

## LAB 5: KẾT NỐI API

### Bài 1: Tạo một ứng dụng đơn giản kết nối API

## Danh Sách Người Dùng

- Leanne Graham - Sincere@april.biz
- Ervin Howell - Shanna@melissa.tv
- Clementine Bauch - Nathan@yesenia.net
- Patricia Lebsack - Julianne.OConner@kory.org
- Chelsey Dietrich - Lucio\_Hettinger@annie.ca
- Mrs. Dennis Schulist - Karley\_Dach@jasper.info
- Kurtis Weissnat - Telly.Hoeger@billy.biz
- Nicholas Runolfsson - Sherwood@rosamond.me
- Glenna Reichert - Chaim\_McDermott@dana.io
- Clementina DuBuque - Rey.Padberg@karina.biz

Hướng dẫn:

**Bước 1:** vue create my\_bai1\_api

**Bước 2:** npm install axios

**Bước 3:** Tạo components/UserList.vue

```
<template>
  <div>
    <h1>Danh Sách Người Dùng</h1>
    <ul>
      <li v-for="user in users" :key="user.id">
        {{ user.name }} - {{ user.email }}
      </li>
    </ul>
    <p v-if="error">{{ error }}</p>
  </div>
</template>

<script>
import axios from 'axios';

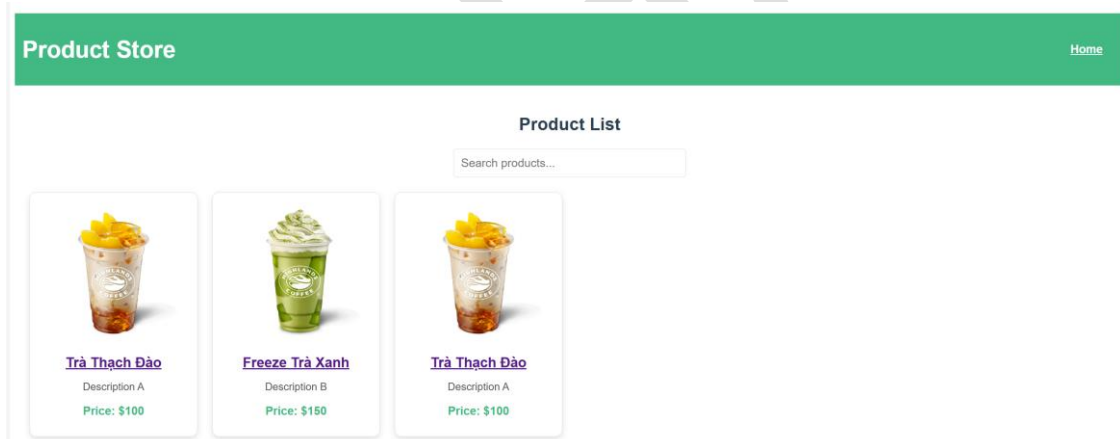
export default {
  data() {
    return {
      users: [],
      error: null
    };
  },
  mounted() {
    this.fetchUsers();
  },
  methods: {
    async fetchUsers() {
      try {
        const response = await axios.get('https://jsonplaceholder.typicode.com/users');
        this.users = response.data;
      } catch (error) {
        this.error = 'Có lỗi xảy ra khi lấy dữ liệu.';
      }
    }
  }
};
</script>
```

