



NETWORKING

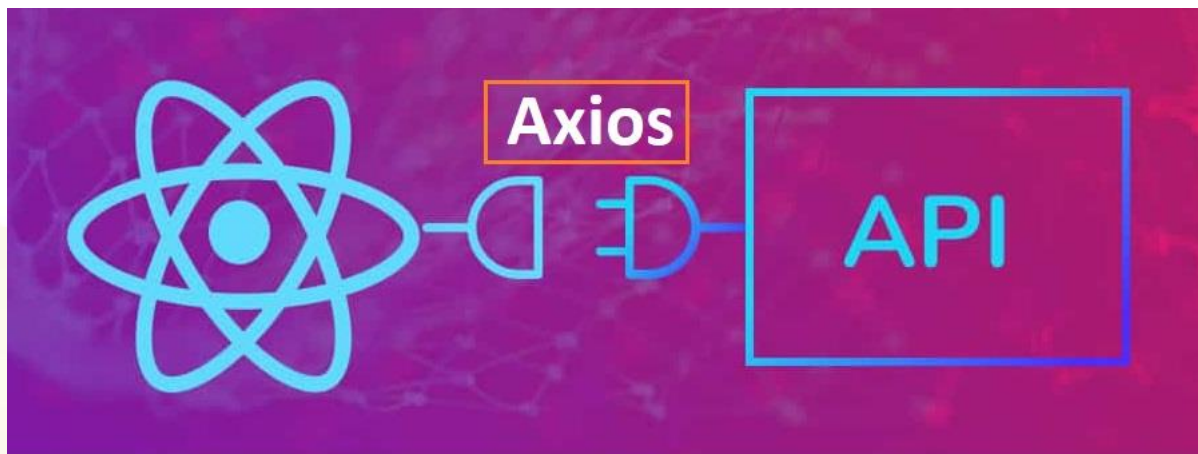
CYBERSOFT.EDU.VN



- ☐ Axios là gì?
- ☐ Gửi request với axios.
- ☐ Một số thuộc tính thường dùng.
- ☐ Các phương thức hay dùng.
- ☐ Xử lý response.
- ☐ Xử lý lỗi.

CYBERSOFT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



- ❑ **Axios** là một thư viện HTTP Client được tạo ra với mục đích xử lý request và response khi làm việc với Restful API.
- ❑ **Axios** được viết hoàn toàn bằng javascript, chạy ở client. Nhiệm vụ chủ yếu của nó là kết nối client với các API của server.
- ❑ **Cài đặt Axios:**

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

Request với Axios

- ❑ Bạn có thể tạo bất kỳ một request HTTP nào bằng cách truyền vào các object option cho Axios:

```
axios({  
  method: 'post',  
  url: '/signup',  
  data: {  
    email: 'admin@gmail.com',  
    fullname: 'Nguyễn Văn Tèo',  
    password: '123456'  
  }  
});
```

- ❖ **method**: phương thức HTTP muốn dùng (GET/POST/DELETE...).
- ❖ **url**: đường dẫn được sử dụng để gửi request lên.
- ❖ **data**: dữ liệu để có thể gửi kèm với request. Mặc định, axios sẽ serialize các object này thành JSON và gắn chúng ở phần body của request.

Các option cho Request

- ❑ **baseURL**: base URL sẽ được đính vào trước bất cứ một URL tương đối nào mà bạn sử dụng.
- ❑ **headers**: một object gồm các cặp key/value có thể gửi trong header của request.
- ❑ **params**: một object gồm các cặp key/value mà sẽ được serialize và đính vào URL dưới dạng một query string.
- ❑ **responseType**: Chỉ định kiểu dữ liệu trả về từ server (mặc định là json).
- ❑ **data**: dữ liệu sẽ được gửi theo body của request.
- ❑ **timeout**: chỉ định số mili giây khi request vượt quá thời gian truy cập và bị hủy bỏ.
- ❑ **maxContentLength**: chỉ định độ dài tối đa mà response được trả về.

Các option cho Request

```
axios({
  // `url` là đích đến của request
  url: '/user',
  // `method` là phương thức được sử dụng để thực hiện request
  method: 'get', // mặc định là GET
  // `baseUrl` sẽ được gán vào trước url khi url là đường dẫn tương đối.
  baseUrl: 'https://some-domain.com/api/',
  // `headers` là các header được đặt lại trước khi gửi lên server
  headers: {'X-Requested-With': 'XMLHttpRequest'},
  // `params` là các tham số URL sẽ được gửi lên cùng request
  // Giá trị của nó phải là một object thuần hoặc là một đối tượng URLSearchParams
  params: { ID: 12345 },
  // `data` là dữ liệu sẽ được gửi theo body của request
  // Chỉ khả dụng cho các request có phương thức là 'PUT', 'POST', và 'PATCH'
  data: {
    fullname: 'Nguyễn Văn Tèo'
  },
  // `timeout` chỉ định số mili giây khi request vượt quá thời gian truy cập và bị hủy bỏ
  timeout: 1000,
  // `responseType` chỉ định kiểu dữ liệu mà server sẽ trả về
  // có thể là 'arraybuffer', 'blob', 'document', 'json', 'text', 'stream'
  responseType: 'json', // default
  // `maxLength` chỉ định độ dài tối đa mà response được trả về
  maxLength: 2000
});
```

Request với GET

```
axios.get('/user?ID=12345')  
  .then(function (response) {  
    console.log(response);  
  })  
  .catch(function (error) {  
    console.log(error);  
  });
```

```
axios.get('/user', {  
  params: {  
    ID: 12345  
  }  
})  
  .then(function (response) {  
    console.log(response);  
  })  
  .catch(function (error) {  
    console.log(error);  
  });
```

Request với POST



```
axios.post('/users', {  
  firstName: 'First name',  
  lastName: 'Last name'  
})  
  .then(function (response) {  
    console.log(response);  
  })  
  .catch(function (error) {  
    console.log(error);  
  });
```

SOFT
LẬP TRÌNH

Phương thức request


- ❑ `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]])`

❖ **Note:** *Khi sử dụng các alias này thì url, method, và data không phải cần phải chỉ định trong config nữa.*


Axios response

- ❑ Khi một request được thực hiện thành công, thì hàm **then()** của bạn sẽ nhận được một response object với các thuộc tính như sau:
 - ✓ **data**: dữ liệu được trả về từ server (mặc định axios 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.

Axios response



```
.then(function (response) {  
  // `data` là dữ liệu trả về được cung cấp bởi server  
  console.log(response.data);  
  
  // `status` là mã HTTP status trả về từ server  
  console.log(response.status);  
  
  // `statusText` là thông điệp HTTP status trả về từ server  
  console.log(statusText);  
  
  // `headers` là các header mà server phản hồi  
  // Các tên của header được viết thường toàn bộ  
  console.log(headers);  
  
  // `config` là các cấu hình khi thực hiện request  
  console.log(config);  
  
  // `request` là request thực hiện để nhận được response này  
  console.log(request);  
})
```



- ❑ Nếu có bất kỳ vấn đề gì xảy ra với request, promise sẽ bị reject với một error object chứa những thuộc tính sau đây.
 - ✓ **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.

Tổng kết

- ✓ Axios là gì?
- ✓ Gửi request với axios.
- ✓ Một số thuộc tính thường dùng.
- ✓ Các phương thức hay dùng.
- ✓ Xử lý response.
- ✓ Xử lý lỗi.

CYBERSOFT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH