**API**

1. **¿Qué es una API?**

API *App Programming Interface* (por sus siglas en inglés), es un programa de software de interfaz que ayuda a interactuar con otro software de usuario (UI). Es la puerta de entrada para que una aplicación incluya un software de terceros, facilitando el trabajo del desarrollador.

Las APIs proporcionan una forma de interacción de una aplicación con otra que necesite las funciones de esta. Por ejemplo, una API puede solicitar datos a otra y obtener estos datos para poder utilizarlos en el programa. También se pueden compartir funcionalidades de una API a otra. World Wide Web Consortium la define como “un conjunto de definiciones de subrutinas, protocolos y servicios para construir aplicaciones de software”.

1. **Tipos de APIs**
   1. *APIs basadas en XML*

También se conocen como APIs de servicios web y APIs REST (Transferencia de Estado Representacional). Su diferencia entre las XML y las de servicios web es la sintaxis, básicamente ambas son lo mismo. Mientras que las APIs de XML son compatibles con navegadores web Y Windows Phone, las APIs REST son un tipo de API moderna que utiliza menos ancho de banda.

* 1. *APIs SOAP*

Son APIs mas antiguas, utilizan la misma sintaxis de XML cargando y descargando los servicios web con HTTP como protocolo.

Estas APIs hacer llamadas a procedimientos remotos. Estas llamadas utilizan SOAP sobre HTTP para la comunicación entre el cliente y el servidor con una clave API. Incluye puntos finales para que el cliente envíe, reciba y pase comandos entre sí.

1. **¿Cómo se integra una API?**

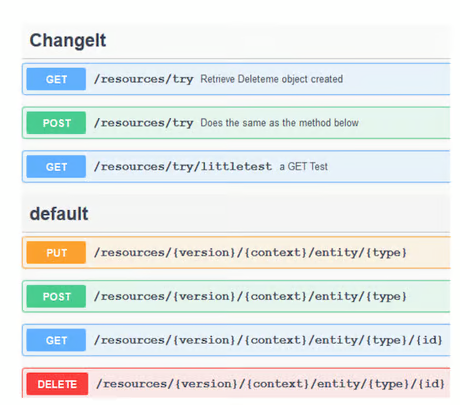
* Lo primero que se debe hacer es entender que funciones tiene la API, leer la documentación y comprender los puntos finales, métodos de solicitud, autenticación y cualquier otro parámetro necesario.
* Luego, obtener las credenciales de acceso de ser necesario, varias APIs suelen tener claves o tokens de acceso para autorizar las solicitudes de la API.
* Implementar bibliotecas o herramientas para procesar las solicitudes HTTP y procesar respuestas.
* Realizar pruebas exhaustivas para asegurar la que integración funciona correctamente en diversos escenarios.
* Implementar manejos de errores, optimización y monitoreo para agilitar el rendimiento de solicitudes y supervisar la integración.

1. **Inicios de una API**

La forma más fácil para empezar con una API es utilizar un servicio online. Existen servicios gratis y de pago mínimo para poder probar y desarrollar la interfaz de programación. Hay claves para estos productos, una vez que hagamos el proceso para acceder a la API que necesitamos implementar, el servicio web nos dará una clave de acceso de uso único.

1. **Beneficios**
2. *Interoperabilidad:* Las APIs permiten la comunicación entre diferentes sistemas y plataformas de manera eficaz. Esto permite que las aplicaciones compartan datos y funcionalidades sin depender del mismo entorno.
3. *Reutilización:*  En lugar de recrear una funcionalidad específica, se puede utilizar una API existente acelerando el proceso de desarrollo y reduciendo la duplicación de esfuerzos.
4. *Escalabilidad:* Si se necesita agregar nuevas características o servicios se pueden utilizar las APIs en vez de reconstruir las aplicaciones de cero.
5. *Acceso a Datos Externos:* Se puede integrar información de terceros, como datos de redes sociales, mapas, pronósticos meteorológicos, etc. Todo en la aplicación sin necesidad de gestionar los datos directamente.
6. *Comunidad y Ecosistema:* Dado que las APIs son consumidas por varios usuarios a la vez, se puede generar colaboración, intercambios de ideas y mantenimiento conjunto.
7. **Puntos finales**

Son métodos específicos que el usuario puede utilizar para interactuar con su aplicación y extraer las funciones, datos e información que se requiere.



Los puntos finales de una API son las URL específicas a las que se conecta para interactuar con diferentes funciones o servicios de la API. Generalmente se proporciona información detallada de como utilizar el punto final, incluyendo parámetros y respuestas esperadas.

1. **Comandos**
   1. *GET:* Se utiliza para recuperar datos de una URL. También se utiliza para obtener el contenido completo de una página HTML incluida en una API.
   2. *POST:* Enviar datos de vuelta al servidor y guardarlos. Crea una nueva entrada en la API.
   3. *PUT:* Se utiliza para actualizar los datos en una base de datos concreta. El servidor envía mensajes de confirmación si es que los datos se han cambiado o no.
   4. *DELETE:* Este método se utiliza para eliminar una entrada en la base de datos. Para la petición se llama a la URL y se busca la entrada que se quiere eliminar.
2. **Conclusión:**

Las APIs pueden ser muy beneficiosas en el momento de ahorrar líneas de código y tiempo. Ofrece una forma eficiente para que diferentes sistemas y apps se comuniquen entre sí. Los beneficios de utilizar una API son diversos, incluyendo la interoperabilidad, reutilización, desarrollo rápido y acceso a datos externos, entre otros. Integrar una API en el sistema también refiere un trabajo entorno a comprender los puntos finales, que URL especificas tiene y que funciones y acciones se pueden utilizar. Este procedimiento, aunque pueda resultar tedioso es más eficaz que reescribir un código completo que ya esta implementado en otro lugar.

1. **Bibliografía**

Appmaster. (2020, 10 de septiembre). APIs para principiantes: cómo utilizar una API (una guía completa). Recuperado el 25 de febrero de 2024 de <https://appmaster.io/es/blog/apis-para-principiantes-como-utilizar-una-api-una-guia-completa>

‌