**Bước 4: App.vue**

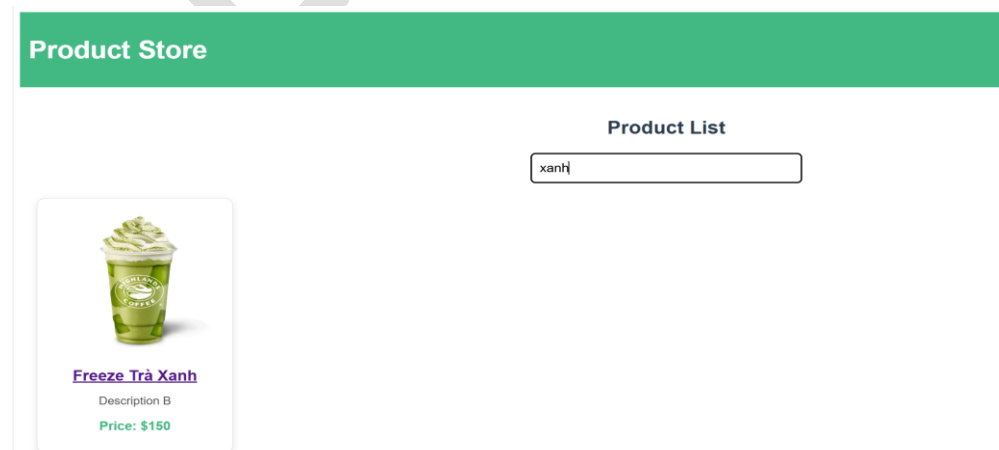
```
<template>
  <div id="app">
    <UserList />
  </div>
</template>

<script>
import UserList from './components/UserList.vue';

export default {
  name: 'App',
  components: {
    UserList
  }
};
</script>
```

**Bài 2: Hiển thị thông tin sản phẩm sử dụng API**

Tìm kiếm



Hướng dẫn:

### Tạo API với Express.js

**Bước 1:** Tạo thư mục server trong project Vue 3

```
mkdir server
```

```
cd server
```

**Bước 2:**

```
npm init -y
```

```
npm install express cors body-parser
```

**Bước 3:** Tạo file server.js trong product-api

```
const express = require('express');
```

```
const path = require('path');
```

```
const cors = require('cors');
```

```
const app = express();
```

```
// Sử dụng CORS để cho phép truy cập từ ứng dụng Vue
```

```
app.use(cors());
```

```
// Giả lập dữ liệu sản phẩm
```

```
const products = [
```

```
  { id: 1, name: 'Trà Thạch Đào', description: 'Description A', price: 100, image: 'trathachdao.jpg' },
```

```
  { id: 2, name: 'Freeze Trà Xanh', description: 'Description B', price: 150, image: 'traxanh.jpg' },
```

```
  { id: 3, name: 'Trà Thạch Đào', description: 'Description A', price: 100, image: 'trathachdao.jpg' },
```

```
];
```

```
// API lấy tất cả sản phẩm
```

```
app.get('/api/products', (req, res) => {
```

```
  res.json(products);
```

```
});
```

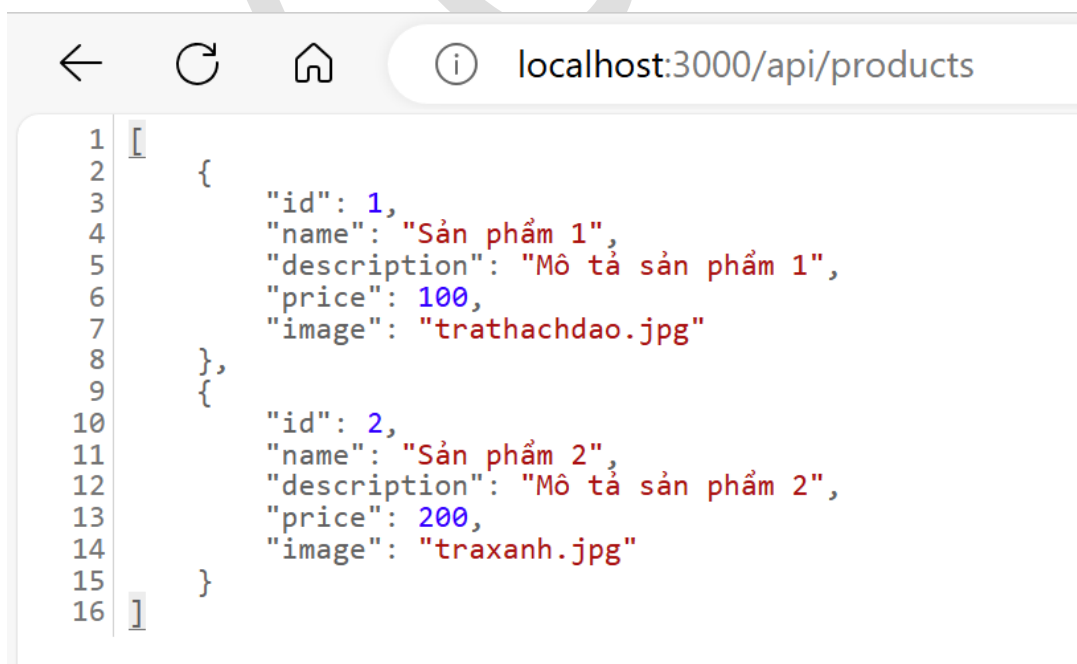
```
// API lấy chi tiết sản phẩm theo ID
app.get('/api/products/:id', (req, res) => {
  const productId = parseInt(req.params.id, 10);
  const product = products.find(p => p.id === productId);
  if (product) {
    res.json(product);
  } else {
    res.status(404).send('Product not found');
  }
});

app.listen(3000, () => {
  console.log('Server running on http://localhost:3000');
});
```

