

## Slot5

Xử lý không đồng bộ trong Javascript

Tại sao????

- Tránh request nhiều lần, tốn thời gian
- Trình duyệt xử lý song song
- Không block UI

=> bắt buộc phải xử lý không đồng bộ

Xử lý đồng bộ như thế nào????

Callback → promise → async/await

ví dụ: async/await

```
async function loadData(){  
    const response = await fetch(url); //đọc dữ liệu từ đường dẫn  
    const data = await response.json(); //chuyển sang json  
}
```

====

Fetch API????

- Tích hợp sẵn trong browser
- Dùng để gửi http request
- trả về Promise

ví dụ:

```
fetch(url)  
    .then(response => response.json())  
    .then(data => console.log(data));
```

Áp dụng trong React JS

=> fetch dữ liệu trong React Component

```
useEffect(()=>{  
    fetch("/api/products")  
        .then(res => res.json())  
        .then(data => setProducts(data));  
},[]);
```

Lý do phải đặt trong useEffect:

- để tránh gọi API nhiều lần (chỉ gọi 1 lần duy nhất khi component mount)

=====

Gửi dữ liệu lên server thì làm thế nào????

Ví dụ: Thêm mới => POST

```
fetch("/api/products", {
    method: "POST",
    header: {
        "Content-Type": "application/json",
    },
    body: JSON.stringify(product),
});
```

Ví dụ: sửa, xóa

```
fetch(`/api/product/${id}`, {method: "DELETE"});
```

=====

Mã lỗi:

2xx: response.ok

4xx → lỗi client

5xx → lỗi server

=====

Cors????

- browser chặn request khác domain vì lý do bảo mật
- => cần cấu hình CORS ở SpringBoot hoặc dùng proxy

=====

Bootstrap: định dạng trang web

Navbar:

- tạo menu điều hướng, có logo, button, icon,...

Form:

- tạo form đăng nhập, đăng ký, tìm kiếm

Modal:

- Hộp thoại bật lên để hiển thị thông báo hoặc nhập liệu

-Thường dùng cho Xác nhận, Thêm mới, Chính sửa

=====