ xem cũng được cập nhật trên mô hình. Khi cần xử lý một lượng dữ liệu lớn, việc này có thể sẽ làm giảm tốc độ ứng dụng. React sử dụng JSX, có một DOM ảo và cung cấp dữ liệu kết nối một chiều. Vì vậy, thay đổi dữ liệu không cần phải được thể hiện trong cả mô hình và chế độ xem.

Khả năng tái sử dụng của code



Trong React, code rất dễ hiểu và sử dụng ít trừu tượng hơn, mặc dù phải mất thời gian để thiết lập các dự án sử dụng React. Angular phức tạp và khó điều khiển vì thư viện của bên thứ ba và các cú pháp.

Khả năng mở rộng: So với React, thiết kế và Giao diện dòng lệnh (CLI) mạnh mẽ giúp cho việc mở rộng các ứng dụng Angular dễ dàng. React đã được kiểm chứng và có thể được mở rộng hơn nữa.

Nhìn chung, ReactJS thích hợp cho những dự án web phát triển lâu dài, phù hợp với việc xây dựng sản phẩm đòi hỏi phải cập nhật, bổ sung tính năng suốt đời.

Thực hành tạo React App và cài đặt thư viện

Đầu tiên cần thiết lập môi trường trên máy tính của bạn.

Cài đặt NodeJS phiên bản mới ≥ 6 (<https://nodejs.org/en/download/>) và NPM (Node package manager)

Cài đặt Visual Studio Code/Sublime Text

Chúng ta có thể tự tạo một React app, nhưng Facebook đã tạo một node module *create-react-app* để tạo ra phiên bản mẫu của một React application. Bên cạnh đó còn cung cấp các task ngoài luồng, dễ dàng thêm tiện ích structure thích hợp cho React app.

Chúng ta sẽ sử dụng **npm** để cài đặt *create-react-app* command line.

```
C:\Users\Your Name>npm install -g create-react-app
```

Thực hiện theo các bước theo đường link: <https://github.com/facebook/create-react-app>

Thực hiện command để tạo một React application tên là react-demo:

```
C:\Users\Your Name>npx create-react-app react-demo
```

Create-react-app sẽ set up mọi thứ cho bạn để chạy React application.

```
nguye@DESKTOP-STE5Q6P MINGW64 /e/VNPT/T10-2019/Project/demo-install-react
$ npx create-react-app react-demo
npx: installed 91 in 28.884s

Creating a new React app in E:\VNPT\T10-2019\Project\demo-install-react\react-de
mo.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

> core-js@2.6.10 postinstall E:\VNPT\T10-2019\Project\demo-install-react\react-d
emo\node_modules\babel-runtime\node_modules\core-js
> node postinstall || echo "ignore"

> core-js@3.2.1 postinstall E:\VNPT\T10-2019\Project\demo-install-react\react-de
mo\node_modules\core-js
> node scripts/postinstall || echo "ignore"

+ react@16.10.2
+ react-dom@16.10.2
+ react-scripts@3.2.0
added 1445 packages from 686 contributors and audited 904901 packages in 291.149
s
found 0 vulnerabilities

Initialized a git repository.

Success! Created react-demo at E:\VNPT\T10-2019\Project\demo-install-react\react
-demo
```

Trong thư mục gốc có một số tệp tin và thư mục sau:

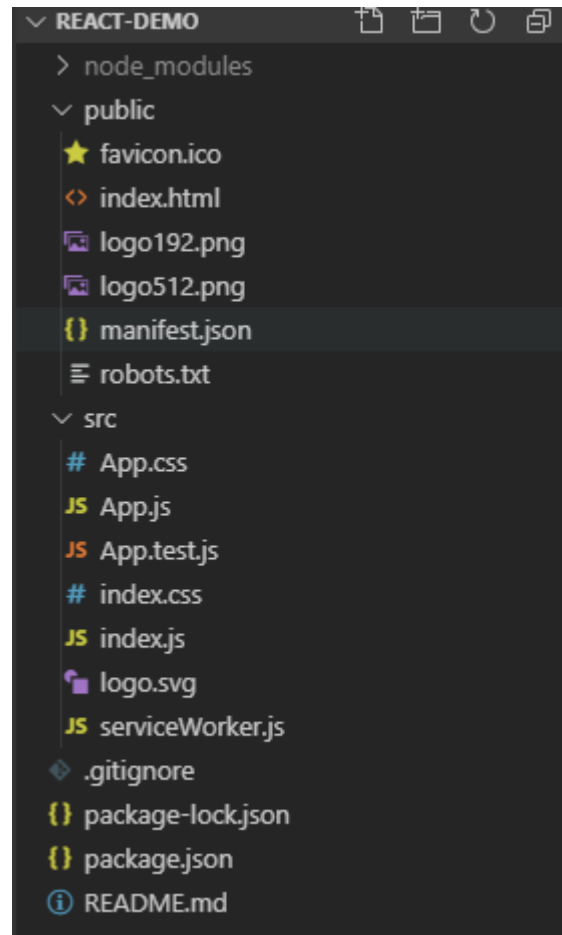
Node_modules: Thư mục này là một phần của hệ thống npm, chứa thư viện React và bất kỳ thư viện bên thứ 3 nào khác.