#### Bước 4: Chạy server

node index.js

Kiểm tra trong trình duyệt hoặc công cụ như Postman để đảm bảo rằng API đang hoạt động tại <http://localhost:3000/api/products>.



Tạo dự án với Vue 3 CLI

**Bước 1:** vue create product-app

**Bước 2:** Cài Đặt Axios

npm install axios

**Bước 3:** Cấu Hình Vue Router

npm install vue-router@4

hoặc

npm install vue-router

**Bước 4:** Tạo tệp cấu hình router

Tạo thư mục src/router và tạo file index.js:

```
import { createRouter, createWebHistory } from 'vue-router';
import ProductList from '../components/ProductList.vue';
import ProductDetail from '../components/ProductDetail.vue';

const routes = [
  {
    path: '/',
    name: 'Home',
    component: ProductList,
  },
  {
    path: '/product/:id',
    name: 'ProductDetail',
    component: ProductDetail,
  },
];

const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes,
});

export default router;
```

**Bước 5:** Cấu hình main.js

```
JS main.js
import { createApp } from 'vue';
import App from './App.vue';
import router from './router'; // Import router

createApp(App)
  .use(router) // Sử dụng router trong ứng dụng
  .mount('#app');
```

## Bước 6: Tạo các components

### ProductList.vue

```
<template>
  <div class="product-list-container">
    <h2>Product List</h2>
    <!-- Thanh tìm kiếm sản phẩm -->
    <input
      type="text"
      placeholder="Search products..."
      v-model="searchQuery"
      @input="filterProducts"
      class="search-input"
    />
    <div class="product-grid">
      <div
        v-for="product in filteredProducts"
        :key="product.id"
        class="product-item"
      >
        <router-link :to="{ name: 'ProductDetail', params: { id: product.id } }">
          
          <h3 class="product-name">{{ product.name }}</h3>
        </router-link>
        <p class="product-description">{{ product.description }}</p>
        <p class="product-price">Price: ${{ product.price }}</p>
      </div>
    </div>
  </div>
</template>
```

```
<script>
import axios from 'axios';

export default {
  name: 'ProductList',
  data() {
    return {
      products: [],
      searchQuery: '',
      filteredProducts: [],
    };
  },
  created() {
    axios.get('http://localhost:3000/api/products')
      .then(response => {
        this.products = response.data;
        this.filteredProducts = this.products; // Khởi tạo danh sách sản phẩm lọc
      })
      .catch(error => {
        console.error('There was an error fetching the products!', error);
      });
  },
  methods: {

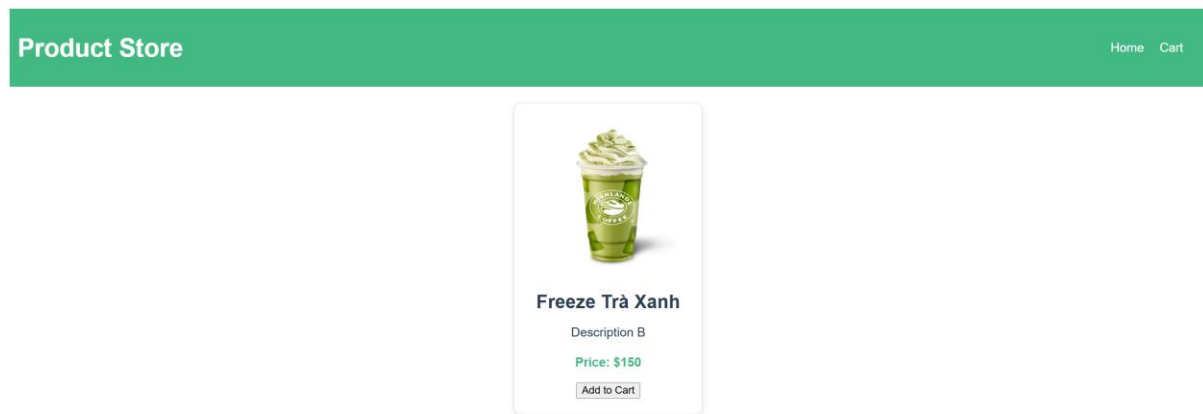
    filterProducts() {
      const query = this.searchQuery.toLowerCase();
      this.filteredProducts = this.products.filter(product =>
        product.name.toLowerCase().includes(query) ||
        product.description.toLowerCase().includes(query)
      );
    }
  }
};
</script>
```

