Engineer

API

🔀 la forma descrita en que un programa llama a otro

Qué es

Una API, abreviatura de interfaz de programación de aplicaciones, es el conjunto publicado de operaciones que un programa ofrece a otros programas: cómo se llama cada operación, qué acepta, qué devuelve y cómo comunica los problemas. En español escrito, la sigla no cambia en plural: las API. Quien la invoca conoce esa superficie y nada de lo que hay detrás, de modo que el programa al otro lado puede reescribirse, trasladarse a otra máquina o dividirse en tres, y quien lo llama sigue funcionando mientras se mantenga esa superficie.

La palabra abarca varias capas. La API de una biblioteca son las funciones y los tipos que exporta, a los que se llama dentro de un mismo proceso. La API de un sistema operativo es su conjunto de llamadas al sistema, y POSIX estandariza un núcleo común para que un programa escrito con `open` y `read` pueda compilarse para más de un kernel de tipo Unix. Una API web se ofrece a través de una red, casi siempre sobre HTTP, y ese es hoy el sentido habitual del término sin más: un servidor que responde a solicitudes en direcciones como `/v1/orders`, normalmente con JSON en el cuerpo.

Las API web adoptan unos pocos estilos. Las orientadas a recursos, a menudo llamadas de estilo REST, asignan una dirección a cada cosa y usan los métodos de HTTP para leerla y modificarla. Las de estilo RPC, como gRPC, publican procedimientos con nombre cuyos argumentos y resultados se declaran de antemano. GraphQL ofrece un endpoint y un sistema de tipos, y el cliente escribe una consulta que nombra exactamente los campos que quiere recibir. Los webhooks invierten la dirección: cuando ocurre algo, el proveedor llama a una dirección que el cliente había registrado.

Se confunden con ella tres términos cercanos. Un protocolo, como HTTP, define cómo viajan los mensajes; una API es lo que un servicio concreto dice mediante ese protocolo. Un SDK es una biblioteca escrita en un lenguaje de programación que envuelve la API de alguien, para que quien desarrolla llame a una función en vez de ensamblar solicitudes. La implementación, por su parte, es el código detrás de la API, que quien la invoca nunca necesita leer.

Utilice el concepto en una decisión

Aplique «¿Qué es una API?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. La parte que el usuario no ve: dónde se guardan los datos, qué reglas se aplican y cómo pueden comunicarse otros sistemas con ella.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Diseño y operación de 6 flujos ETL/ELT — 6 flujos ETL en tres motores de bases de datos

Diseño y operación de 6 flujos ETL/ELT Póngase en contacto →

Por qué importa

Una API es la parte de un sistema sobre la que otras personas construyen, lo que la convierte en una promesa que sobrevive al código que tiene detrás. Reescribir un manejador lleva una tarde. Cambiar lo que devuelve puede romper una app ya instalada en los teléfonos de los clientes, una importación nocturna de un socio o un script que nadie recuerda haber escrito.

La primera decisión es el público. Una API privada usada solo por el frontend del equipo puede cambiar en la misma pull request que su único cliente. Una pública cambia mediante avisos y retirada gradual; cada campo entregado deberá mantenerse mientras alguien lo lea.

La segunda decisión es el rigor del contrato. Una API estricta declara un tipo y un límite para cada entrada, rechaza lo que no reconoce y responde a cada fallo con un código de estado preciso. Detecta los errores en la entrada y resulta más fácil de documentar, probar y usar para generar clientes. Una API tolerante acepta lo que llega y hace lo que puede: parece amable el primer mes, pero después convierte cada accidente de la implementación en un comportamiento del que alguien depende. El rigor cuesta tiempo de diseño al principio; la tolerancia cuesta más después, en un momento que nadie elige.

La tercera decisión se refiere a las API ajenas. Un proveedor de pagos, un servicio de mapas o un modelo de lenguaje constituye una API que el producto invoca y no controla: cambian los precios, se retiran versiones y se reducen las cuotas. Llamar directamente a un proveedor desde cincuenta lugares convierte sustituirlo en una reescritura. Llamarlo desde un módulo que controla el propio equipo convierte la sustitución en un cambio de ese módulo. La envoltura cuesta un poco de indirección y permite disponer de una salida.

Semantic Versioning da vocabulario a esa promesa: el número de versión mayor aumenta cuando la API pública cambia de forma incompatible, de modo que un cliente puede saber solo por la versión si es seguro actualizar. Sobre HTTP hay varios lugares en los que señalar una ruptura de compatibilidad, y RFC 9205, la guía del IETF para protocolos construidos sobre HTTP, enumera entre ellos una nueva relación de enlace, un nuevo tipo de medio o un nuevo campo de cabecera.

