

Vue3 和 API 整合

在 Vue 3 中進行 HTTP 請求並處理 API 響應是一個常見的需求，這通常是用於從後端服務獲取資料或者向後端服務提交資料。這裡我們會透過一個簡單的範例，展示如何在 Vue 3 應用中整合並使用 API。

使用 Axios 進行 HTTP 請求

Axios 是一個基於 Promise 的 HTTP 客戶端，用於瀏覽器和 node.js，非常適合在 Vue 應用中使用來發起 HTTP 請求。首先，我們需要安裝 Axios：

```
npm install axios
```

創建一個 Vue 3 組件

在這個範例中，我們將創建一個 Vue 3 組件，用於顯示從 API 獲取的資料。這個組件將使用 Axios 發起 HTTP 請求並處理響應。

```
<template>
  <div>
    <h1>API 資料</h1>
    <ul>
      <li v-for="item in items" :key="item.id">
        {{ item.ability.name }}
        {{ item.ability.url }}
      </li>
    </ul>
  </div>
</template>

<script setup>
import { ref, onMounted } from 'vue';
import axios from 'axios';

const items = ref([]);

onMounted(async () => {
  try {
    const response = await axios.get('https://pokeapi.co/api/v2/pokemon/pikachu');
    // console.log(response)
    items.value = response.data.abilities;
  } catch (error) {
    console.error("API 請求錯誤：", error);
  }
});
</script>
```

說明

1. 安裝 **Axios**：首先，我們需要安裝 Axios，這是一個流行的 HTTP 客戶端庫，用於在 Vue 應用中發送 HTTP 請求。
2. 創建 **Vue 組件**：我們創建了一個 Vue 組件，其中包含一個 <template> 部分用於展示從 API 獲取的資料，並使用 script setup 語法糖來編寫 JavaScript 部分。
3. 定義響應式資料：我們使用 ref 函數來定義一個響應式的資料 items，它將用於存儲從 API 獲取的資料。
4. 使用 **onMounted** 生命週期鉤子：在組件掛載後，我們使用 onMounted 函數來執行異步操作，這裡是發送 HTTP 請求。
5. 發送 HTTP 請求：使用 Axios 的 get 方法來發送 HTTP GET 請求到指定的 API URL。然後，將響應資料賦值給 items。
6. 錯誤處理：透過 try...catch 來捕獲並處理可能發生的錯誤。

透過這個簡單的範例，你可以看到在 Vue 3 中如何使用 Axios 進行 HTTP 請求和處理 API 響應。這樣的模式可以輕易地應用到更複雜的應用場景中，實現資料的動態加載和處理。