## Bước 7: App.vue

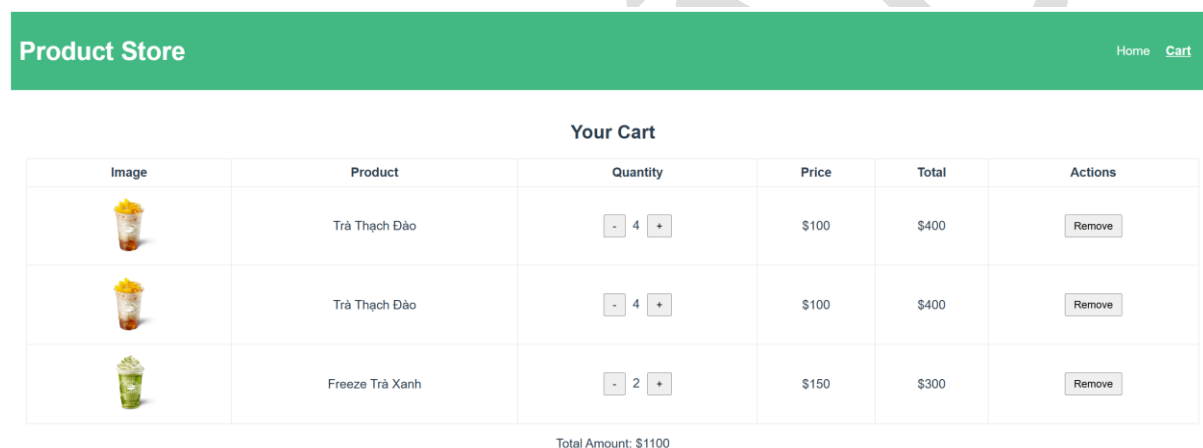
```
<template>
  <div id="app">
    <header>
      <h1>Product Store</h1>
      <!-- Sử dụng router-link để điều hướng giữa các trang -->
      <nav>
        <router-link to="/">Home</router-link>
      </nav>
    </header>

    <!-- router-view sẽ hiển thị component tương ứng với đường dẫn hiện tại -->
    <router-view></router-view>
  </div>
</template>

<script>
export default {
  name: 'App',
};
</script>
```

