

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN GIỮA KÌ MÔN
LẬP TRÌNH WEB VÀ ỨNG DỤNG
SINGLE PAGE APPICATION**

Người hướng dẫn: **GV. NGUYỄN THÁI DUY**

Người thực hiện: **NGUYỄN THỊ THU NGUYỄN – 52000786**

VÕ QUANG HUY – 51800287

NGUYỄN MINH PHÚ – 52100920

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN GIỮA KÌ MÔN
LẬP TRÌNH WEB VÀ ỨNG DỤNG
SINGLE PAGE APPICATION**

Người hướng dẫn: **GV. NGUYỄN THÁI DUY**

Người thực hiện: **NGUYỄN THỊ THU NGUYỄN – 52000786**

VÕ QUANG HUY – 51800287

NGUYỄN MINH PHÚ – 52100920

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

Nhóm em xin gửi lời cảm ơn tới các thầy cô Khoa Công Nghệ Thông Tin, Trường Đại Học Tôn Đức Thắng chân thành nhất. Đặc biệt, nhóm em xin bày tỏ lòng biết ơn sâu sắc với thầy Nguyễn Thái Duy, người đã truyền đạt kiến thức tận tâm trong quá trình giảng dạy và hướng dẫn cách trình bày đề tài nghiên cứu với đề tài “Single Page Application”. Cảm ơn thầy Nguyễn Thái Duy đã tạo điều kiện cho nhóm em nghiên cứu và tìm hiểu thêm về một số tài liệu trong quá trình học tập nhằm nâng cao hiểu biết về phương diện mới của các bài học, và cảm ơn thầy đã cung cấp thêm một số tài liệu hướng dẫn cách trình bày để báo cáo được hoàn chỉnh hơn.

Do kiến thức còn hạn chế nên không tránh khỏi những thiếu sót. Nhóm em mong nhận được sự thông cảm và góp ý kiến của quý thầy cô trong trường.

Cuối cùng nhóm em kính chúc quý thầy cô dồi dào sức khỏe, vạn sự bình an và thành công trong sự nghiệp. Nhóm em xin chân thành cảm ơn!

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của GV Nguyễn Thái Duy. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Võ Quang Huy

Nguyễn Thị Thu Nguyên

Nguyễn Minh Phú

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(ký và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(ký và ghi họ tên)

TÓM TẮT

Đất nước ngày càng phát triển cùng với ngành công nghệ thông tin càng ngày có những bước tiến mạnh mẽ và tin học trở thành chìa khóa dẫn đến thành công cho nhiều lĩnh vực như quản lý kinh tế, kinh doanh, thương mại, ... góp phần tạo nên một bộ mặt mới trong xã hội. Bên cạnh đó, hầu hết hiện nay các lĩnh vực trên đều quản lý dựa vào các website. Các thương hiệu càng lớn thì càng có những yêu cầu cao hơn trong việc kiểm soát trang, tối ưu hóa tốc độ, thì kiến trúc tải trang truyền thống của các ứng dụng web lại chậm gây khó chịu cho người sử dụng.

Chính vì thế, bài toán “Single Page Application” được đặt ra với mục tiêu tạo cho việc quản lý và sử dụng website được nhanh chóng, tiện lợi. Một website dễ dàng trong việc kiểm soát trạng thái người dùng, không cần sử dụng đến cookie, lưu trữ thông tin trong session,... trong việc xử lý các trạng thái giữa hai web request.

Với đề tài “**Single Page Application**”, nhóm chúng em tiến hành đi vào phân tích, khảo sát những vấn đề được đặt ra để có thể tiến hành xây dựng website theo yêu cầu.

Website tập trung phát triển chủ yếu bằng ngôn ngữ lập trình JavaScript để thực hiện các xử lý bên phía server, sử dụng Web API với thư viện React để thiết kế các tính năng trên của trang web. Giúp tạo ra trang web với thiết kế đơn giản, thân thiện với người dùng, dễ sử dụng.

MỤC LỤC

LỜI CẢM ƠN.....	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN.....	iii
TÓM TẮT.....	iv
MỤC LỤC.....	1
DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT.....	2
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ.....	3
CHƯƠNG 1 - SINGLE PAGE APPLICATION.....	4
1.1 Giới thiệu.....	4
1.2 Lợi ích của SPA.....	4
1.3 Kiến trúc của SPA.....	5
1.4 Multi - Page Application.....	6
1.5 So sánh SPA và MPA.....	7
1.6 Lựa chọn SPA hay MPA.....	9
CHƯƠNG 2 - REACT.....	10
2.1 Giới thiệu về React.....	10
2.2 Tại sao sử dụng React.....	10
2.3 JSX.....	11
2.4 Virtual DOM.....	12
CHƯƠNG 3 - WEB API.....	13
3.1 Giới thiệu.....	13
3.2 ASP.NET Web API.....	13
3.3 Tại sao nên chọn Web API.....	14
3.4 Làm thế nào để sử dụng Web API.....	15
TÀI LIỆU THAM KHẢO.....	17

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

CÁC CHỮ VIẾT TẮT

SPA	Single Page Application
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
DOM	Document Object Model
MPA	Multi – Page Application
SEO	Search Engine Optimization
GUI	Graphical User Interface
JSX	JavaScript extension
XML	Xtensible Markup Language

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

Bảng 1: Bảng so sánh SPA và MPA.....	8
Hình 1. 1: Mô hình hoạt động SPA.....	5
Hình 2. 1: JSX code style.....	11
Hình 2. 2: Sử dụng Virtual DOM.....	12

CHƯƠNG 1 - SINGLE PAGE APPLICATION

1.1 Giới thiệu

Single Page Application (SPA) là một ứng dụng web hoặc trang web tương tác với người dùng bằng cách tự động viết lại trang web hiện tại bằng dữ liệu mới từ máy chủ web, thay vì phương pháp mặc định của trình duyệt web là tải lại toàn bộ trang mới.

Mục đích cuối cùng cơ bản của việc sử dụng SPA là có chuyển đổi nhanh hơn, mượt mà hơn, có thể mang lại cho website hoặc webpage một trải nghiệm giống như ứng dụng tự nhiên hơn. Ở trong SPA, việc làm mới trang không bao giờ xảy ra. Thay vào đó, tất cả code HTML, CSS, JavaScript cần thiết sẽ được trình duyệt truy xuất chỉ bằng một lần tải trang, hoặc các tài nguyên thích hợp được tải động và thêm vào trang khi cần, thường là để phản hồi lại hành động của người dùng.

Một số trang web lớn có ứng dụng SPA: Google, Facebook, Twitter, Shoppe, chotot, zingmp3,...

1.2 Lợi ích của SPA

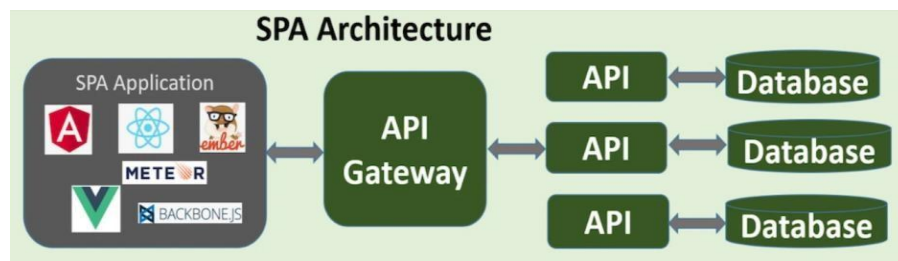
Sử dụng Single Page application có nhiều lợi ích và tăng trải nghiệm cho người dùng như:

- Dễ dàng trong việc kiểm soát trạng thái người dùng, không cần sử dụng đến cookie, lưu trữ thông tin trong session,... trong việc xử lý các trạng thái giữa hai web request.
- Giảm lượng thông tin trao đổi giữa client và server, ví dụ các phần khung trang được giữ nguyên và chỉ tải các nội dung cần thay đổi. Chính vì vậy, SPA có thể sử dụng rất tốt với các đường truyền tốc độ thấp.
- Gần như không có độ trễ khi chuyển trang.

