Engineer

API

🔀 die beschriebene Art, wie ein Programm ein anderes aufruft

Was es ist

Eine API, kurz für Application Programming Interface, ist die veröffentlichte Menge von Operationen, die Software anderer Software anbietet: wie jede Operation heißt, was sie annimmt, was sie zurückgibt und wie sie Probleme meldet. Der Plural lautet APIs. Der Aufrufer kennt diese Oberfläche und nichts dahinter. Das Programm auf der anderen Seite kann deshalb neu geschrieben, auf einen anderen Rechner verschoben oder in drei Teile zerlegt werden. Solange die Oberfläche bestehen bleibt, funktioniert der Aufrufer weiter.

Der Begriff umfasst mehrere Ebenen. Die API einer Bibliothek besteht aus exportierten Funktionen und Typen, die innerhalb eines Prozesses aufgerufen werden. Die API eines Betriebssystems ist seine Menge von Systemaufrufen. POSIX standardisiert einen gemeinsamen Kern, damit ein Programm mit `open` und `read` auf mehreren Unix-artigen Kerneln gebaut werden kann. Eine Web-API wird über ein Netzwerk angeboten, fast immer über HTTP. Das ist heute meist gemeint: ein Server, der Anfragen an Adressen wie `/v1/orders` beantwortet, üblicherweise mit JSON im Nachrichtentext.

Web-APIs haben mehrere Formen. Ressourcenorientierte, häufig REST-artig genannt, geben jedem Gegenstand eine Adresse und verwenden HTTP-Methoden zum Lesen und Ändern. RPC-artige APIs wie gRPC veröffentlichen benannte Prozeduren mit vorab deklarierten Argumenten und Ergebnissen. GraphQL bietet einen Endpunkt und ein Typsystem. Der Client schreibt eine Abfrage, die genau die gewünschten Felder benennt. Webhooks kehren die Richtung um: Bei einem Ereignis ruft der Anbieter eine vom Kunden registrierte Adresse auf.

Drei benachbarte Begriffe werden verwechselt. Ein Protokoll wie HTTP bestimmt den Transport der Nachrichten; eine API legt fest, was ein bestimmter Dienst darin sagt. Ein SDK ist eine Bibliothek in einer Programmiersprache, die eine API umhüllt, sodass Entwickler eine Funktion aufrufen, statt Anfragen zusammenzubauen. Die Implementierung ist schließlich der Code hinter der API, den der Aufrufer nie lesen muss.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine API?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Der Teil, den Nutzer nie sehen: wo die Daten liegen, welche Regeln gelten und wie andere Systeme darauf zugreifen dürfen.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 6 ETL/ELT-Pipelines entworfen und betrieben — 6 ETL-Pipelines über drei Datenbankengines hinweg

6 ETL/ELT-Pipelines entworfen und betrieben Kontakt aufnehmen →

Warum es wichtig ist

Eine API ist der Teil eines Systems, auf dem andere aufbauen. Damit ist sie eine Zusage, die ihren Code überdauert. Einen Handler umzuschreiben dauert einen Nachmittag. Seine Rückgabe zu ändern kann eine bereits auf Kundentelefonen installierte App, den nächtlichen Import eines Partners oder ein vergessenes Skript beschädigen.

Die erste Entscheidung betrifft deshalb die Nutzer. Eine private API nur für das eigene Frontend kann im selben Pull Request wie ihr einziger Client geändert werden. Eine öffentliche API verändert sich durch Abkündigung und Vorankündigung. Jedes ausgelieferte Feld bleibt zu unterstützen, solange jemand es liest.

Die zweite Entscheidung ist die Strenge des Vertrags. Eine strenge API deklariert für jede Eingabe Typ und Grenze, weist Unbekanntes zurück und beantwortet jeden Fehler mit einem genauen Statuscode. Sie fängt Fehler am Eingang ab und lässt sich leichter dokumentieren, testen und zur Clientgenerierung nutzen. Eine tolerante API nimmt alles an und versucht das Beste. Das wirkt im ersten Monat freundlich und macht später jede zufällige Eigenheit der Implementierung zu einem Verhalten, auf das jemand baut. Strenge kostet am Anfang Entwurfszeit; Toleranz kostet später mehr, zu einem Zeitpunkt, den niemand auswählt.

Die dritte Entscheidung betrifft fremde APIs. Ein Zahlungsanbieter, Kartendienst oder Sprachmodell wird vom Produkt aufgerufen, aber nicht kontrolliert. Preise ändern sich, Versionen werden eingestellt, Kontingente enger. Direkte Aufrufe eines Anbieters an fünfzig Stellen machen den Wechsel zur Neuentwicklung. Ein Aufruf aus einem eigenen Modul beschränkt ihn auf dieses Modul. Der Wrapper kostet etwas Indirektion und schafft einen Ausweg.

Semantic Versioning liefert das Vokabular für diese Zusage: Bei einer inkompatiblen Änderung der öffentlichen API steigt die Hauptversionsnummer. Ein Client kann deshalb allein an der Version erkennen, ob ein Upgrade verträglich ist. Über HTTP lässt sich ein Bruch an mehreren Stellen signalisieren. RFC 9205, die IETF-Empfehlung für HTTP-basierte Protokolle, nennt unter anderem eine neue Linkrelation, einen neuen Medientyp oder ein neues Headerfeld.

Wie es in der Praxis aussieht

Die APIs mehrerer Projekte auf dieser Website wurden von einer Frage geprägt: Was gehört in die API und was anderswohin?

