**AXIOS NÂNG CAO**

A – LÝ THUYẾT

I – AXIOS METHOD

**1. Response Object**

Khi một request được thực hiện thành công thì hàm then() sẽ nhận được một Response Object với đầy đủ các thuộc tính như: data, status, statusText, headers, config, request

**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const [data, setData] = React.useState([]);  axios({  method: "GET",  url: "https://jsonplaceholder.typicode.com/users",  data: {},  status: 200,  statusText: 'OK',  headers: {},  config: {},  request: {}  }).then((response)=>{  console.log(response.data);  }).catch((error)=>{  console.log(error);  });  return(  <h1>Axios</h1>  )  }  export default App; |

**Trong đó:**

* **method**: phương thức truyền dữ liệu được sử dụng
* **url**: đường dẫn đến API
* **data**: dữ liệu cần lấy từ máy chủ
* **status**: mã trạng thái HTTP của yêu cầu
* **statusText**: mô tả trạng thái tương ứng với mã trạng thái ở trên
* **headers**: thông tin phản hồi về header, chính là response
* **config**: các thông sẽ được thiết lập trước khi gửi request (sẽ nghiên cứu kỹ phần này sau)
* **request**: là thành phần của ClientRequest nếu sử dụng Node.js và XMLHttpRequest

**2. GET Request**

Khi người dùng truy cập vào các URL để gửi lên API thì chúng ta có thể sử dụng GET để gửi yêu cầu

**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const eventHandler = ()=>{  axios.get("https://jsonplaceholder.typicode.com/users/")  .then((response)=>{  console.log(response);  console.log(response.data);  });  }  return (  <button onClick={eventHandler}>Action</button>  )  }  export default App; |

**3. POST/PUT Requests**

Khi người dùng nhập dữ liệu qua Form và submit gửi lên API thì chúng ta có thể sử dụng POST/PUT để gửi yêu cầu

**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const user = {  username: "vietpro",  email: "vietpro.edu.vn@gmail.com"  }  const eventHandler = ()=>{  axios.post("https://jsonplaceholder.typicode.com/users/", user)  .then((response)=>{  console.log(response);  console.log(response.data);  });  }  return (  <button onClick={eventHandler}>Action</button>  )  }  export default App; |

**4. DELETE Requests**

Chúng ta có thể xóa một hoặc nhiều phần tử khỏi API axios.delete bằng cách truyền thêm vào URL với tham số ID

**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const userId = 2  const eventHandler = ()=>{  axios.delete(`https://jsonplaceholder.typicode.com/users/${userId}`)  .then((response)=>{  console.log(response);  console.log(response.data);  });  }  return (  <button onClick={eventHandler}>Action</button>  )  }  export default App; |

II – AXIOS CONFIG

*Config trong Axios cho phép chúng ta bổ sung thêm nhiều ràng buộc về dữ liệu trong quá trình thao tác với API*

**1. Config với params là giá trị đơn**

**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const config = {  params: {  id: 5  }  }  axios.get("https://jsonplaceholder.typicode.com/users/",  config).then((res) => {  console.log(res.data);  });  return (  <h1>Axios</h1>  )  }  export default App; |

**Trong đó:**

* params: sẽ nhận vào một đối tượng lưu trữ một hoặc nhiều thuộc tính cần lọc

**2. Config với params là một mảng**

**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const config = {  params: {  id: [2,4,5,7,9]  }  }  axios.get("https://jsonplaceholder.typicode.com/users/",  config).then((res) => {  console.log(res.data);  });  return (  <h1>Axios</h1>  )  }  export default App; |

III – AXIOS INSTANCE

Tạo một Instance bằng module Axios cho phép chúng ta thao tác với Axios một cách đơn giản và linh hoạt hơn rất nhiều

**1. Tạo một Instance với Axios**

**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const Http = axios.create({  baseURL: "https://jsonplaceholder.typicode.com/users/"  });  Http.get().then((res) => {  console.log(res.data);  });  return (  <h1>Axios</h1>  )  }  export default App; |

**2. Axios Instance với custom config**

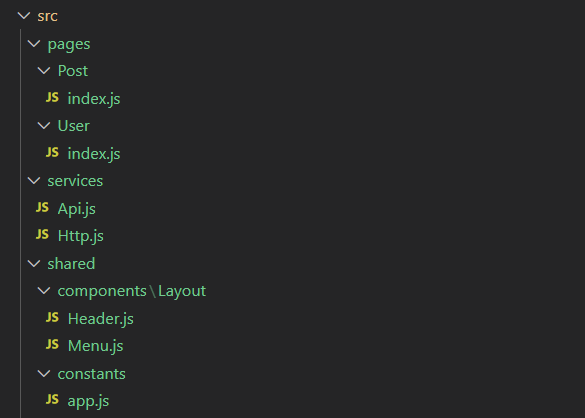
**Code:**

|  |
| --- |
| import React from "react";  import axios from "axios";  const App = () => {  const Http = axios.create({  baseURL: "https://jsonplaceholder.typicode.com"  });  const config = {  params:{  id:1  }  }  Http.get("/users", config).then((res) => {  console.log(res.data);  });  return (  <h1>Axios</h1>  )  } |