- Giảm số lượng truy vấn đến máy chủ, tưởng tượng trong Gmail có hàng nghìn mail, mỗi lần nó sẽ tải 50 email về vì vậy việc mở các email này sẽ rất nhanh do được tải sẵn.
- Đem đến một trải nghiệm người dùng tuyệt vời hơn, các xử lý chạy ngầm giúp các hoạt động thực hiện nhanh chóng.

1.3 Kiến trúc của SPA

Có 3 tùy chọn liên quan đến kiến trúc của SPA:



Hình 1. 1: Mô hình hoạt động SPA

- Kết xuất phía máy khách:
 - + Trình duyệt web nhận yêu cầu từ máy khách và gửi nó đến máy chủ định dạng HTML.
 - + Máy chủ nhận yêu cầu của máy khách và phản hồi bằng tệp HTML bao gồm văn bản, liên kết cho hình ảnh, CSS và JavaScript.
 - + Khi máy chủ yêu cầu tệp HTML, người dùng sẽ thấy tải hình ảnh hoặc trang trống khi thực thi JavaScript.
 - + Trong khi đó, SPA truy xuất dữ liệu, tạo dạng xem và đặt nó vào Document Object Model (DOM).
 - + Cuối cùng, SPA sẵn sàng để sử dụng.
- ⇒ Tuy nhiên, việc hiển thị dữ liệu phía máy khách có thể gây ra gánh nặng cho trình duyệt web vì nó cần sử dụng nhiều tài nguyên của thiết bị di động hơn. Đây là tùy chọn chậm nhất.
- Kết xuất phía máy chủ:

- + Công cụ tìm kiếm hoặc trình duyệt web yêu cầu máy chủ yêu cầu tệp HTML.
- + Máy chủ truy xuất thông tin cần thiết, kết xuất ứng dụng web và tạo tệp HTML một cách nhanh chóng.
- + Ở giai đoạn này người dùng có thể xem được thông tin được yêu cầu.
- + Trong kết xuất phía máy chủ, các SPA liên kết các sự kiện với nhau và tạo DOM ảo cho nội dung được yêu cầu.
- + Cuối cùng, SPA sẵn sàng để sử dụng.
- ⇒ Đây là tùy chọn tốt nhất để tối ưu hóa tốc độ của các ứng dụng web và giảm gánh nặng cho trình duyệt web.
- Trình tạo trang tĩnh:
 - + Máy khách đưa ra yêu cầu tới máy chủ với tệp HTML.
 - + Máy chủ nhanh chóng gửi lại tệp HTML được tạo sẵn cho máy khách.
 - + Người dùng có thể xem trang.
 - ⇒ Ứng dụng web của bạn truy xuất dữ liệu, tạo chế độ xem dữ liệu và đặt nó vào DOM.
 - ⇒ Tùy chọn này nhanh hơn 2 tùy chọn còn lại. Tuy nhiên bạn chỉ có thể chọn tùy ý cho một trang web tĩnh vì nó chỉ hỗ trợ cho trang web tĩnh.

1.4 Multi - Page Application

Multi – Page Application (MPA): là một ứng dụng web bao gồm một số lượng lớn các trang được làm mới hoàn toàn mỗi khi dữ liệu thay đổi trên chúng. Bất kỳ thay đổi dữ liệu hoặc chuyển dữ liệu đến máy chủ đều dẫn đến một trang mới được hiển thị trong trình duyệt.

Đây là một cách tiếp cận cổ điển để phát triển ứng dụng web, yêu cầu điều hướng đa cấp và tập trung nhiều hơn vào tốc độ và hiệu suất. Một số ví dụ về các ứng dụng web MPA như: các blog, diễn đàn, thư mục, trang web xuất bản trực tuyến, ...

