**Tìm hiểu Axios, Fetch () và sự khác biệt giữa chúng.**

# **1.Cài đặt Axios và sử dụng**

## Tổng quan Axios.

- Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho lý việc xử XHR (XMLHttpRequests):

* Request and response interception
* Streamlined error handling
* Protection against XSRF
* Support for upload progress
* Response timeout
* The ability to cancel requests
* Support for older browsers
* Automatic JSON data transformation
* Mutil request

## Cài đặt Axios.

- Cài đặt với lệnh **npm install axios** hoặc **yarn add axios.**

- Tạo requests:

// send a POST request

axios({

method: 'post',

url: '/login',

timeout: 4000, // 4 seconds timeout

data: {

firstName: 'Finn',

lastName: 'Williams'

}

});

* Axios tự động chuyển đổi dữ liệu thành JSON và gửi dưới dạng request body.

## Shorthand methods:

* axios.request(config)
* axios.get(url[, config])
* axios.delete(url[, config])
* axios.head(url[, config])
* axios.options(url[, config])
* axios.post(url[, data[, config]])
* axios.put(url[, data[, config]])
* axios.patch(url[, data[, config]])

Link tham khảo : <https://blog.logrocket.com/how-to-make-http-requests-like-a-pro-with-axios/#new_ta>

## Response

* Kết quả tả về là 1 object: response or error
* Success object:
* data: Payload được trả về từ server. Mặc định, Axios mong chờ JSON và sẽ parse nó trở thành một Javascript Object cho bạn.
* status: mã HTTP được trả về từ server.
* statusText: message của HTTP status được trả về bởi server.
* headers: tất cả các header được gửi về từ server.
* config: thiết lập ban đầu của request.
* request: một object XMLHttpRequest thực khi request được thực hiện ở trình duyệt.
* Error object:
  + message: message của error ở dạng text.
  + response: response object được trả về.
  + config: thiết lập ban đầu của request.
  + request: một object XMLHttpRequest thực khi request được thực hiện ở trình duyệt.

## Xử lý dữ liệu

* **Axios** cho phép bạn cung cấp các function để biến đổi dữ liệu đến và đi, dưới dạng 2 config của **Transforms**: transformRequest và transformResponse . Chỉ nhận dữ liệu hay headers. Giúp format dữ liệu theo ý.  có thể set up một vài hàm transform để convert dữ liệu gửi đi và trả về. Mặc định, hàm transformRequest và transformResponse biến đổi dữ liệu thành JSON.
* **Ngoài ra Axios cũng cấp thêm interceptors.request && interceptors.response. Interceptors nhận hoàn toàn response, request,** có thể chọn cung cấp một hàm xử lý error cho phép bạn bắt và xử lý lỗi. có thể được sử dụng để thực hiện những điều như: lấy token từ localStorage và gửi kèm các request. Response interceptor có thể được dùng để bắt lỗi 401 và redirect đến trang login trong việc xác thực.

## Third-Party Add-Ons

* [axios-mock-adaptor](https://github.com/ctimmerm/axios-mock-adapter): cho phép bạn dẽ dàng mock request để thực hiện việc kiểm tra code.
* [axios-cache-plugin](https://github.com/jin5354/axios-cache-plugin): một wrapper được dùng để lựa chọn lưu Cache các GET request.
* [redux-axios-middleware](https://github.com/svrcekmichal/redux-axios-middleware): nếu bạn dùng Redux, middleware này cung cấp một cách gọn gàng để dispatch một Ajax request với các actions. Một danh sách các [add-ons và extensions khác](https://github.com/axios/axios/blob/master/ECOSYSTEM.md) có sẵn trên Github.

# **2.Fetch ()**

* Fetch API là một API đơn giản cho việc gửi và nhận requesst bằng js. Với fetch thì việc thực hiện các yêu cầu web và xử lý phản hồi dễ dàng hơn so với XMLHttpRequest cũ.

// fetch()

const url = 'http://localhost/test.htm';

const options = {

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json;charset=UTF-8'

},

body: JSON.stringify({

a: 10,

b: 20

})

};

fetch(url, options)

.then(response => {

console.log(response.status);

});

# **3.So sánh Axios vs Fetch ()**

* Để gửi dữ liệu, fetch () sử dụng property body, trong khi Axios sử dụng data property
* Dữ liệu trong fetch () được xâu chuỗi (cần được ép kiểu)
* URL được truyền dưới dạng argument trong fetch (). Tuy nhiên, trong Axios, URL được đặt trong options object.
* Axios sử dụng default headers còn Fetch () thì ko có.

Link tham khảo : <https://blog.logrocket.com/axios-or-fetch-api/>