Giới thiệu bản thân:

Mạnh:

Yếu:

Giới thiệu những dự án đã làm:

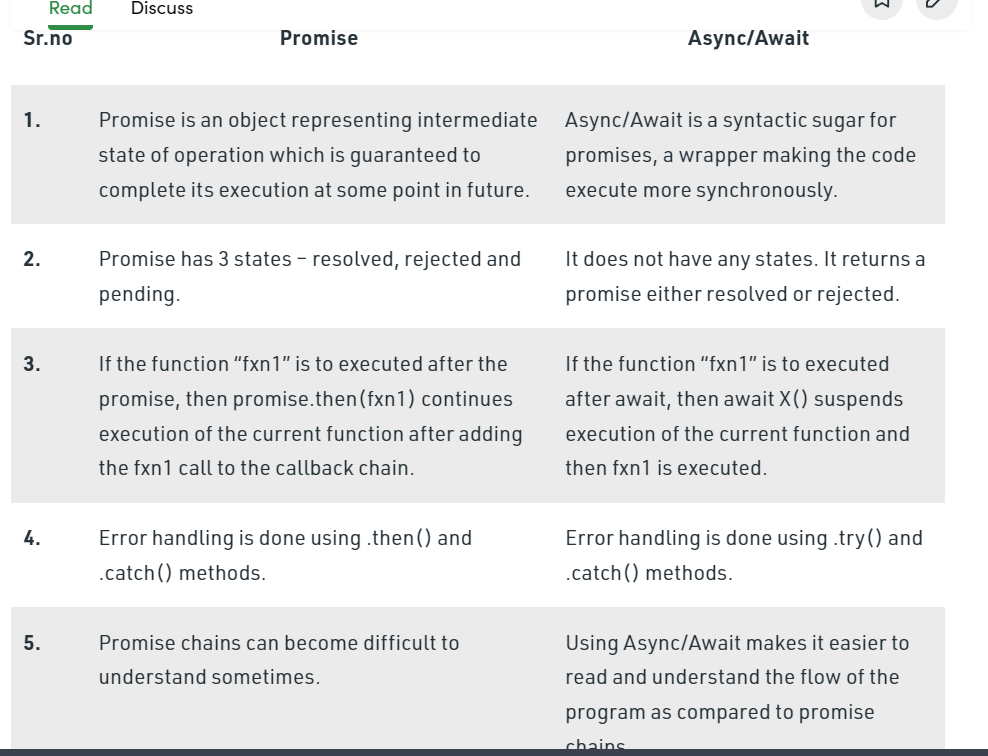
Slice: Lấy từ a->b không lấy b, (a > b)

Substring tương tự như slice nhưng có thể a<b. không nhận -a,-b;

Kiểu dữ liệu Js: Number, String,Obj,boolean,null,undefine,NaN

ES6: arrow func,Default parameter,Template Literals,Spread,for…of,classes.

Promise vs Async await:



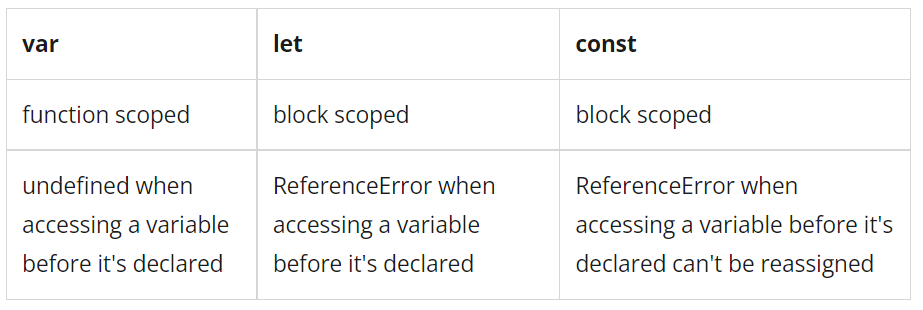
Cookies localStorage

|  |  |  |
| --- | --- | --- |
|  | Cookies | localStorage |
| Data Size | 4096 bytes ->authentication | 5MB |
| Data Expiry | You can set expire | data in the local storage does not expire and will be available any time the user opens your website |