1.5 So sánh SPA và MPA

	SPA	MPA
Tốc độ	Tải nhanh hơn vì nó tải phần lớn tài nguyên ứng dụng chỉ một lần. Trang sẽ không tải lại hoàn toàn khi người dùng yêu cầu một phần dữ liệu mới.	Chậm hơn vì trình duyệt phải tải lại toàn bộ trang từ đầu khi người dùng muốn truy cập dữ liệu mới hoặc di chuyển đến một phần khác của trang web.
Kết nối	Được tách rời mạnh mẽ, nghĩa là phần đầu và phần cuối tách biệt. Các ứng dụng SPA sử dụng API do các nhà phát triển phía máy chủ dùng để đọc và hiển thị dữ liệu.	Front – end và back-end phụ thuộc lẫn nhau nhiều hơn. Tất cả code đều thường được đặt trong một dự án.
SEO	Đối với SEO thì đây là một yếu điểm. Vì hầu hết SPA đều chạy trên JavaScript, thứ mà hầu hết các công cụ tìm kiếm không hỗ trợ. Trình thu thập thông tin của các công cụ tìm kiếm tải xuống các tệp HTML của trang, điều này làm cho các trang web HTML tĩnh dễ xếp hạng hơn.	Cho phép định vị trang web tốt hơn vì mỗi trang có thể được tối ưu hóa cho một từ khóa khác. Ngoài ra, các thẻ meta có thể được bao gồm trên mọi trang – điều này tác động tích cực đến thứ hạng của Google.
Trải nghiệm người dùng	Thân thiện với thiết bị di động hơn. Điều đáng nói là, rất nhiều lưu lượng truy cập đến từ thiết bị di động. Ngay cả Google cũng bắt đầu ưu tiên trải nghiệm di động hơn máy tính.	Cho phép kiến trúc thông tin tốt hơn. Bạn có thể tạo bao nhiêu trang tùy thích và bạn có thể có bao nhiêu thông tin trên một trang mà không có bất kỳ giới hạn nào. Điều hướng rõ ràng

		hơn, vì vậy người dùng dễ dàng tìm đường trên trang web, điều này tác động tích cực đến trải nghiệm của họ.
Độ an toàn	Dễ bị tin tặc tấn công hơn vì chạy trên JavaScript, vốn không thực hiện biên dịch mã nên dễ bị phần mềm độc hại tấn công hơn. Để đủ an toàn cho trang web thì cần bảo mật các điểm cuối dữ liệu nhanh hơn nhưng không nhất thiết phải an toàn hơn.	Phải bảo mật trên mọi trang web.
Quá trình phát triển	Có thể tái sử dụng là một lợi thế lớn nhất. Chúng ta có thể áp dụng một đoạn code đã sử dụng trong ứng dụng web cho ứng dụng di động gốc của mình, vì các ứng dụng và trang web thường được sử dụng trên thiết bị di động.	Có sự phân chia rõ ràng giữa front-end và back-end, cả hai phần có thể được phát triển đồng thời. Bên cạnh đó MPA mất nhiều thời gian hơn để phát triển vì trong hầu hết các trường hợp, phía máy chủ phải được mã hóa ngay từ đầu.

Bảng 1: Bảng so sánh SPA và MPA

Không có kiến trúc nào là hoàn hảo – cả hai đều có ưu và nhược điểm riêng. SPA chiến thắng về tốc độ và khả năng sử dụng lại code, có thể được áp dụng để phát triển ứng dụng dành cho thiết bị di động nhưng lại có thiếu sót trong việc tối ưu hóa SEO. Sử dụng MPA sẽ giúp trang web xếp hạng cao hơn trong Google và có khả năng mở rộng hơn nhưng lại chậm hơn nhiều so với SPA.

1.6 Lựa chọn SPA hay MPA

Việc lựa chọn tùy chọn nào phải phù hợp với mục tiêu kinh doanh cùng với các yêu cầu của mình. Đối với SPA: được sử dụng tốt hơn trong các ứng dụng mạng xã hội, nền tảng SaaS – bất cứ nơi nào mà thứ hạng SEO không phải là yếu tố quyết định. Đối với MPA: được sử dụng tốt nhất trong các ứng dụng thương mại điện tử, danh mục doanh nghiệp và thị trường. Nếu là một công ty cung cấp nhiều loại sản phẩm thì MPA là lựa chọn tốt nhất.

