

[HOME](#)[CONCEPT](#)[HISTORY](#)[CV](#)

Overview

1990s

1995

1997

1999

2000

2004

2005

2006 – 2010

2013

2016

2017

2020

Current

Reference

Overview

At the beginning of internet, websites were static. However, there are many tasks can only be done with apps and websites that are dynamic and responsive to different conditions. Global increase in hardware with open architecture formats which enabled the design of open architecture software and open APIs leveraging REpresentational State Transfer (REST). The optional constraint on REST called Code On Demand (COD) enables client functionality to be extended by downloading and executing code in the form of applets or scripts. Back to mid-1990s, there were no certain modelling of server-client communication until Roy Fielding described COD in 2000. Companies around the

global quickly adopt REST. The high demand of COD and vastly implementation of REST lead to development and collaboration which enable new modules, frameworks, technology and services such as (Cloud, Git, DOM, AJAX, Flask ...) . This opening enormous opportunities and challenges. Below is the history of REST (mainly focus on COD) which are categorized by timeline.

Early to mid-1990s

During that early days, website were static, communication between server and client were limited and there was no way of making websites dynamic.

1995: Java-applets were introduced

Some scripting languages were introduced to make website dynamic and responsive to different condition.

In September 1995, a Netscape programmer named Brendan Eich developed a new scripting language. It was originally named Mocha, but quickly became known as LiveScript and, later, JavaScript. JavaScript is a scripting language that enables developer to create dynamically updating content, control multimedia, animate images, and almost everything requires dynamic behaviour implementation. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, it is one of the three essential technologies of World Wide Web content production.

JavaScript and Java are similar in some ways but fundamentally different because JavaScript inherits dynamic typing and object-oriented, while Java's static and class-based.

1997: ECMAScript Is Born

ECMAScript is a JavaScript standard meant to ensure the interoperability of web pages across different web browsers. It is standardized by [ECMA](#) according to the document ECMA-262.

1999: API

API allow developers to release software as a service, and is a key building block for modern web applications.

The functionality built on top of the client-side JavaScript language called Application Programming Interfaces (APIs) provide extra superpowers to use in your JavaScript code.

APIs are ready-made sets of code building blocks that allow a developer to implement programs that would otherwise be hard or impossible to implement. API was free. At that time, most developers use a protocol called Simple Object Access Protocol (SOAP) to integrate APIs.

2000: The beginning of REST

REST was defined in 2000 by [Roy Fielding](#). REST is an architecture for the web that describe interactions with web based resources. HTTP is stateless, so there is no memory between transactions. REST uses the current page as a proxy for state, and operations to move from one to the other.

On Roy's doctoral dissertation, he model down a core set of principles, properties, and 6 constraints that are now called REST. The 6 constraints are client-server, layered system, cache, code on demand, stateless, uniform interface. Most companies implement REST because of it characteristic including usable with any language, HTTP methods explicitly, lightweight, simple data structures, exposed URLs

[Code on Demand](#) COD is the optional constrain of REST. The COD principle states that the server can provide executable code in responses to a client. This is common practice with web browsers, where JavaScript is provided to be run by the client. However this isn't commonly included in REST APIs since there is no standard for executable code, so for example, iOS won't execute JavaScript.

In comparison between REST and SOAP, REST is a better choice for simple, CRUD-oriented services, because of the way REST repurposes HTTP methods (GET, POST, PUT, and DELETE). It is also popular because it's lightweight and has a smaller learning curve. SOAP, on the other hand, has stricter implementation, higher standards for security, addressing, etc.

2004: The rise of Social Network

Flickr launched its own REST API, just in time for the rise of social networking and blogging. Enabling option to post and share. Facebook and later Twitter hastily joined. The request for public APIs escalated, the Web was changing. Easily accessible REST APIs now enable everyone to add a feature to a website in a short time. To be distinguished, REST is the set of constraints. RESTful refers to an API adhering to those constraints. Facebook, Twitter, and Google expose their functionality in the form of Restful web services. This allows any client application to call these web services via REST.

2005: AJAX

A paper released by Jesse James Garrett introduced Ajax which stands for Asynchronous JavaScript and XML, is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications. Ajax vastly improved user experience by allowing web pages to feel more like native desktop apps. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page because it request data from a server without refreshing the page or blocking the application. This really pushed JavaScript into the spotlight as a professional programming language with large and popular JavaScript frameworks and libraries, such as Dojo and MooTools and jQuery. [jQuery](#) is one of the most popular libraries in the world because it it simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.

2006 – 2010: Cloud Computing – AWS, Google Cloud, Microsoft Azure released

Amazon Web Services (AWS) launch the cloud with services that can create RESTful API with easily-accessible REST APIs enabled sites to add a functionality to the site in no time. Since then, REST APIs

have become the backbone of the Internet and creators of huge business opportunities due to their ability to extend the scope of a brand beyond the audience of a website.

The CommonJS project set out to define and promote JavaScript development outside the browser by using modules to package useful code and functionality. Node.js is a more advance server side version of JavaScript. In REST API-based applications—JavaScript is used both in the frontend and backend of sites. Thus, a server can easily communicate with the frontend via REST APIs using Node.js.

Node.js allows JavaScript to be run without a browser (browserless), and is commonly used to run web servers.

Before Node.js, developers would have to use different languages for the backend and frontend of their application. For example, PHP, Java, ASP.Net would run on the server, and JavaScript would run in the client browser. Now with Node.js, developers can use JavaScript on the server as well as the client

2013: the end of Java Applets

Major web browsers began to phase out support for the underlying technology applets used to run.

2016: Angular

Angular is a JavaScript framework which is made for building large, single-page web applications.

2017: WebAssembly is released

WebAssembly is a new type of code that can be run in modern web browsers with huge implication and high performance

2013: Flash support ended.

Adobe blocked Flash content from running in Flash Player beginning January 12, 2021. There was criticism on Flash, one of the well-known case was published on [“Thoughts on Flash”](#) is an open letter published by Steve Jobs. The letter criticizes Flash platform and outlines reasons why the technology would not be allowed on Apple's iOS hardware products. In retrospect many publications came to agree with Jobs.

Current

RESTful refers to an API adhering to RESTconstraints. RESTful APIs make storing and sharing data of any kind of possible, integrating functionality into software. It has led APIs to the sky's the limit.

Reference

- <https://www.springboard.com/blog/data-science/history-of-javascript/>
- http://restfulwebapis.org/RESTful_Web_Services.pdf
- <https://blog.readme.com/the-history-of-rest-apis/>
- <https://organicdonut.com/2013/08/technical-understanding-javascript-node-js-and-their-libraries/>
- <https://mobapi.com/history-of-rest-apis/>
- <https://kinsta.com/knowledgebase/what-is-node-js/>

- https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf
- <https://www.w3schools.com/>
- <https://stackoverflow.com/>
- <https://github.com/tomsmoker>
- CITS5505 Lectures, Tim French

Hudi Baldacchino | 23219385@student.uwa.edu.au | Find me on   