

Que es una API

Las API de JavaScript son interfaces que permiten que un programa o aplicación se comunice con otros dispositivos o servicios usando JavaScript. En términos sencillos, una API es un intermediario que toma una solicitud de tu aplicación, la envía a un servidor o base de datos, y luego devuelve una respuesta.

Imagina que una API es como un camarero en un restaurante: toma tu pedido (solicitud), lo comunica a la cocina (el servidor) y te devuelve la comida (la respuesta). En este caso, el menú sería la documentación de la API, que describe qué solicitudes se pueden hacer y qué respuestas se pueden esperar.

Las API de JavaScript permiten a los desarrolladores interactuar con el contenido web de manera dinámica, modificando y actualizando datos en una página sin tener que recargarla.

Arquitectura REST

La mayoría de las API de JavaScript utilizan el estilo arquitectónico REST, conocido como API RESTful, que sigue el paradigma CRUD: Crear, Leer, Actualizar y Eliminar. Estas son las operaciones básicas para interactuar con servicios y bases de datos. En el contexto REST, estas acciones suelen tener alias específicos para su uso.

- **Creat** → **POST**
- **Read** → **GET**
- **Update** → **PUT**
- **Delete** → **DELETE**

Metodo	URL	Descripcion
POST	api/customer	Crear un nuevo cliente bancario
GET	api/customers/{id}	Recuperar la información de un cliente
PUT	api/customers/{id}	Actualizar la información de un cliente específico
DELETE	api/customers/{id}	Eliminar un cliente bancario

Las URL en las **API RESTful** son clave para acceder a un cliente o recurso específico.

- **POST** se usa para crear un nuevo cliente, proporcionando datos como nombre o detalles de tarjeta de crédito, y a veces genera un ID automáticamente.
- **GET** recupera la información de un cliente usando un ID único en la URL. La respuesta puede venir en formatos como JSON o XML.
- **PUT** actualiza la información de un cliente específico, sobrescribiendo los datos existentes. Al igual que con GET, se especifica un cliente mediante su ID y se pueden enviar nuevos datos en el cuerpo de la solicitud.
- **DELETE** puede eliminar un cliente por completo proporcionando una vez más el identificador.

Document Object Model (DOM) API

La API de Document Object Model (DOM) permite conectar páginas web con scripts en JavaScript, representando la estructura de un documento (como una página HTML) en la memoria. Esto facilita modificar dinámicamente el contenido de la página según sea necesario.

Ejemplo:

```
// Seleccionar un elemento del DOM y cambiar su contenido
document.getElementById('titulo').textContent = 'Nuevo Título';
```

XMLHttpRequest

Una API de JavaScript popular es XMLHttpRequest (XHR), que le permite recuperar datos sin actualizar toda la página. Esto es importante cuando desea actualizar solo una parte de una página sin interrumpir lo que un usuario está haciendo actualmente en la página.

XMLHttpRequest se usa mucho en la programación asíncrona de JavaScript y XML (AJAX). La documentación completa sobre su uso se puede encontrar en (esta página)

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>

APIs Avanzadas

Hay API de JavaScript más avanzadas disponibles, cada una con un uso y una especificación diferentes. Muchas de estas API se pueden encontrar en el [sitio web oficial de Mozilla Developer](#) o puede buscar en Internet la API requerida.

Conceptos Claves

Término	Definición
AJAX	"JavaScript y XML asíncronos" que abarca más que llamadas de servidor asíncronas a través de JavaScript y XML. No es un lenguaje de programación o tecnología, sino más bien un concepto de programación. Ajax representa una serie de técnicas que proporcionan aplicaciones web más ricas e interactivas a través de HTML, JavaScript, hojas de estilo en cascada y la modificación de la página web a través del modelo de objetos de documento. Sin embargo, el nombre es engañoso porque hoy en día, JSON se usa comúnmente en lugar de XML.
Funciones anónimas	Un tipo de función que no tiene nombre o podemos decir que no tiene ningún nombre. Se declara sin ningún identificador y a menudo se usa como parámetro de otra función. Es una forma común de ejecutar una función inmediatamente después de su declaración.
Clases	Las clases actúan como un modelo o plantilla para construir objetos con características y comportamientos similares. Una clase