CHƯƠNG 2 - REACT

2.1 Giới thiệu về React

React là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Nó cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web. Được duy trì bởi Meta và cộng đồng các nhà phát triển và công ty riêng lẻ. React có thể được sử dụng làm cơ sở trong việc phát triển các ứng dụng SPA, mobile hoặc máy chủ kết xuất với frameworks. Tuy nhiên, React chỉ quan tâm đến việc quản lý trạng thái và hiển thị trạng thái đó cho DOM. Vì vậy, việc tạo các ứng dụng React thường yêu cầu sử dụng các thư viện bổ sung để định tuyến, cũng như một số chức năng phía máy khách.

Một số công ty lớn trên thế giới sử dụng React như: Netflix, Airbnb, American Express, Facebook, Instagram,...

2.2 Tại sao sử dụng React

Dễ sử dụng: React là một thư viện GUI nguồn mở JavaScript tập trung vào một điều cụ thể, hoàn thành nhiệm vụ UI hiệu quả. Dễ dàng hiểu được những điều cơ bản về React và có thể bắt đầu phát triển ứng dụng dựa trên web bằng cách sử dụng React chỉ trong vài ngày.

Hỗ trợ Reusable Component trong Java: Cho phép sử dụng lại components đã được phát triển thành các ứng dụng khác có cùng chức năng.

Viết component dễ dàng hơn: Dễ viết hơn vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho JavaScript cho phép kết hợp HTML với JavaScript. JSX làm rõ toàn bộ quá trình viết cấu trúc trang web. Ngoài ra, phần mở rộng cũng giúp render nhiều lựa chọn dễ dàng.

Hiệu suất tốt hơn với Virtual DOM:

- React cập nhật hiệu quả quá trình DOM nên có thể tránh được vấn đề thất vọng trong các dự án ứng dụng trên web.

- Cho phép xây dựng các virtual DOM và host chúng trong bộ nhớ. Nhờ vậy, mỗi khi có sự thay đổi trong DOM thực tế, thì virtual DOM sẽ thay đổi ngay lập tức.
- Hệ thống này sẽ ngăn DOM thực tế để buộc các bản cập nhật được liên tục. Do đó, tốc độ của chúng sẽ không bị gián đoạn.

Thân thiện với SEO: React cho phép bạn tạo giao diện người dùng có thể truy cập trên các công cụ tìm kiếm khác nhau. React có thể tăng tốc quá trình của ứng dụng nên có thể cải thiện kết quả SEO.

2.3 JSX

JSX là một React Extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt mà bạn đang làm việc.

```
function App() {
  return (
    <div className="App">
      <Fragment>
        <div className='header'>
          <Navb />
          <Outlet />
        </div>
      </Fragment>
    </div>
  );
}

export default App;
```

Hình 2. 1: JSX code style

JSX = JavaScript + XML. Nó transform cú pháp gần như XML về thành JavaScript. Người lập trình có thể code ReactJS bằng cú pháp của XML thay vì sử dụng JavaScript. JSX là một phần quan trọng trong ReactJS. JSX giống như một syntax extension giúp code trở nên mượt mà và dễ đọc hơn.

2.4 Virtual DOM

Virtual DOM là một định dạng dữ liệu JavaScript nhẹ được dùng để thể hiện nội dung của DOM tại một thời điểm nhất định nào đó. Nó có tất cả các thuộc tính giống như DOM nhưng không có khả năng tương tác lên màn hình như DOM.

Ở DOM có thẻ div và các thẻ p ở trong, ReactJS sử dụng Virtual DOM bằng cách tạo ra các object React.div và React.p và khi tương tác, ta sẽ tương tác qua các object đó một cách nhanh chóng mà không phải đụng tới DOM hay DOM API của nó.