Cookie: text, lưu ở client,

Session: Lưu trong ram của serve,

Function vs Arrow function:



Arrow function: <https://www.programiz.com/javascript/arrow-function>

Function syntax,contructor,return,

let x = function () {

console.log(arguments);

}

x(4,6,7); // Arguments [4, 6, 7]

let x = () => {

console.log(arguments);

}

x(4,6,7);

// ReferenceError: Can't find variable: arguments

### Example 3: Arrow Function as an Expression

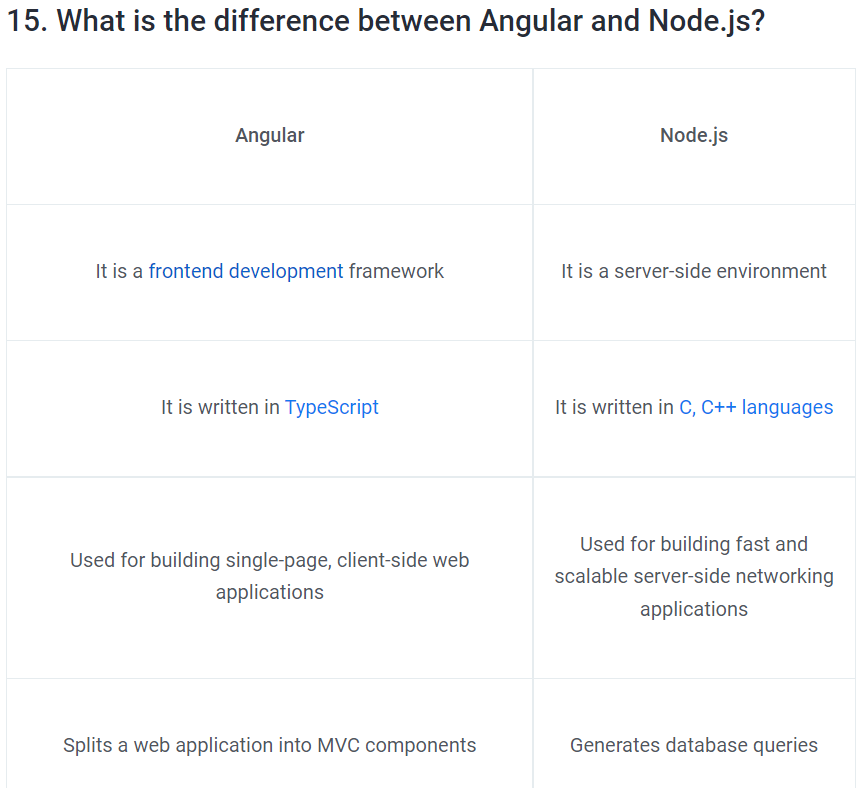
let age = 5;

let welcome = (age < 18) ?

() => console.log('Baby') :

() => console.log('Adult');

welcome(); // Baby



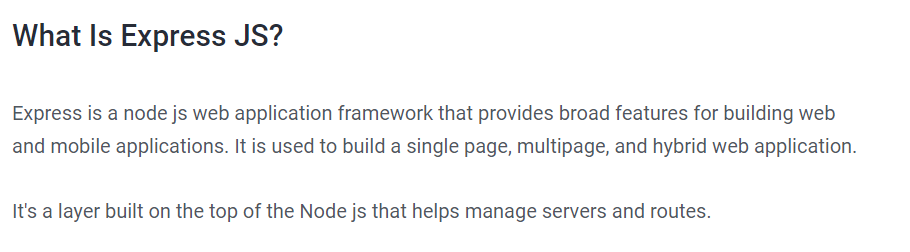
V8 engine, do Google phát triển, **là mã nguồn mở và được viết bằng C ++.** Google Chrome sử dụng công cụ này. Không giống như các động cơ khác, V8 cũng được sử dụng cho thời gian chạy Node.js phổ biến. Ban đầu V8 nhằm mục đích cải thiện tốc độ thực thi JavaScript trong các trình duyệt web. Thay vì sử dụng trình thông dịch, V8 chuyển đổi mã JavaScript thành mã máy hiệu quả hơn để tăng hiệu suất. Nó biến mã JavaScript thành mã máy trong quá trình thực thi bằng cách sử dụng trình biên dịch JIT (Just-In-Time), cũng như nhiều công cụ JavaScript hiện tại như SpiderMonkey hoặc Rhino (Mozilla).

