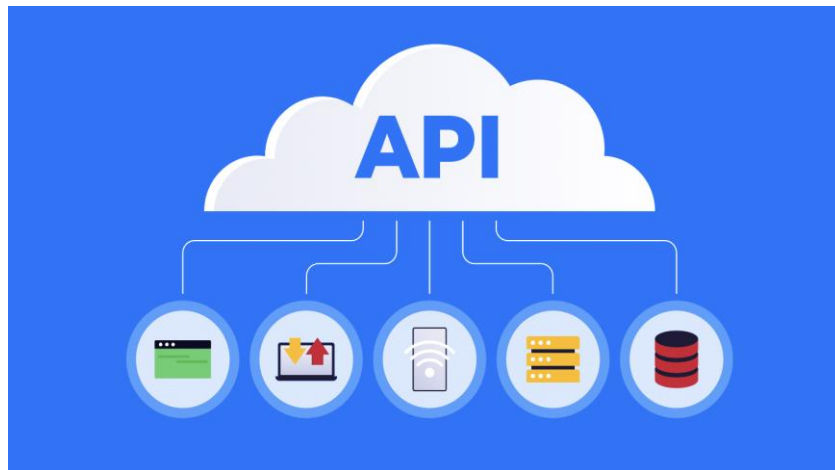


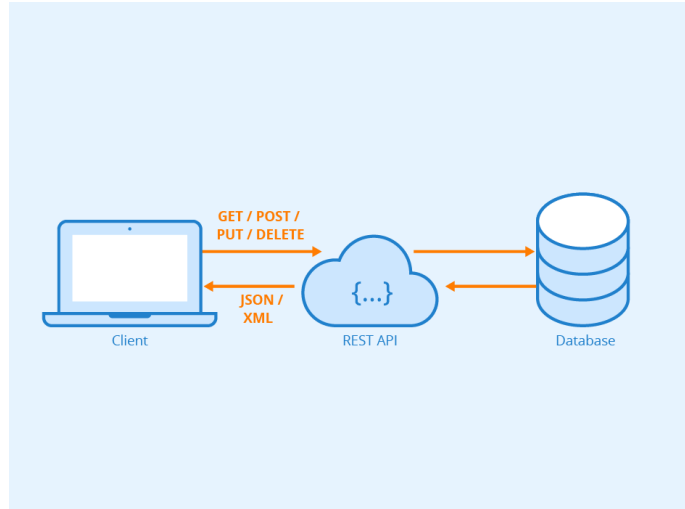
Consumo de API



¿Qué es una API?

Una API (**Application Programming Interface**, por sus siglas en inglés) es un conjunto de reglas y protocolos que se utilizan para permitir la comunicación entre diferentes aplicaciones o sistemas. En otras palabras, es un intermediario que permite que dos aplicaciones se comuniquen y compartan información entre sí.

Cuando hablamos de una API web, se trata de una interfaz que se ofrece a través de la web, que permite a los desarrolladores acceder a recursos y funcionalidades de un servicio en línea, como un sitio web o una aplicación, desde sus propias aplicaciones. De esta forma, pueden integrar y combinar diferentes servicios y funcionalidades en sus propias aplicaciones, sin tener que desarrollar todo desde cero.



Un ejemplo común de una API web es la API de Google Maps, que permite a los desarrolladores integrar mapas y funcionalidades de ubicación en sus propias aplicaciones, como aplicaciones de viajes, restaurantes, entre otras.

Las API también pueden ser utilizadas para automatizar procesos, compartir información entre aplicaciones y sistemas, y para crear aplicaciones personalizadas para dispositivos móviles y otros dispositivos.



En resumen, una API es una herramienta fundamental para la creación de aplicaciones modernas, ya que permite que los desarrolladores integren y utilicen servicios y funcionalidades de otras aplicaciones y servicios, de manera sencilla y eficiente.

¿Como la utilizamos?