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<App />} />
        <Route index element={<Home />} />
        <Route path='pt' element={<PT />} />
        <Route path='contact' element={<Contact />} />
      </Routes>
    </BrowserRouter>
  </React.StrictMode>
);
```

Hình 2. 2: Sử dụng Virtual DOM

CHƯƠNG 3 - WEB API

3.1 Giới thiệu

API – Application Programming Interface: là một giao diện lập trình ứng dụng cho máy chủ web hoặc trình duyệt web. API là một giao diện có một tập hợp các chức năng cho phép lập trình viên truy cập các tính năng hoặc dữ liệu cụ thể của một ứng dụng, hệ điều hành, hoặc các dịch vụ khác.

Web API là một API qua web có thể truy cập bằng giao thức HTTP. Đó là một khái niệm và không phải một công nghệ. Chúng ta có thể xây dựng web API bằng các công nghệ khác nhau như Java, .NET,... API Web được sử dụng trong máy chủ web hoặc trình duyệt web. Về cơ bản Web API là một khái niệm phát triển web. Nó được giới hạn ở phía máy khách của ứng dụng web và nó cũng không bao gồm máy chủ web hoặc chi tiết trình duyệt web. Nếu một ứng dụng được sử dụng trên một hệ thống phân tán và để cung cấp dịch vụ trên các thiết bị khác nhau như máy tính xách tay, điện thoại di động, v.v. thì các dịch vụ web API sẽ được sử dụng. Web API là hình thức nâng cao của ứng dụng web.

3.2 ASP.NET Web API

ASP.NET – Active Server Pages.NET. Nó chủ yếu được sử dụng để tạo các trang web và công nghệ web. Nó được coi là một công cụ rất quan trọng đối với các nhà phát triển để xây dựng các trang web động bằng cách sử dụng các ngôn ngữ như C # và Visual Basic.

ASP.NET Web API là một khuôn khổ có thể mở rộng giúp xây dựng các dịch vụ dựa trên HTTP giúp dễ dàng tiếp cận nhiều loại ứng dụng khách bao gồm trình duyệt, điện thoại di động, máy tính bảng, v.v. Nó hoạt động ít nhiều giống như ứng dụng web MVC ASP.NET ngoại trừ việc nó gửi dữ liệu dưới dạng phản hồi thay vì chế độ xem html. Nó giống như một webservice hoặc dịch vụ WCF nhưng ngoại lệ là nó chỉ hỗ trợ giao thức HTTP.

Đặc điểm ASP.NET Web API:

- ASP.NET Web API là một nền tảng lý tưởng để xây dựng các dịch vụ RESTful.
- ASP.NET Web API được xây dựng dựa trên ASP.NET và hỗ trợ quy trình yêu cầu/phản hồi ASP.NET.
- ASP.NET Web API ánh xạ động từ HTTP đến tên phương thức.
- ASP.NET Web API hỗ trợ các định dạng dữ liệu phản hồi khác nhau. Hỗ trợ tích hợp cho định dạng JSON, XML, BSON.
- ASP.NET Web API có thể được lưu trữ trong IIS, Self-hosted hoặc máy chủ web khác hỗ trợ .NET 4.0+.
- ASP.NET Web API Framework bao gồm HttpClient mới để giao tiếp với máy chủ Web API. HttpClient có thể được sử dụng trong ASP.MVC phía máy chủ, ứng dụng Windows Form, ứng dụng Bảng điều khiển hoặc các ứng dụng khác.

3.3 Tại sao nên chọn Web API

Web API rất hữu ích trong việc triển khai các dịch vụ web RESTFUL sử dụng .NET framework. Bên cạnh đó còn có một số lý do để chọn Web API:

- Các dịch vụ Web API thích hợp hơn các dịch vụ khác để sử dụng với một ứng dụng gốc không hỗ trợ SOAP nhưng yêu cầu các dịch vụ web.
- Để tạo các dịch vụ định hướng tài nguyên, các dịch vụ web API là lựa chọn tốt nhất. Bằng cách sử dụng HTTP hoặc dịch vụ nghỉ ngơi, các dịch vụ này được thiết lập.