B – THỰC HÀNH

I – BÀI TẬP THỰC HÀNH

**Bài 1:** Xây dựng cấu trúc ứng dụng quản lý dữ liệu theo mẫu sau

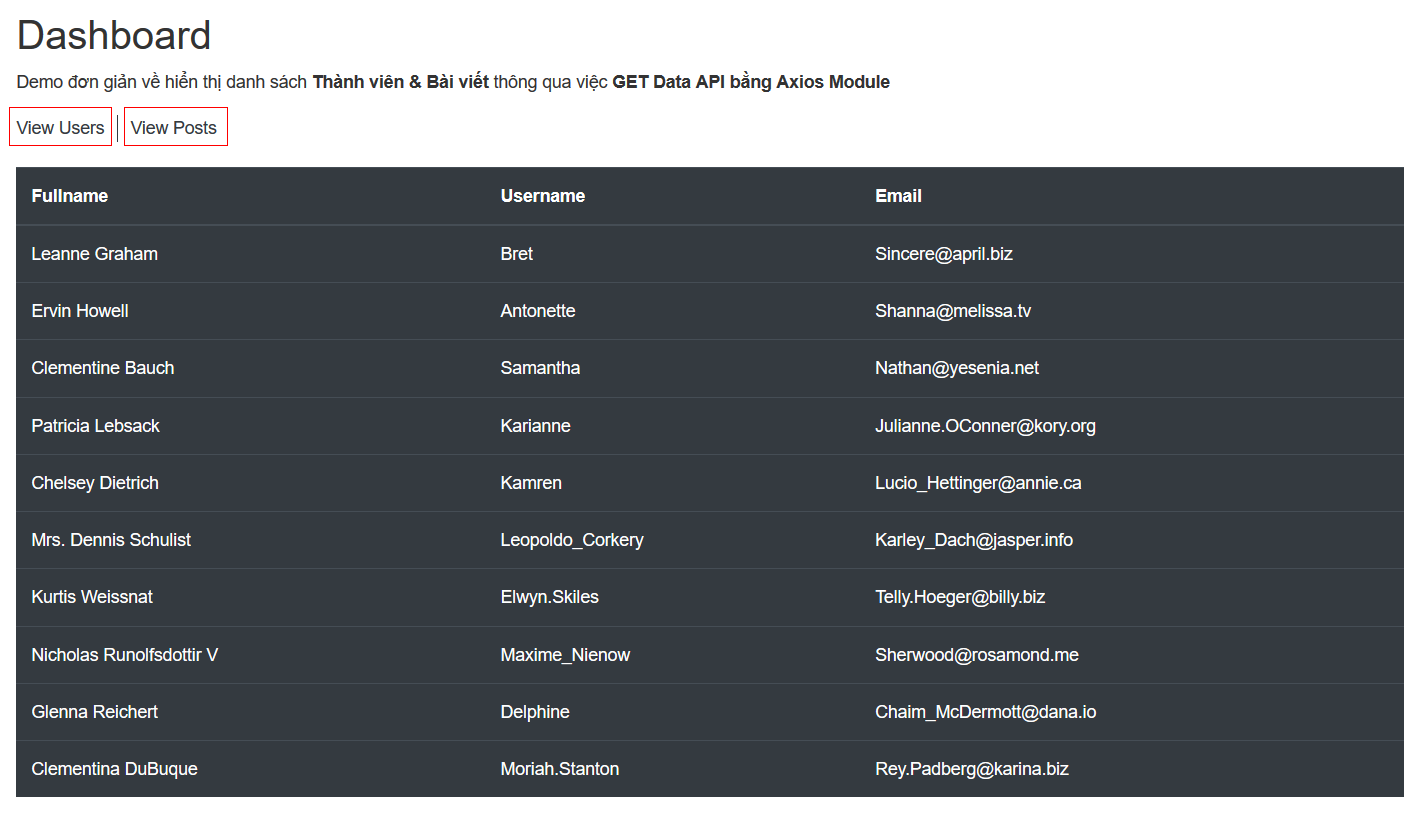


**Trong đó:**

* **pages**: thư mục chứa giao diện phần Main của tất cả các Page trên Website
* **Post**: thư mục chứa giao diện phần Main của trang hiển thị danh sách bài viết
* **User**: thư mục chứa giao diện phần Main của trang hiển thị danh sách thành viên
* **index.js**: file chạy chính (component)
* **services**: thư mục chứa các file liên quan đến việc khởi tạo Server và đọc API
* **Api.js**: file lấy dữ liệu từ các API
* **Http.js**: file xây dựng một Axios Instance từ Axios
* **shared**: thư mục chia sẻ tài nguyên dùng chung
* **components**: thư mục chứa các component sử dụng chung
* **Layout**: thư mục chứa các giao diện sử dụng chung
* **Header.js**: Component Header
* **Menu.js**: Component Menu
* **constants**: thư mục chứa các thành phần không đổi (thông thường được sử dụng cho config)
* **app.js**: các phân đoạn API được sử dụng chung cho tất cả các API

**Bài 2:** Xây dựng ứng dụng quản lý dữ liệu (User & Post) theo các yêu cầu sau

* Ứng dụng được khởi động thì sẽ hiển thị danh sách User (thành viên) theo API sau: <https://jsonplaceholder.typicode.com/users>
* Khi click vào menu Posts thì sẽ hiển thị danh sách Post (bài viết) theo API sau: <https://jsonplaceholder.typicode.com/posts>



II – BÀI TẬP VỀ NHÀ