1. Identificar la API que necesitas: Esto implica encontrar una API que tenga la funcionalidad que necesitas para tu proyecto. Hay muchas APIs disponibles en línea, tanto gratuitas como de pago, que ofrecen una amplia gama de funcionalidades, desde datos de ubicación hasta análisis de texto.
2. Obtener una clave API: La mayoría de las APIs requieren que los desarrolladores se registren y obtengan una clave API para poder utilizarlas. La clave API es un código único que te identifica como usuario autorizado de la API y te permite acceder a sus funcionalidades.
3. Conectar tu aplicación a la API: Una vez que hayas obtenido una clave API, necesitarás conectar tu aplicación a la API utilizando un lenguaje de programación como JavaScript, Python, Ruby, entre otros. Esto implica enviar solicitudes HTTP a la API y recibir respuestas de ella.
4. Analizar la respuesta de la API: Las respuestas de la API generalmente se envían en formato JSON o XML. Tendrás que analizar la respuesta para extraer los datos que necesitas y utilizarlos en tu aplicación.
5. Manejar errores y excepciones: Es importante que tu aplicación esté preparada para manejar errores y excepciones que puedan surgir al utilizar la API. Por ejemplo, si la API está temporalmente fuera de línea o si los datos que solicitas no están disponibles, tu aplicación debe ser capaz de manejar estos casos de manera adecuada.
6. En general, el proceso de utilizar una API puede ser un poco complejo al principio, pero una vez que comprendas cómo funciona, podrás utilizar una amplia variedad de APIs para crear aplicaciones interesantes y útiles.



@hdtoledo

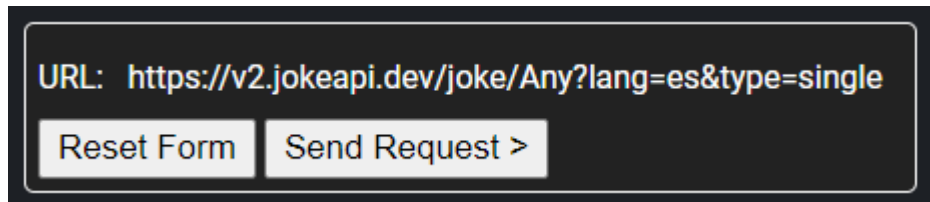
En este ejercicio utilizaremos una API de chistes que nos permitirá generar un chiste aleatorio, la pagina que trabajaremos será: <https://sv443.net/jokeapi/v2/?ref=apispublicas.com>



Esta API nos permite configurarla para obtener los datos que necesitamos en este caso haremos una pequeña configuración como lo es:

- Categoría: Any
- Idioma: español
- La respuesta será en formato JSON
- El tipo de broma será en single

Al final nos generara un enlace para hacer el consumo de esta.



Ahora empezaremos por darle la estructura HTML a nuestro proyecto:

```
<body>
  <div class="container">
    <h1>Chistes Ramdom</h1>
    <div class="emoticon">😂</div>
    <p class="broma deslizar">
      Presiona el boton "Reir" para hacerlo funcionar!
    </p>
    <button class="btn">Reir 🎲</button>
  </div>
  <script src="js/script.js"></script>
</body>
```

Aplicaremos los estilos CSS:

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;700&display=swap');

*{
  font-family: "Poppins", sans-serif;
}

body{
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  overflow-y: hidden;
}

.container{
  width: 500px;
  min-height: 200px;
  padding: 20px;
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  background-color: yellowgreen;
  border-radius: 20px;
  box-shadow: inset 0 0 40px rgba(0, 0, 0, 0.6);
}
```



@hdtoledo

```

  ✓ .emoticon{
    font-size: 80px;
  }

  ✓ .broma{
    text-align: center;
    color: black;
    font-weight: 500;
    opacity: 0;
  }

  ✓ .deslizar{
    opacity: 1;
    transition: opacity 1.5s;
  }

  ✓ button{
    padding: 10px 25px;
    margin-top: 20px;
    font-weight: bold;
    box-shadow: 5px 5px 0;
    cursor: pointer;
    position: relative;
  }

  ✓ button:active{
    position: relative;
    top: 5px;
    left: 5px;
    box-shadow: none;
  }

```



@hdtoledo

Y por último nuestro JS:

```
const btnReir = document.querySelector(".btn");
const txtDeslizar = document.querySelector(".broma");

var URL = "https://v2.jokeapi.dev/joke/Any?lang=es&type=single";

btnReir.addEventListener("click", getMethod);

