

1. Callback hell là gì?

Callback hell là trạng thái nhiều callback lồng vào nhau nhiều cấp khiến cho việc đọc hiểu code trở nên cực kì khó khăn

VD:

```
let doSomething(function() {  
    doSomethingAfter(function() {  
        doSomethingAfterAfter(function() {  
            });  
        });  
    });  
});
```

2. Làm sao để không gặp phải callback hell

Phương pháp chính là phân tách code thành nhiều function nhỏ, mỗi callback được viết thành một function riêng và gọi tới function xử lý tiếp theo hoặc sử dụng async await

3. NodeJs đơn luồng hay đa luồng?

NodeJs đơn luồng nghĩa là nói tới việc trong NodeJs chỉ có một tác vụ Javascript được xử lý tại một thời điểm

Bản thân NodeJs khi chạy sử dụng một luồng chính để thực thi Javascript, là nơi event loop chạy và một thread pool chứa n luồng để Node dùng cho các xử lý nội bộ

4. Điều gì khiến NodeJs có thể chạy các tác vụ bất đồng bộ

NodeJs sử dụng một cơ chế có tên gọi là Event Loop để xử lý bất đồng bộ các tác vụ I/O

Event Loop cho phép NodeJs có thể xử lý đồng thời cùng lúc nhiều tác vụ mà không cần phải đợi tác vụ trước đó xử lý xong

Khi các tác vụ hoàn tất, Event Loop sẽ trả kết quả về dưới dạng các callback.

5. NodeJs có khả năng xử lý bất đồng bộ, nhưng có trường hợp nào không thể xử lý bất đồng bộ không, hậu quả là gì?

Do Nodejs chạy đơn luồng nên các xử lý liên quan tới CPU sẽ làm ảnh hưởng tới tốc độ xử lý của Nodejs

Các xử lý đó bao gồm:

- Mã hóa dữ liệu
- Nén dữ liệu
- Xử lý json dung lượng lớn
- Vòng lặp lồng nhau

...

Hậu quả là Event Loop của Nodejs phải chờ một callback xử lý xong mới có thể tiếp nhận và xử lý các yêu cầu khác

6. NODE_ENV là gì, cách sử dụng ?

NODE_ENV là một biến môi trường được chứa trong khung máy chủ web express. Khi một ứng dụng được chạy, nó có thể kiểm tra giá trị của biến môi trường và làm những việc khác nhau dựa trên giá trị đó.

VD: `process.env.NODE_ENV`

7. Phân biệt ES module và commonJS?

Cả hai Module là một cách để tái sử dụng code được linh hoạt hơn, đây cũng là cách để chúng ta tương tác giữa các file JavaScript với nhau.

ECMAScript modules sử dụng cú pháp `import`, `export`. Đuôi file phải chấm `mjs`

commonJS sử dụng cú pháp `module.export`, `require`. Đuôi file chấm `js`

8. Callback function là gì ?

Trong `node.js`, về cơ bản chúng ta sử dụng callback để xử lý các hoạt động không đồng bộ như - thực hiện bất kỳ yêu cầu I / O nào, hoạt động cơ sở dữ liệu hoặc gọi API để tìm nạp một số dữ liệu

9. How to migrate from a monolithic app to a microservice system?:

<https://www.youtube.com/watch?v=GoDS0X7RTyI>

10. Restful API , graphql là gì ?

11. Thao tác các file hệ thống thì làm thế nào?

Sử dụng FS (file system):

12. Tại sao sử dụng Express.js?

ExpressJS là một framework NodeJS được xây dựng sẵn có thể giúp bạn tạo các ứng dụng web phía máy chủ nhanh hơn và thông minh hơn. Tính đơn giản, tối giản, tính linh hoạt, khả năng mở rộng là một số đặc điểm của nó và vì nó được tạo ra trong chính NodeJS nên nó cũng thừa hưởng hiệu suất của nó.

- Different between b-tree and hash
- Different between TRUNCATE and DROP, TRUNCATE and DELETE
- How to secure a JWT token because anyone/middleman could get the token and decode it?
- Algorithm challenges:
- How do you check the API when it does not meet the expected response time?
- Response headers contain which properties?
- What is CORS / How does the client deal with it?
- Different between session authorization vs token authorization?
- How to choose the right column for indexing?
- How do you know that the SQL query hits the indices

Tham khảo thêm:

<https://github.com/learning-zone/nodejs-interview-questions#q-what-does-the-runtime-environment-mean-in-nodejs>