NextMariner ist eine maritime Plattform für berufliche Vernetzung, Bewertungen von Unternehmen und geförderte Weiterbildung. Zum Architekturbeitrag der Gründerin gehörte es, die Go-API bewusst schlank zu halten. Mit Huma auf Fiber übernimmt sie Routing, Eingabevalidierung, Sicherheitsfilterung und Ablaufsteuerung, enthält aber keinerlei Geschäftslogik. Die Regeln liegen in PostgreSQL-Funktionen, sodass eine Regeländerung in einer einzigen Schicht in SQL geschieht. Die Grenzen wurden aufgeschrieben und in Reviews durchgesetzt. Damit ist die API ein langweiliger Adapter, im guten Sinn.

Ein verwandtes Projekt nahm auch die Formung der Antworten aus der API. PostgreSQL-Funktionen setzen jede Antwort als JSON zusammen. Der Go-Handler erhält sie als `json.RawMessage` und leitet sie unverändert über einen kleinen `QueryJSON`-Helfer weiter. DTOs, Mapper und Antwortstrukturen, die eine geschichtete API sonst ansammelt, entfielen. Handler wurden erheblich kürzer, und das Frontend hing nicht länger von Go ab. Der Tausch verdient eine klare Benennung: Die Form der API-Antworten steht nun in SQL. Dort liest ein Reviewer den Vertrag.

Eine öffentlich erreichbare API muss sich auch verteidigen. Vor einem öffentlichen Verzeichnis von Firmen und Fachleuten stehen 22 jeweils auf ihren Pfad zugeschnittene Ratenbegrenzer. Ihr Zustand liegt in Redis, damit Grenzen über Instanzen hinweg gelten. Ist Redis unerreichbar, verweigern sie den Verkehr. Cloudflare Turnstile, eine Ursprungssperre, eine eigene Challenge am Verzeichnis und 351 Zeilen Idempotenz-Middleware gegen doppelt ankommende Schreibwiederholungen vervollständigen den Schutz. Die Seite benennt den Preis offen: Ein Redis-Problem wird für Nutzer sichtbar. Das wurde einem Redis-Problem vorgezogen, das zum Abrechnungsproblem wird.

Wo es schiefgeht

API-Fehler kommen meist spät, wenn Clients und Daten gewachsen sind.

Ein umbenanntes Feld bricht einen unbekannten Client. Nach der Veröffentlichung hängt jemand an jedem beobachtbaren Detail: Feldnamen, Listenreihenfolge und Fehlertext. Aus `customer_name` wird `name`, und bei der nächsten Anfrage bricht die Partnerintegration. Ändern Sie stattdessen durch Ergänzung: Fügen Sie das neue Feld neben dem alten hinzu, markieren Sie das alte als abgekündigt und beobachten Sie die Protokolle, bis es niemand mehr liest. Entfernen Sie es erst in einer neuen Hauptversion.

Ein Listenendpunkt wird jeden Monat langsamer. Er liefert alle passenden Zeilen. Bei fünfzig Datensätzen zur Veröffentlichung ging das gut, bei zweihunderttausend läuft er ins Zeitlimit. Paginieren Sie ab der ersten Version, mit Standardgröße und einer harten Obergrenze für Clients. Für große oder lebhafte Sammlungen eignet sich ein Cursor besser als ein Offset. Beim Offset schiebt eine zwischen zwei Anfragen eingefügte Zeile eine andere auf die Folgeseite, wo der Client sie erneut liest.

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

Ein Kunde liest den Auftrag eines anderen. Die API prüft die Anmeldung und lädt `/v1/orders/42` allein über die Kennung. Eine andere Zahl in der Adresse führt durch fremde Daten. Die Authentifizierung beantwortete, wer anruft. Niemand fragte, ob dieser Aufrufer Auftrag 42 besitzt. Begrenzen Sie jede Suche bereits in der Abfrage auf den Eigentümer und behalten Sie einen Test, der als ein Konto Datensätze eines anderen anfordert.

Die Antwort enthält nie zur Veröffentlichung bestimmte Spalten. Ein Endpunkt serialisiert eine Datenbankzeile oder ein ORM-Objekt unverändert. Passwort-Hash, internes Flag oder der Einkaufspreis eines Lieferanten gehen damit an jeden Client. Eine nächstes Jahr ergänzte Spalte wird am Tag ihrer Anlage veröffentlicht. Derselbe Fehler auf der Eingabeseite erlaubt, `role` oder `is_admin` einfach im Nachrichtentext zu setzen. Geben Sie jedem Endpunkt ausdrückliche Ein- und Ausgabetypen mit erlaubten Feldern. So wird die Veröffentlichung eines Feldes zu einer bewussten Entscheidung.

Weiterführende Lektüre

- RFC 9110: HTTP Semantics — die vom IETF definierten Methoden, Statuscodes und Header als Grundlage jeder Web-API.
- RFC 9205: Building Protocols with HTTP — aktuelle Empfehlungen für Anwendungen auf HTTP, einschließlich ihrer Weiterentwicklung.
- RFC 9457: Problem Details for HTTP APIs — eine standardisierte JSON-Form für Fehlermeldungen, damit Clients Fehler einheitlich auswerten.
- RFC 6585: Additional HTTP Status Codes — die Definition von 429 Too Many Requests und dem `Retry-After` -Hinweis für begrenzte Clients.
- Semantic Versioning 2.0.0 — die Regeln, um inkompatible API-Änderungen über Versionsnummern zu signalisieren.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist eine API?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/api/