## Breve historia de las API

Las API surgieron en los comienzos de la informática, mucho antes que la computadora personal. En esa época, una API normalmente se usaba como biblioteca para los sistemas operativos. Después de casi 30 años, las API se expandieron más allá de los entornos locales. A principios del año 2000, ya eran una tecnología importante para la integración remota de datos.

¿Qué es una API?

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero

NOTAS :Por ejemplo, el software del instituto de meteorología contiene datos meteorológicos. La aplicación meteorológica de su teléfono “habla” con este sistema a través de las API y le muestra las actualizaciones meteorológicas diarias en su teléfono. NOTAS

¿Qué significa API?

interfaz de programación de aplicaciones”.

NOTAS la palabra aplicación se refiere a cualquier software con una función distinta. La interfaz puede considerarse como un contrato de servicio entre dos aplicaciones. Este contrato define cómo se comunican entre sí mediante solicitudes y respuestas. NOTAS

¿Cómo funcionan las API?

La aplicación que envía la solicitud se llama cliente, y la que envía la respuesta se llama servidor.

NOTAS En el ejemplo del tiempo, la base de datos meteorológicos del instituto es el servidor y la aplicación móvil es el cliente. NOTAS

API de SOAP  simple object access protocol

En estas API. El cliente y el servidor intercambian mensajes mediante XML. Se trata de una API menos flexible que era más popular en el pasado.

Las API diseñadas con SOAP usan XML para el formato de sus mensajes y reciben solicitudes a través de HTTP. Con SOAP, es más fácil que las aplicaciones que funcionan en entornos distintos o están escritas en diferentes lenguajes compartan información.

API de WebSocket

utiliza objetos JSON para pasar datos. Admite la comunicación bidireccional entre las aplicaciones cliente y el servidor.

JSON

**JSON**

formato de datos basado en texto que sigue la sintaxis de objeto de JavaScript, puede ser utilizado independientemente de JavaScript, y muchos entornos de programación poseen la capacidad de leer y generar JSON.

NOTAS Los JSON son cadenas - útiles cuando se quiere transmitir datos a través de una red. Debe ser convertido a un objeto nativo de JavaScript cuando se requiera acceder a sus datos. NOTAS

Un objeto JSON puede ser almacenado en su propio archivo, que es básicamente sólo un archivo de texto con una extension .json

API de REST

NOTAS Estas son las API más populares y flexibles que se encuentran en la web actualmente. NOTAS

¿Qué son las API de REST?

NOTAS significa transferencia de estado representacional. NOTAS

El cliente envía las solicitudes al servidor como datos. El servidor utiliza esta entrada del cliente para iniciar funciones internas y devuelve los datos de salida al cliente.

REST define un conjunto de funciones como GET, PUT, DELETE, etc. que los clientes pueden utilizar para acceder a los datos del servidor. Los clientes y los servidores intercambian datos mediante HTTP.

¿Cuáles son los diferentes tipos de API?

API privadas

Estas son internas de una empresa y solo se utilizan para conectar sistemas y datos dentro de la empresa.

API públicas

Están abiertas al público y pueden cualquier persona puede utilizarlas. Puede haber o no alguna autorización y coste asociado a este tipo de API.

¿Cómo proteger una API?

NOTAS las API le permiten ofrecer acceso a los recursos y, al mismo tiempo, mantener la [seguridad](https://www.redhat.com/es/topics/security) y el control. Usted decide cómo habilita el acceso y a quiénes se lo otorga. NOTAS

1. Tokens de autenticación

NOTAS Se utilizan para autorizar a los usuarios a hacer la llamada a la API. NOTAS Los tokens de autenticación comprueban que los usuarios son quienes dicen ser y que tienen los derechos de acceso para esa llamada concreta a la API.

1. Claves de API

NOTAS Las claves de API verifican el programa o la aplicación que hace la llamada a la API NOTAS

Identifican la aplicación y se aseguran de que tiene los derechos de acceso necesarios para hacer la llamada a la API en cuestión. Las claves de API no son tan seguras como los tokens, pero permiten supervisar la API para recopilar datos sobre su uso.

NOTAS Es posible que haya notado una larga cadena de caracteres y números en la URL de su navegador cuando visita diferentes sitios web. Esta cadena es una clave de la API que el sitio web utiliza para hacer llamadas internas a la API. NOTAS

**HTTP REQUESTS**

# Métodos de petición HTTP

HTTP define un conjunto de **métodos de petición** para indicar la acción que se desea realizar para un recurso determinado.

[GET](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET)

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

[HEAD](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/HEAD)

El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

[POST](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST)

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

[PUT](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/PUT)

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

[DELETE](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/DELETE)

El método DELETE borra un recurso en específico.

**HTTP Request / Response**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **La solicitud HTTP** La línea de salida de una petición HTTP se conoce como línea de la petición. Siempre es la primera línea del mensaje de solicitud y contiene tres campos: • Un método HTTP • Un identificador universal de recursos (URI) • Una versión del protocolo HTTP  Aunque hay varios métodos de HTTP para recuperar datos de un servidor, las dos más utilizados son GET y POST.  El método GET solicita un recurso del servidor indicado en el campo URI. Si la URI apunta a una base de datos de producción de recursos como un servlet, los datos serán devueltos dentro del mensaje de respuesta.  El método POST se utiliza para pasar explícitamente datos al servidor en el propio mensaje de solicitud.Estos datos estarán a disposición del recurso URI especificado en la solicitud.  Como se mencionó anteriormente, la petición HTTP puede contener cero o más campos de cabecera. Estos campos de cabecera permiten pasar al servidor informacion del propio cliente y de la solicitud misma.  **La respuesta HTTP** Una vez que el servidor ha recibido y procesado la solicitud, éste debe devolver un mensaje de respuesta . El mensaje de respuesta se compone de una línea de estado y cero o más campos de cabecera, seguido por una línea vacía.  La primera línea del mensaje de respuesta HTTP que se conoce como la línea de estado, Se compone de HTTP versión del protocolo que se ajusta a la respuesta, seguida por un código numérico y su estatuto de texto explicación.  La siguiente tabla enumera los primeros dígitos y permite ver las categorías correspondientes   |  |  | | --- | --- | | Código de estado | Significado del valor numérico | | 100-199 | Informativo - La solicitud fue recibida y se está procesando. | | 200-299 | Éxito       - La acción fue recibida con éxito, entendido y aceptado. | | 300-399 | Redireción  - Otras peticiones hay que realizar para completar la petición. | | 400-499 | Error en Cliente - La solicitud contiene mala sintaxis y no pueden llevarse a cabo. | | 500-599 | Error de servidor - El servidor no pudo cumplir con una solicitud aparentemente válida. |   La siguiente tabla recoge algunas de las respuesta más común los códigos de estado.   |  | | --- | | Código Significado | | OK-200 La solicitud éxito | | 302 Movido temporalmente-La solicitud de residencia temporal en un URI. Si el nuevo URI es un ubicación, la ubicación en el campo de encabezado de respuesta dará a la nueva URL. Este código normalmente se utiliza cuando el cliente se redirige | | 400 Bad Request-El servidor no puede comprender la solicitud debido a un error de sintaxis. | | 401-La solicitud no autorizada requiere autenticación y / o autorización. | | 403 Forbidden-El servidor entiende la petición, pero por alguna razón se niega a cumplirlo. El servidor puede o no puede revelar la razón por la que se niega la petición. | | 404 Not Found-El servidor no ha encontrado nada que coinciden con el URI de solicitud. | | 500-Internal Server Error "El servidor encontró un inesperado condición que le impidió el cumplimiento de la petición. | |