Middleware:

Cập nhật hoặc sửa đổi yêu cầu và các đối tượng phản hồi

Kết thúc chu kỳ yêu cầu-phản hồi

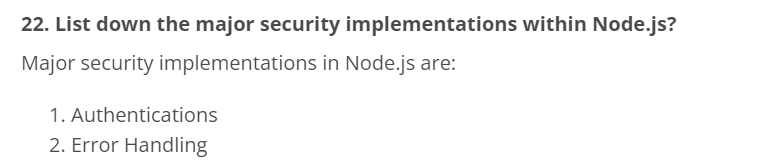
Gọi phần mềm trung gian tiếp theo trong ngăn xếp



<https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js>

JSON Web Token (JWT) là một tiêu chuẩn mở cung cấp một phương pháp nén và độc lập để truyền dữ liệu an toàn giữa các bên dưới dạng một đối tượng JSON.

https://www.edureka.co/blog/interview-questions/top-node-js-interview-questions-2016/



NodeJS vs Ajax

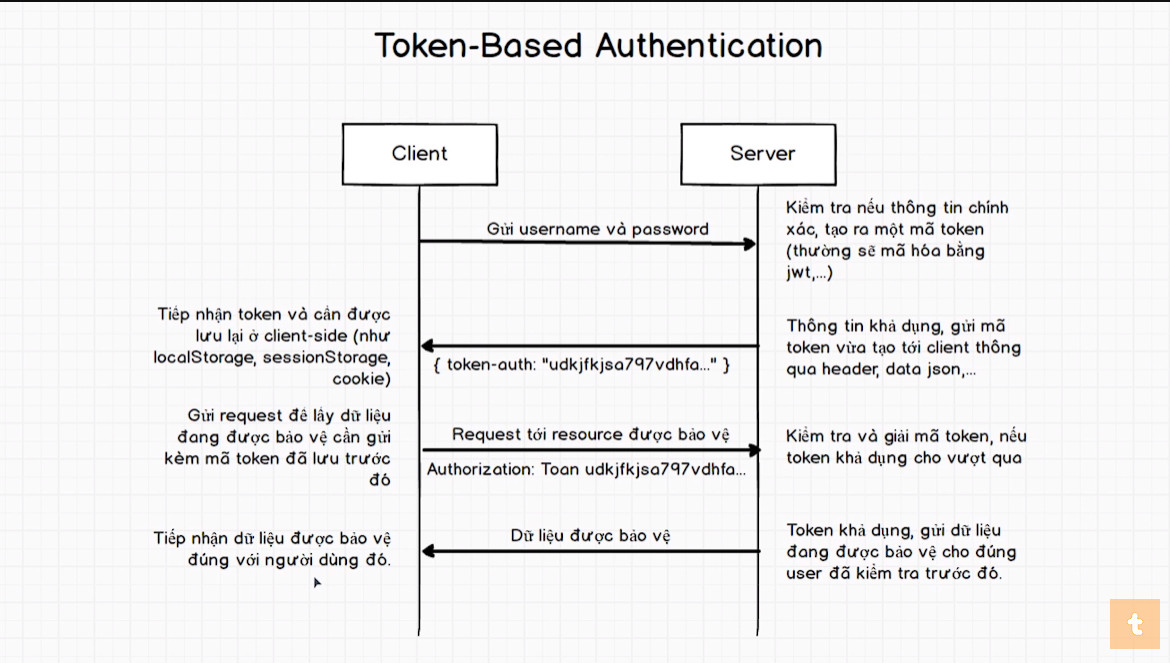
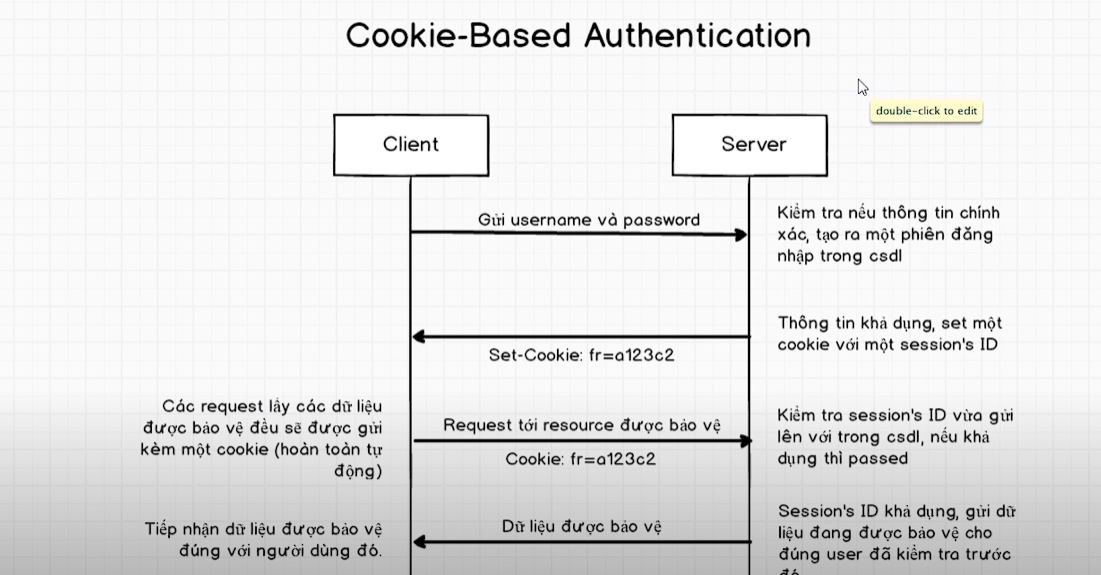
Sự khác biệt cơ bản nhất giữa Node.js và Ajax đó là, Node.js là một JavaScript phía máy chủ trong khi Ajax là một công nghệ phía máy khách. Nói một cách đơn giản hơn, Ajax chủ yếu được sử dụng để cập nhật hoặc sửa đổi nội dung trang web mà không cần phải làm mới nó. Mặt khác, Node.js được yêu cầu phát triển phần mềm máy chủ thường được thực thi bởi các máy chủ thay vì trình duyệt web

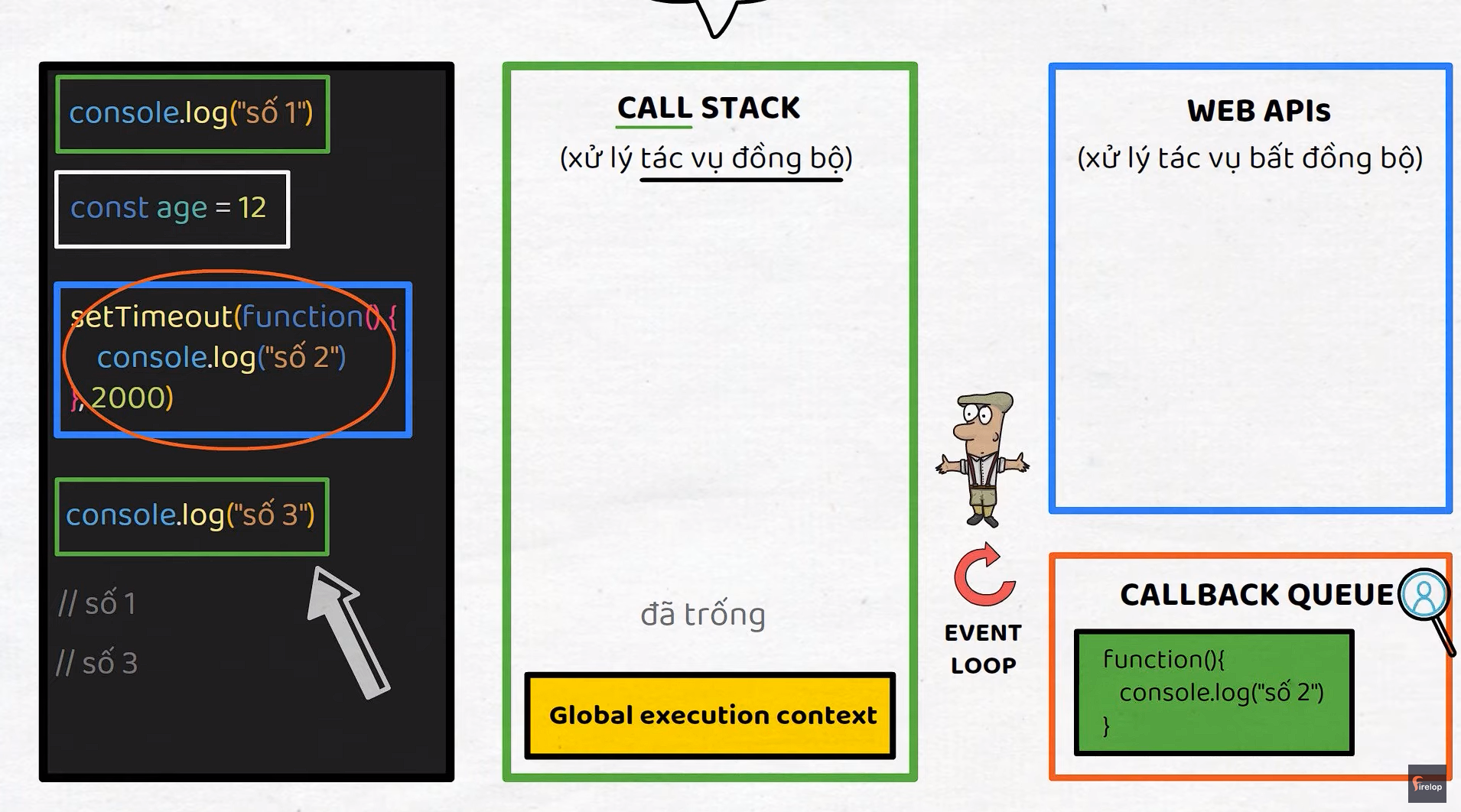
Array unique let uniqueItems = [...new Set(items)];