async function getMethod(){
    txtDeslizar.classList.remove("deslizar");
    const data = await fetch(URL).then((e) => e.json());
    if(data){
        console.log(data);
        txtDeslizar.innerHTML = data.joke;
        txtDeslizar.classList.add("deslizar");
    }
}
```

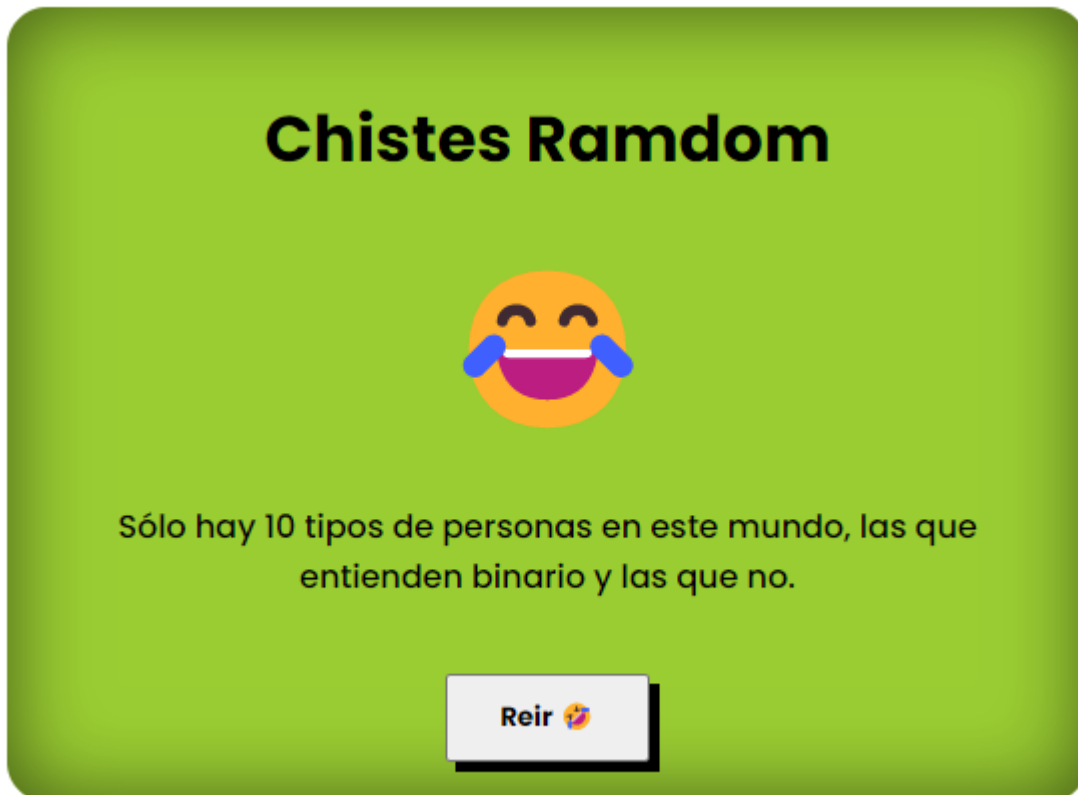
Entendamos el código:

1. **const btnReir = document.querySelector(".btn");** - En esta línea, estamos seleccionando el botón "Reir" de nuestro HTML usando la función **document.querySelector()** y asignando ese elemento a una variable llamada **btnReir**.
2. **const txtDeslizar = document.querySelector(".broma");** - De manera similar, en esta línea estamos seleccionando el elemento que contiene el texto del chiste que queremos mostrar y lo asignamos a la variable **txtDeslizar**.
3. **var URL = "https://v2.jokeapi.dev/joke/Any?lang=es&type=single";** - En esta línea, estamos almacenando la URL de la API que vamos a utilizar en una variable llamada **URL**. Esta API proporciona chistes al azar en español.
4. **btnReir.addEventListener("click", getMethod);** - Aquí estamos agregando un evento de escucha al botón "Reir". Cuando se hace clic en el botón, se llama a la función **getMethod()**.
5. **async function getMethod(){** - Esta es la función que se llama cuando se hace clic en el botón. Es una función asíncrona, lo que significa que utiliza el operador **await** para esperar a que se complete la solicitud a la API antes de continuar con el resto de la función.



@hdtoledo

6. **txtDeslizar.classList.remove("deslizar");** - Antes de mostrar el chiste, eliminamos la clase "deslizar" del elemento que contiene el texto del chiste. Esto asegura que el chiste se muestre sin animaciones previas.
7. **const data = await fetch(URL).then((e) => e.json());** - En esta línea, estamos haciendo una solicitud HTTP a la API utilizando la función **fetch()**. La solicitud se realiza a la URL almacenada en la variable **URL**. Utilizamos el método **.then()** para procesar la respuesta de la solicitud y convertirla a un objeto JSON.
8. **if(data){** - Verificamos si la respuesta de la API es exitosa y si hay datos disponibles para mostrar.
9. **console.log(data);** - Mostramos los datos obtenidos de la API en la consola para fines de depuración.
10. **txtDeslizar.innerHTML = data.joke;** - Aquí asignamos el chiste obtenido de la API al elemento HTML que contiene el texto del chiste.
11. **txtDeslizar.classList.add("deslizar");** - Finalmente, agregamos la clase "deslizar" al elemento que contiene el texto del chiste. Esto inicia una animación que muestra el chiste de manera suave y agradable.



@hdtoledo