- Nếu bạn muốn hiệu suất tốt và phát triển nhanh chóng các dịch vụ, các dịch vụ web API rất hữu ích.
- Để phát triển các dịch vụ web có trọng lượng nhẹ và có thể duy trì, các dịch vụ web API thực sự hữu ích để phát triển dịch vụ đó. Nó hỗ trợ bất kỳ mẫu văn bản nào như JSON, XML, v.v.
- Các thiết bị có băng thông hẹp hoặc có giới hạn về băng thông, thì các dịch vụ Web API là tốt nhất cho các thiết bị đó.

3.4 Làm thế nào để sử dụng Web API

Web API nhận các yêu cầu từ các loại thiết bị khác nhau như thiết bị di động, máy tính xách tay, v.v., sau đó gửi các yêu cầu đó đến máy chủ web để xử lý các yêu cầu đó và trả về đầu ra mong muốn cho máy khách. Web API là một tương tác giữa Hệ thống với Hệ thống, trong đó dữ liệu hoặc thông tin từ một hệ thống có thể được truy cập bởi một hệ thống khác, sau khi hoàn thành việc thực thi, dữ liệu kết quả hoặc chúng ta có thể nói là đầu ra được hiển thị cho người xem.

API cung cấp dữ liệu cho các lập trình viên của mình được cung cấp cho người dùng bên ngoài. Khi các lập trình viên quyết định cung cấp một số dữ liệu của họ cho công chúng, họ "phơi bày các điểm cuối", có nghĩa là họ xuất bản một phần ngôn ngữ mà họ đã sử dụng để xây dựng chương trình của mình. Các lập trình viên khác sau đó có thể trích xuất dữ liệu từ ứng dụng bằng cách xây dựng URL hoặc sử dụng máy khách HTTP để yêu cầu dữ liệu từ các điểm cuối đó.

Phía máy chủ: API web phía máy chủ là một giao diện lập trình. Nó bao gồm một hoặc nhiều điểm cuối được tiếp xúc công khai. Nó xác định một hệ thống tin nhắn yêu cầu-phản hồi. Mashup là một ứng dụng web là một API phía máy chủ kết hợp một số API phía máy chủ. Webhook là một API phía máy chủ lấy đầu vào làm mã định danh tài nguyên thống nhất.

Phía khách hàng: Web API phía máy khách nhắm mục tiêu các ràng buộc JavaScript được tiêu chuẩn hóa. Google đã tạo kiến trúc máy khách gốc của họ được thiết kế để thay thế các plugin gốc bằng các ứng dụng và tiện ích mở rộng hộp cát gốc an toàn.

Các bước sử dụng API Web:

- Hầu hết các API đều yêu cầu khóa API. Khi bạn tìm thấy API mà bạn muốn sử dụng, hãy xem trong tài liệu để biết các yêu cầu về quyền truy cập. Hầu hết các API sẽ yêu cầu bạn hoàn tất quy trình xác minh danh tính, chẳng hạn như

đăng nhập bằng Tài khoản Google của bạn. Bạn sẽ nhận được một chuỗi chữ cái và số duy nhất để sử dụng khi truy cập API.

- Cách dễ nhất để bắt đầu sử dụng API là tìm một máy khách HTTP trực tuyến, như REST-Client, Postman hoặc Paw. Các công cụ được tạo sẵn này giúp bạn cấu trúc các yêu cầu của mình để truy cập các API hiện có bằng khóa API mà bạn nhận được. Bạn vẫn sẽ cần biết một số cú pháp từ tài liệu, nhưng có rất ít kiến thức về mã hóa cần thiết.
- Cách tốt nhất tiếp theo để lấy dữ liệu từ API là xây dựng URL từ tài liệu API hiện có.

TÀI LIỆU THAM KHẢO

1. [Single-page application - Wikipedia](#)
2. [A Comparison of Single-Page and Multi-Page Applications - DZone](#)
3. [Single-Page Application vs Multi-Page Application: Pros, Cons, and Which is Better? \(lvivivity.com\)](#)
4. [Giới thiệu cách thức hoạt động của ReactJS \(tino.org\)](#)
5. [JSX trong React \(viblo.asia\)](#)
6. [Hiểu sao về Virtual DOM trong ReactJs? \(viblo.asia\)](#)
7. [Web API - Wikipedia](#)