	encapsula datos (en forma de propiedades) y funciones (en forma de métodos) que trabajan con esos datos.
Arreglo	Una estructura de datos que ayuda al programador en el almacenamiento y la recuperación de datos mediante claves indexadas. Las matrices utilizan un esquema de indexación basado en cero, lo que significa que el primer elemento de una matriz tiene un índice de cero. Las matrices crecen o se reducen dinámicamente agregando o eliminando elementos.
Script del lado del cliente	Un programa que acompaña a un documento HTML o incrustado en HTML. Los scripts se ejecutan durante la carga de un documento o cuando se realiza una acción. Se pueden utilizar para validar formularios, procesar entradas o crear dinámicamente elementos de documentos. Incruste un script en HTML, con la etiqueta <code><script></code> de una de las siguientes maneras: <code><script> // JS code</script></code> , <code><script src="path name"></script></code> . Utilice la etiqueta <code><noscript></code> para los navegadores con JavaScript deshabilitado o los que no admiten JavaScript.
Objetos de documento	Documento que representa la página web principal que da acceso a todos los elementos HTML de la página. Cuando se carga la página, el documento HTML se convierte en un documento. Se hace referencia a él con "documento".
Evento	Un evento es algo que hace un navegador o un usuario a lo que JavaScript puede reaccionar, como un clic en un botón o cuando un usuario envía una entrada en un formulario.
Enlace de eventos	Se refiere a decirle al navegador que se debe llamar a una función cada vez que ocurre algún 'evento'.
Controladores de eventos	Función que declara qué hacer cuando se realiza una acción, como hacer clic en un botón. Ejemplo: <code><button type="button" onclick="showAnswers()">Show Solution</button><script>function showAnswers() { // code alert("A")}</script></code> . Tenga en cuenta que <code>showAnswers()</code> es un controlador de eventos.
Extender	Esta palabra clave se utiliza en declaraciones de clase o expresiones de clase para crear una clase que sea un elemento secundario de otra clase.
Funciones	Las funciones son módulos de código que ejecutan una tarea en particular. Pueden tomar datos, llamados argumentos o parámetros, y a veces también devolver datos, llamados valor de retorno. Las funciones se definen comúnmente con esta sintaxis: <code>function functionName() { // function code }</code> .

IIFE	"Expresión de función invocada inmediatamente" se ejecuta inmediatamente después de definirse. Una vez que se ejecuta la función, no se puede volver a llamar en ninguna otra parte del programa. Es un tipo de función autoejecutable.
Nodos de texto	Los nodos que contienen texto real que van entre una etiqueta de inicio y una etiqueta de fin de un elemento.
this	La palabra clave "this" se refiere a la instancia actual del objeto. El valor de "this" puede variar dependiendo de cómo se llame al objeto.
Nodos	La base de todos los elementos de la estructura del Document Object Model (DOM).
Objetos	Los objetos son instancias creadas a partir de una clase. Son entidades del mundo real que representan las características definidas por la clase. Los objetos tienen un conjunto especial de propiedades que almacenan datos y métodos que especifican comportamientos.
Prototipos	<p>Un prototipo de función permite definir y agregar fácilmente propiedades o métodos a un objeto. Existen prototipos para todos los objetos que se pueden crear con la palabra clave "new". Todos los constructores de objetos crean objetos que heredan las propiedades y los métodos definidos por el prototipo. En la instanciación, los objetos heredan el estado actual del prototipo. Sin embargo, tenga en cuenta que los scripts pueden invalidar las propiedades y funciones de los prototipos. A continuación se muestra un ejemplo del uso de un prototipo para agregar un método a la clase Car:</p> <pre>function Car(make, model, year) { this.make = make; this.model = model; this.year = year; } Car.prototype.getName = function() { return this.make + ' ' + this.model + ' ' + this.year; }</pre>
Guión	Ofrece a los desarrolladores medios para modificar y ampliar documentos HTML de formas altamente interactivas. Los scripts se pueden activar por eventos como un clic en un botón o la entrada de datos, entre otros.
Funciones autoejecutables	A menudo se utiliza para inicializar datos o declarar elementos DOM. Estas funciones pueden ser anónimas.