Cómo se aplica en la práctica

Las API de varios proyectos del sitio se diseñaron alrededor de una pregunta: qué va en la API y qué va en otro lugar.

En NextMariner, una plataforma marítima de contactos profesionales, reseñas de empresas y educación patrocinada, la contribución de la fundadora a la arquitectura incluyó mantener la API Go deliberadamente ligera. Construida con Huma sobre Fiber, enruta, valida la solicitud, aplica el filtrado de seguridad y orquesta, sin contener ninguna lógica de negocio. Las reglas residen en funciones PostgreSQL, por lo que cambiar una regla ocurre en una sola capa, en SQL. Los límites se documentaron y se hicieron cumplir en la revisión. Eso convierte a la API en un adaptador aburrido, en el buen sentido.

Un proyecto relacionado también sacó de la API la tarea de dar forma a los datos. Las funciones PostgreSQL ensamblan cada respuesta como JSON, y el manejador Go la recibe como `json.RawMessage` y la reenvía intacta mediante un pequeño auxiliar `QueryJSON`. Desaparecieron los DTO, los mapeadores y las estructuras de respuesta que suele acumular una API por capas; los manejadores se acortaron drásticamente y el frontend dejó de depender de Go. La contrapartida merece nombrarse: la forma de las respuestas de la API se define ahora en SQL, de modo que es allí donde quien revisa lee el contrato.

Una API accesible a cualquiera debe defenderse. Un directorio público de empresas y profesionales está tras 22 limitadores de frecuencia, cada uno diseñado para su ruta, con estado en Redis para aplicar el límite entre instancias. Si Redis no responde, deniegan el tráfico. Completan la defensa Cloudflare Turnstile, una restricción de origen, un desafío propio del directorio y 351 líneas de middleware de idempotencia para no duplicar una escritura reintentada. La página declara el coste: un problema de Redis se vuelve visible para el usuario, preferible a convertirlo en uno de facturación.

Dónde surgen los problemas

Los fallos suelen aparecer tarde, cuando ya han crecido clientes y datos.

Renombrar un campo rompe un cliente desconocido. Al publicar, alguien depende de cada detalle observable: nombres, orden de listas y texto de errores. Cambiar `customer_name` por `name` rompe la integración de un socio en la siguiente solicitud. Cambie añadiendo: ponga el campo nuevo junto al antiguo, declare obsoleto este, observe los registros hasta que nadie lo lea y retírelo solo en otra versión mayor.

Un endpoint de lista se ralentiza cada mes. Devuelve todas las filas coincidentes, algo que funcionaba con cincuenta registros en el lanzamiento y que agota el tiempo de espera con doscientos mil. Incluya paginación desde la primera versión, con un tamaño de página predeterminado y un máximo estricto que el cliente no pueda superar. Para colecciones grandes o con mucha actividad, prefiera un cursor a un desplazamiento: con este último, una fila insertada entre dos solicitudes empuja otra fila a la página siguiente, donde el cliente la lee por segunda vez.

```
GET /v1/orders?limit=50&after=ord_8f2c HTTP/1.1
Host: api.example.com
```

Un cliente lee el pedido de otro. La API comprueba la sesión y carga `/v1/orders/42` solo por identificador. Cambiar el número recorre datos ajenos. La autenticación respondió quién llama; nada preguntó si le pertenece el pedido 42. Limite cada búsqueda por propietario dentro de la consulta y mantenga una prueba que inicia sesión con una cuenta y pide registros de otra.

La respuesta expone columnas nunca destinadas a publicarse. Serializar directamente una fila o un objeto ORM envía hashes de contraseña, indicadores internos o costes de proveedor a cada cliente. Una columna añadida el próximo año se publica al crearla. En entrada, el mismo error permite fijar `role` o `is_admin` incluyéndolos en el cuerpo. Dé tipos de entrada y salida explícitos con los campos permitidos para convertir cada publicación en una decisión.

Lecturas adicionales

- RFC 9110: HTTP Semantics — los métodos, códigos de estado y cabeceras de los que se construye toda API web, definidos por el IETF.
- RFC 9205: Building Protocols with HTTP — prácticas recomendadas actuales para diseñar una aplicación sobre HTTP, incluida su evolución.
- RFC 9457: Problem Details for HTTP APIs — una estructura JSON estándar para respuestas de error, de modo que los clientes analicen todos los fallos de la misma manera.
- RFC 6585: Additional HTTP Status Codes — donde se definen 429 Too Many Requests y su indicación `Retry-After` para clientes sujetos a límites de frecuencia.
- Semantic Versioning 2.0.0 — las reglas para señalar un cambio incompatible de API mediante un número de versión.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es una API?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/api/