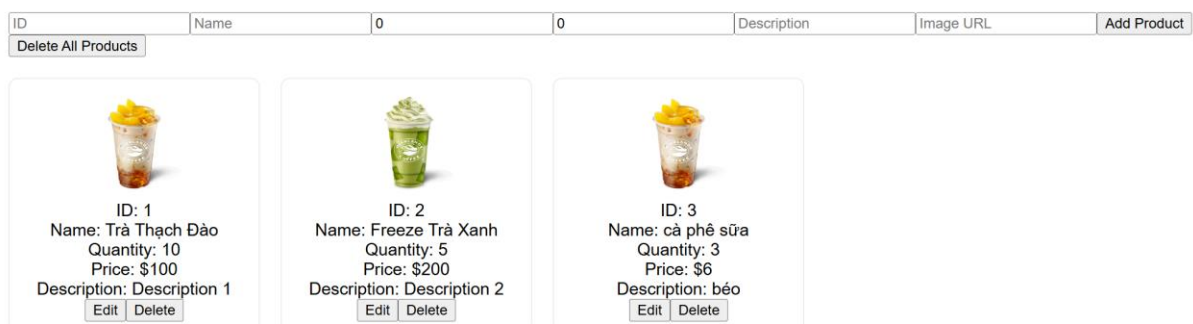
**Bài 3: Mở rộng bài 2 thêm phần chi tiết sản phẩm**

Yêu cầu click vào tên hàng, ra trang chi tiết 1 sản phẩm

**Bài 4: Mở rộng từ bài 3, thêm phần giỏ hàng, click vào Add to Cart thông tin sẽ được thêm vào trong giỏ hàng**

**Yêu cầu:**

- Click nút Add to cart sẽ được thêm thông tin vào giỏ hàng
- Trong giỏ hàng, cột quantity tăng thì giá total tăng theo và Total Amount cũng tăng, khi giảm thì ngược lại, nhưng giảm về 0 thì sản phẩm tự động xóa.

**Bài 5: Hiện thị thông tin sản phẩm thực hiện thêm, xóa, sửa****Product Manager**

Thêm sản phẩm



Product Manager

ID

Name

0


0

Description

Image URL


Add Product

Delete All Products




ID: 1  
Name: Trà Thạch Đào  
Quantity: 10  
Price: \$100  
Description: Description 1

EditDelete




ID: 2  
Name: Freeze Trà Xanh  
Quantity: 5  
Price: \$200  
Description: Description 2

EditDelete



ID: 3  
Name: cà phê sữa  
Quantity: 3  
Price: \$6  
Description: béo


EditDelete



Product Image

ID: 4  
Name: cà phê sữa  
Quantity: 3  
Price: \$39  
Description: Béo, vị đậm đà

EditDelete



ID: 5  
Name: Cà phê đen  
Quantity: 4  
Price: \$49  
Description: Vị đậm đà

EditDelete

Edit sản phẩm

Product Manager

4

cà phê sữa đá

3


39

Béo, vị đậm đà

images/caphesua.jpg


Update Product

Delete All Products




ID: 1  
Name: Trà Thạch Đào  
Quantity: 10  
Price: \$100  
Description: Description 1

EditDelete




ID: 2  
Name: Freeze Trà Xanh  
Quantity: 5  
Price: \$200  
Description: Description 2

EditDelete



ID: 3  
Name: cà phê sữa  
Quantity: 3  
Price: \$6  
Description: béo


EditDelete



Product Image

ID: 4  
Name: cà phê sữa  
Quantity: 3  
Price: \$39  
Description: Béo, vị đậm đà

EditDelete



ID: 5  
Name: Cà phê đen  
Quantity: 4  
Price: \$49  
Description: Vị đậm đà

EditDelete

Product Manager

ID

Name

0


0

Description

Image URL


Add Product

Delete All Products




ID: 1  
Name: Trà Thạch Đào  
Quantity: 10  
Price: \$100  
Description: Description 1

EditDelete




ID: 2  
Name: Freeze Trà Xanh  
Quantity: 5  
Price: \$200  
Description: Description 2

EditDelete




ID: 3  
Name: cà phê sữa  
Quantity: 3  
Price: \$6  
Description: béo

EditDelete



ID: 4  
Name: cà phê sữa đá  
Quantity: 3  
Price: \$39  
Description: Béo, vị đậm đà