Public: chứa các tệp tin không được yêu cầu bổ sung bởi webpack. File public/index.html là page template

Src: thư mục này là trái tim của ứng dụng React vì nó chứa JavaScript cần được xử lý bởi webpack. Trong folder này, có component chính là App.js, app.css, app.test.js, index.js và index.css cung cấp entry point cho app. Cuối cùng là file serviceWorker.js take care bộ nhớ đệm và update file cho người dùng cuối.

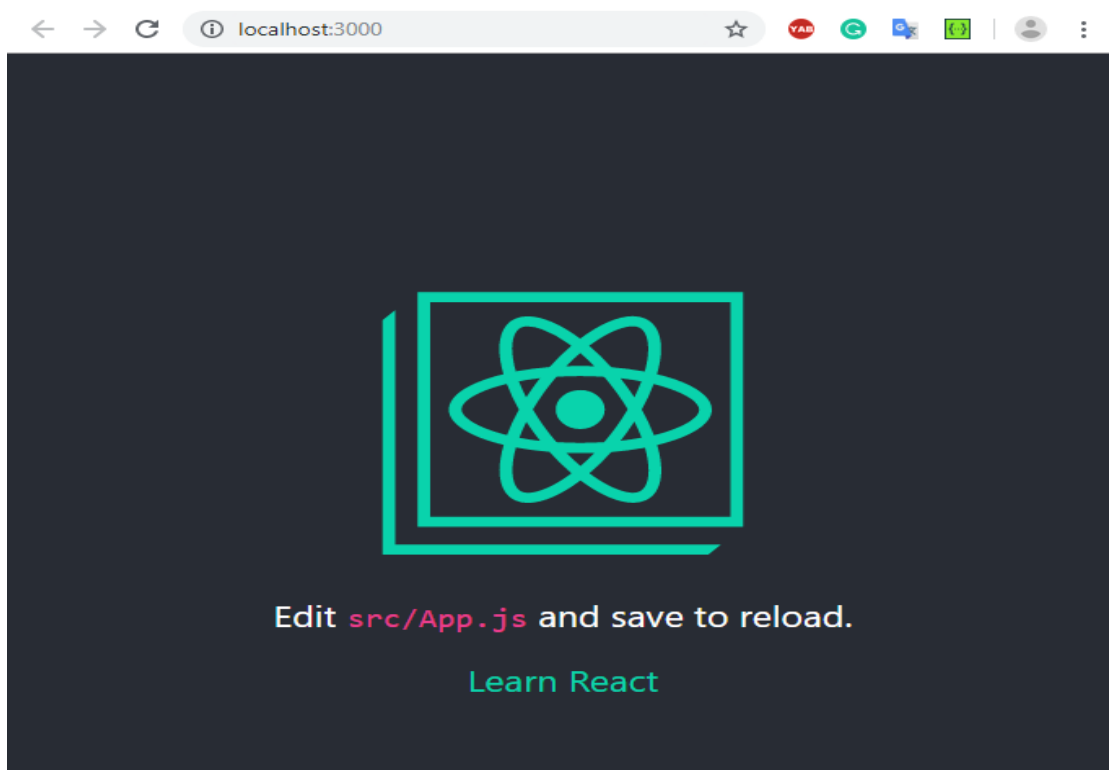
.gitignore : file này là file standard được sử dụng bởi công cụ source control git để xác định các tệp và thư mục nào cần được bỏ qua trong khi commit code.

package.json: file này chứa các dependency và script được yêu cầu cho project.