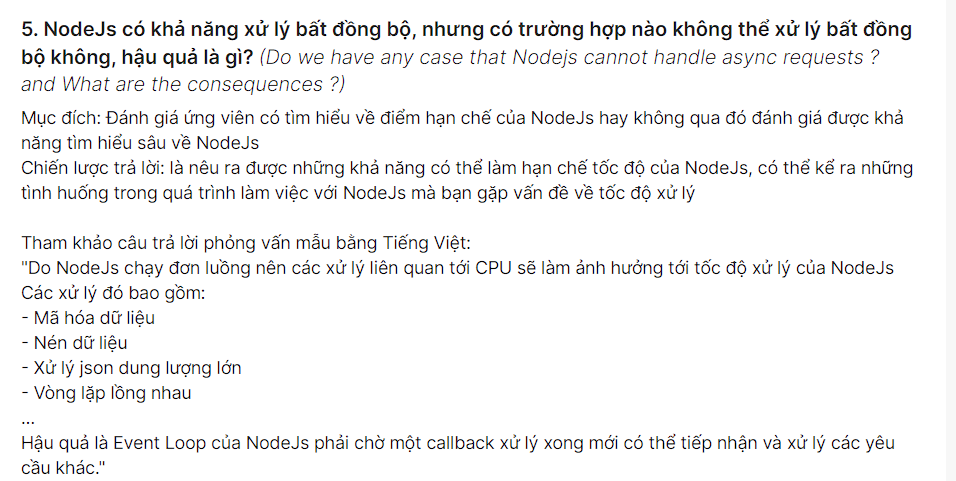
NodeJS là môi trường thời gian chạy được sử dụng để xây dựng các ứng dụng phía máy chủ trong khi AngularJS là một khung JavaScript chủ yếu hữu ích trong việc xây dựng/phát triển phần ứng dụng phía máy khách chạy bên trong trình duyệt web.

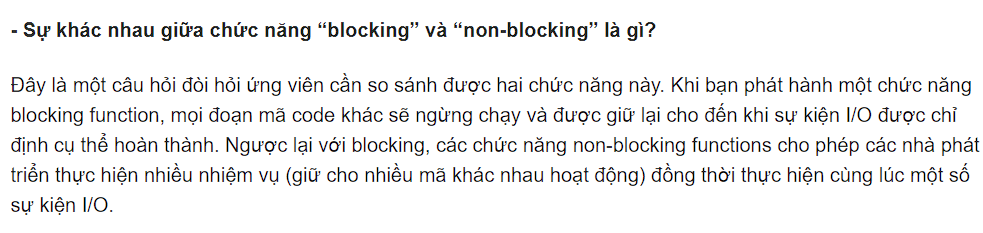
* Const let hoist phần khai báo lên trên đầu phạm vi nhưng không khởi tạo giá trị,đưa vào temporal dead zone

Cookie base and token base









**Closure:** Hàm bên trong có thể truy cập các biến của hàm bên ngoài .

**✅ Ưu điểm**:

* Có schema và model tường minh
* Có sẵn schema validation ở tầng ứng dụng giúp tăng cường thêm tính nhất quán cho database
* Cung cấp các hàm API để thao tác với MongoDB một cách dễ dàng

**❌ Nhược điểm**:

* Tốn thời gian học thêm thư viện
* Bị giới hạn bởi thư viện, tính tùy biến không cao bằng việc sử dụng MongoDB Driver dẫn đến hiệu suất sẽ chậm hơn khi thực hiện các câu lệnh phức tạp
* Việc tương tác với MongoDB qua Mongoose có thể tạo ra các lỗi hoặc xung đột bởi vì các hàm API được định nghĩa bởi Mongoose, vì nó không được phát hành chính thức bởi MongoDB.

Mongoose vs mongdb

* Mongoose được viết bằng JavaScript trên trình điều khiển MongoDB gốc. Vì vậy, vâng, sử dụng nó sẽ khiến bạn bị phạt về hiệu suất. Nhưng các vấn đề thậm chí còn lớn hơn phát sinh khi bạn cố gắng sử dụng nó như một ORM. Thiết kế tồi sẽ làm giảm hiệu suất của bất kỳ ODM hoặc cơ sở dữ liệu nào.
* Nhưng có một cách giải quyết: MongoDB. Mongoose cung cấp quyền truy cập trực tiếp vào trình điều khiển gốc được sử dụng dưới mui xe. Nếu có một phần quan trọng về hiệu suất trong ứng dụng của bạn, bạn có thể sử dụng toàn bộ sức mạnh của MongoDB trong khi vẫn có tùy chọn Mongoose trong phần còn lại của ứng dụng.

Login google:

Tạo ứng dụng

Library-> google + API

Map return về giá trị mới còn foreach không tạo ra mảng mới