EditDelete



ID: 5  
Name: Cà phê đen  
Quantity: 4  
Price: \$49  
Description: Vị đậm đà

EditDelete

Xóa sản phẩm

Product Manager

ID

Name

0


0

Description

Image URL


Add Product

Delete All Products




ID: 1  
Name: Trà Thạch Đào  
Quantity: 10  
Price: \$100  
Description: Description 1

EditDelete




ID: 2  
Name: Freeze Trà Xanh  
Quantity: 5  
Price: \$200  
Description: Description 2

EditDelete



ID: 3  
Name: cà phê sữa  
Quantity: 3  
Price: \$6  
Description: béo

EditDelete



ID: 5  
Name: Cà phê đen  
Quantity: 4  
Price: \$49  
Description: Vị đậm đà

EditDelete

Yêu cầu:

Thực hiện các chức năng thêm, xóa, sửa, và xóa tất cả

Hướng dẫn:

### Tạo API với Express.js

**Bước 1:** Tạo thư mục product-api\

```
mkdir product-api
```

```
cd product-api
```

**Bước 2:**

```
npm init -y
```

```
npm install express cors body-parser
```

**Bước 3:** Tạo file server.js trong product-api

```
const express = require('express');
```

```
const cors = require('cors');
```

```
const bodyParser = require('body-parser');
```

```
const path = require('path'); // Thêm dòng này
```

```
const app = express();
```

```
const port = 5000;
```

```
app.use(cors());
```

```
app.use(bodyParser.json());
```

```
// Cấu hình Express để phục vụ các tệp tĩnh từ thư mục public
```

```
app.use(express.static(path.join(__dirname, 'public')));
```

```
let products = [
```

```
  { id: 1, name: 'Trà Thạch Đào', quantity: 10, price: 100, description: 'Description 1',  
    image: 'images/trathachdao.jpg' },
```

```
  { id: 2, name: 'Freeze Trà Xanh', quantity: 5, price: 200, description: 'Description 2',  
    image: 'images/traxanh.jpg' }  
];
```

```
// Lấy tất cả sản phẩm
```

```
app.get('/api/products', (req, res) => {
```

```
  res.json(products);
```

```
});
```

```
// Thêm sản phẩm mới
app.post('/api/products', (req, res) => {
  const newProduct = req.body;
  newProduct.id = products.length ? products[products.length - 1].id + 1 : 1;
  products.push(newProduct);
  res.status(201).json(newProduct);
});
```

```
// Cập nhật sản phẩm theo ID
app.put('/api/products/:id', (req, res) => {
  const productId = parseInt(req.params.id, 10);
  const updatedProduct = req.body;
  let productFound = false;
  products = products.map(product => {
    if (product.id === productId) {
      productFound = true;
      return { ...product, ...updatedProduct };
    }
    return product;
  });
```

```
  if (productFound) {
    res.json(updatedProduct);
  } else {
    res.status(404).send('Product not found');
  }
});
```

```
// Xóa sản phẩm theo ID
app.delete('/api/products/:id', (req, res) => {
  const productId = parseInt(req.params.id, 10);
  const initialLength = products.length;
  products = products.filter(product => product.id !== productId);
```

```
if (products.length < initialLength) {  
  res.status(204).end();  
} else {  
  res.status(404).send('Product not found');  
}  
});
```

// Xóa tất cả sản phẩm

```
app.delete('/api/products', (req, res) => {  
  products = [];  
  res.status(204).end();  
});
```

```
app.listen(port, () => {  
  console.log(`Server running at http://localhost:${port}`);  
});
```

Bước 6: Chạy server trong terminal

node server.js

Kiểm tra thử [localhost:5000/api/products](http://localhost:5000/api/products)

### Tạo dự án Product-manager

**Bước 1:** Tạo một dự án Vue 3

vue create product-manager

cd product-manager

**Bước 2:** Cài đặt Axios

Cài đặt Axios

**Bước 3:** Tạo các components: ProductManager.vue