Tiếp theo, để chạy app ta sử dụng lệnh: **npm start** hoặc **yarn start**, và open <http://localhost:3000/> trên trình duyệt để xem giao diện đầu tiên được tạo React App.

Kết quả:



Chỉnh sửa React application

React render HTML đến trang web bằng cách sử dụng function: ReactDOM.render().

Sử dụng React-Router

Cài React-Router vào project:

```
npm install react-router-dom
```

Ví dụ:

```
import React from "react";
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link
} from "react-router-dom";
import {
  Home, About
} from "../components/?";

export default function App() {
```

```

return (
  <Router>
    <div>
      <nav>
        <ul>
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/about">About</Link>
          </li>
          <li>
            <Link to="/users">Users</Link>
          </li>
        </ul>
      </nav>

      { /* A <Switch> looks through its children <Route>s and
        renders the first one that matches the current URL. */ }
      <Switch>
        <Route path="/about">
          <About />
        </Route>
        <Route path="/users">
          <Users />
        </Route>
        <Route path="/">
          <Home />
        </Route>
      </Switch>
    </div>
  </Router>
);
}

```

Tạo một component Home

```

import React,{Component} from 'react';
export class Home extends Component{

  render(){
    return(
      <div className="mt-5 d-flex justify-content-left">
        <h3>Welcome to this home page.</h3>
      </div>
    )
  }
}

```

```
}
```

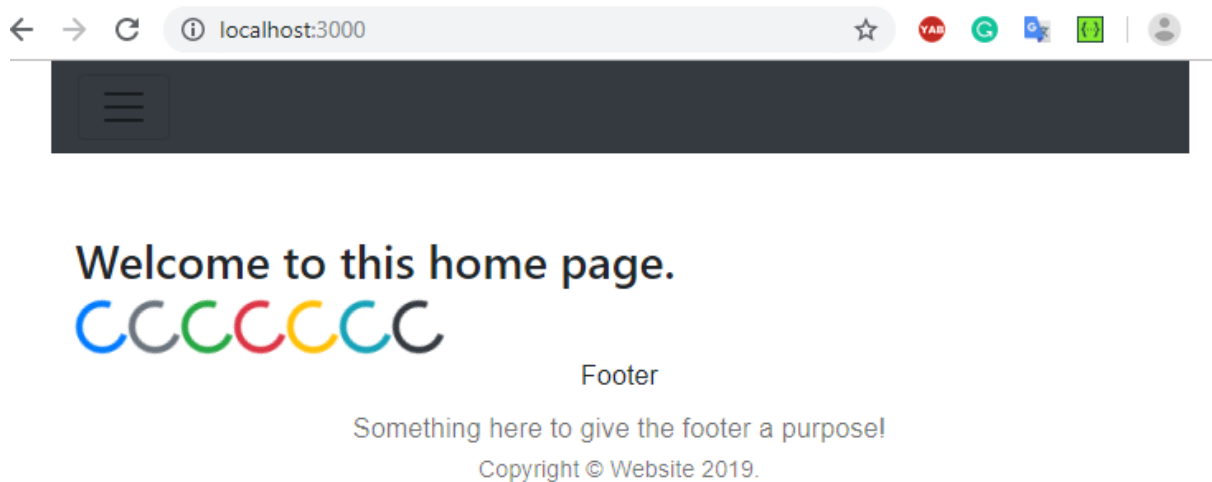
Tiếp theo import vào App.js

```
import { Home } from './components/Home';
```

và sử dụng Router để gọi hiển thị ra view.

```
return (  
  <div className="container">  
    <Navigation />  
    <Home />  
  </div>  
);
```

Và kết quả:



Tổng kết

Trong bài nghiên cứu này, chúng ta thấy ReactJS mở ra thế giới mới về khả năng tương tác cho các ứng dụng front-end. Framework này thân thiện với người dùng, dễ học và mang lại nhiều chức năng để giảm bớt các hoạt động phát triển. Bài viết này trình bày một số hiểu biết sâu sắc về lý do tại sao ReactJS hữu ích cho cả quản trị web chuyên nghiệp và cho những người mới bắt đầu mong muốn nâng cao khả năng học tập của họ. Chúng ta đã tìm hiểu cơ bản về ReactJS các feature, component và react-router cách tạo và sử dụng, đồng thời so sánh ReactJS với AngularJS và ứng dụng demo. Hướng phát triển tiếp sẽ nghiên cứu về React Redux.