



Consumo de datos desde una API REST

Consumiendo una API con Axios

Implementar un aplicativo web que consume datos desde una API REST utilizando la librería Axios para dar solución a una problemática

- Unidad 1: Introducción a Componentes Web y Vue Js
- Unidad 2: Binding de formularios
- Unidad 3: Templates y rendering en Vue
- Unidad 4: Manejo de eventos y reutilización de componentes
- Unidad 5: Consumo de datos desde una API REST



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Implementa un componente Vue que consume datos de una API Rest utilizando la librería Axios para dar solución a un problema.*

¿Qué es fetch y
cómo se utiliza?



Consumo de API's con fetch

fetch

Nativamente JavaScript dispone del método fetch que nos permite consumir datos de un servidor.

```
fetch(<URL>)  
  .then( res => res.json())  
  .then( data => console.log(data))  
  .catch( error => console.log(error));
```

La ejecución de este método nos retorna una **promesa** que podemos manejar por medio de los métodos *then* y *catch*, accediendo a la respuesta positiva o negativa por medio de los **callbacks**, tal y como vimos en el módulo **Programación avanzada en JavaScript**.



Consumo de API's con fetch

fetch

Manejar las promesas de esta manera nos creará scopes internos debido a los callbacks que a la larga perjudica la rápida y fácil interpretación de la lógica.

```
fetch(<URL>)  
  .then( res => res.json())  
  .then( data => console.log(data))  
  .catch( error => console.log(error));
```



Consumo de API's con fetch

fetch

Por lo tanto optamos por usar una función **async/await** que junto con el **try catch** podemos obtener un código más claro y con menos scopes anidados provocados y manejado por callbacks.

```
async function getPokemons() {  
  try {  
    const res = await fetch("https://pokeapi.co/api/v2/pokemon");  
    const data = await res.json();  
    console.log(data);  
  } catch (error) {  
    console.log(error);  
  }  
}
```

Así como vimos en el módulo anterior, el **try catch** nos permite acceder al posible error en un bloque de código, similar a la función que cumple el método **.catch()**



Abre una pestaña nueva en el explorador y utiliza la consola para obtener y mostrar el valor del dólar usando fetch la api

<https://mindicador.cl/api>



/* Consumiendo una API con Axios */

Consumiendo una API

Axios

Usar fetch de forma pura y cruda no es la única manera que podemos optar para consumir datos de una API.

Existen varias librerías que nos ofrecen la misma funcionalidad pero con formatos diversos y casos de uso que pudiera ser más cómodo de incluir en nuestro código.

La librería más popular y usada en JavaScript para esto es **axios**.

Install


```
> npm i axios
```



Repository

 github.com/axios/axios

Homepage

 axios-http.com

Weekly Downloads

36.129.002



Consumiendo una API

Axios

Para agregar **axios** a nuestros proyectos solo debemos **abrir una terminal** ubicada en la carpeta de nuestra aplicación y **ejecutar** el siguiente comando:

```
npm install axios
```

Una vez instalado solo faltará **importarlo** en el componente que vayamos a usarlo justo nuestro del script.

```
import axios from "axios";
```

Ejercicio guiado



Consumo de API's con fetch

Axios

1. Crea una nueva aplicación Vue
2. Instala el paquete axios por npm dentro del proyecto
3. Limpia el componente App.vue
4. Importa axios en el componente App.vue
5. Agrega un método **getClimas()** con el siguiente código

```
async getClimas() {  
  try {  
    const url = "https://api.gael.cloud/general/public/clima";  
    const { data } = await axios.get(url);  
    return data;  
  } catch (error) {  
    console.log(error);  
  }  
},
```



Consumo de API's con fetch

Axios

6. Agrega el siguiente template

```
<template>
  <div>
    <h1>Climas</h1>
    <button @click="setClimas">Cargar climas</button>
    <ul>
      <li v-for="clima of climas">
        {{ clima.Estacion }} - {{ clima.Temp }}°
      </li>
    </ul>
  </div>
</template>
```



Consumo de API's con fetch

Axios

7. Agrega un arreglo en el estado llamado **climas**.

```
data() {  
  return {  
    climas: [],  
  };  
},
```



Consumo de API's con fetch

Axios

8. Agrega un método para asignar los climas de la API en el estado.

```
async setClimas() {  
  const climas = await this.getClimas();  
  this.climas = climas;  
},
```



Consumo de API's con fetch

Axios

Ahora prueba la aplicación presionando click en el botón y observa cómo se renderiza la lista con los datos de la API

Climas

Cargar climas

- Antofagasta - 17°
- Arica - 20°
- Balmaceda - 15°
- Calama - 22°
- Caldera - 20°
- Chillán - 20°
- Concepción - 17°
- Coyhaique - 16°
- Curicó - 24°
- Futaleufú - 16°
- Iquique - 19°
- Isla de Pascua - 23°
- La Serena/Coquimbo - 19°
- Los Ángeles - 20°
- Melinka - 12°



Ejercicio propuesto



Consumo de API's con fetch

axios

- Crea una aplicación Vue Js que cargue todos los indicadores de la API <https://mindicador.cl/api> en una tabla.

Cargar indicadores

| uf | ivp | dolar | dolar_intercambio | euro | ipc | utm | imacec | tpm | libra_cobre | tasa_desempleo | bitcoin |
|----------|----------|--------|-------------------|--------|-----|-------|--------|-------|-------------|----------------|----------|
| 34713.55 | 35382.72 | 893.54 | 758.87 | 906.87 | 0.5 | 60853 | -0.4 | 11.25 | 3.68 | 8.04 | 16990.26 |

¿Qué harías si necesitaras
consumir una API en el
momento que un usuario visita
tu aplicación?



¿Qué te ha parecido hasta
ahora Vue Js?





Próxima sesión...

- *Utiliza hooks del ciclo de vida de un componente para resolver un problema planteado*

{desafío}
latam_

*Academia de
talentos digitales*

