Engineer

API

🗘 den beskrevne måde et program kalder et andet på

Hvad det er

Et API, en forkortelse af application programming interface, er de offentliggjorte operationer, et stykke software stiller til rådighed for anden software: hvad hver operation hedder, hvad den tager imod, hvad den svarer, og hvordan den melder om problemer. I flertal skrives det API'er eller APIer. Den, der kalder, kender overfladen og intet bag den, så programmet på den anden side kan skrives om, flyttes til en anden maskine eller deles i tre, mens kalderen bliver ved med at virke, så længe overfladen holder.

Ordet dækker flere lag. Et biblioteks API er de funktioner og typer, biblioteket eksporterer, og de kaldes inden for én proces. Et styresystems API er dets systemkald, og POSIX standardiserer et fælles sæt af dem, så et program skrevet mod `open` og `read` kan bygges til mere end ét Unix-lignende system. Et web-API udbydes over et netværk, næsten altid over HTTP, og det er det, de fleste mener med ordet i dag: en server, der svarer på forespørgsler til adresser som `/v1/orders`, typisk med JSON som indhold.

Web-API'er findes i en håndfuld stilarter. De ressourceorienterede, ofte kaldt REST-agtige, giver hver ting sin egen adresse og bruger HTTP's metoder til at læse og ændre den. De RPC-agtige, som gRPC, udstiller navngivne procedurer med argumenter og resultater erklæret på forhånd. GraphQL har ét endpoint og et skema, og klienten skriver en forespørgsel, der nævner præcis de felter, den vil have tilbage. Webhooks vender retningen om: udbyderen kalder en adresse, kunden har registreret, når der sker noget.

Tre nabobegreber bliver ofte blandet sammen med det. En protokol som HTTP bestemmer, hvordan beskeder bevæger sig; et API er det, en bestemt tjeneste siger i den protokol. Et SDK er et bibliotek i ét programmeringssprog, der pakker

en andens API ind, så en udvikler kalder en funktion i stedet for at bygge forespørgsler selv. Og implementeringen er koden bag API'et, som den, der kalder, slet ikke behøver at læse.

Brug begrebet i en beslutning

Brug »Hvad er et API?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Den del, brugerne aldrig ser: hvor data ligger, hvilke regler der gælder, og hvordan andre systemer må tale med den.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har designet og drevet 6 ETL/ELT-pipelines — 6 ETL-pipelines på tre databasemotorer

Har designet og drevet 6 ETL/ELT-pipelines Kontakt os →

Hvorfor det betyder noget

Et API er den del af et system, som andre bygger videre på, og derfor er det et løfte, der lever længere end koden bag det. At skrive en handler om tager en eftermiddag. At ændre det, handleren svarer, kan knække en app, der allerede ligger på kundernes telefoner, en partners natlige import eller et script, ingen længere husker at have skrevet.

Den første beslutning handler derfor om publikum. Et privat API, som kun teamets egen frontend bruger, kan ændres i samme pull request som sin ene klient. Et offentligt API ændres gennem udfasning og varsel, og hvert felt, det sender ud, er et felt, der skal understøttes, så længe nogen læser det.

Den anden beslutning er, hvor stram kontrakten skal være. Et stramt API erklærer type og grænse for hvert input, afviser det, det ikke kender, og svarer på hver fejl med en præcis statuskode. Det fanger fejl ved døren og er lettere at dokumentere, teste og generere klienter til. Et eftergivende API tager imod hvad som helst og gør sit bedste. Det føles venligt den første måned og gør siden hvert tilfældigt træk ved implementeringen til adfærd, som nogen er afhængig af. Stramhed koster designtid nu; eftergivenhed koster mere senere, på et tidspunkt ingen selv vælger.

Den tredje beslutning gælder andres API'er. En betalingsudbyder, en korttjeneste eller en sprogmodel er et API, produktet kalder uden at styre det: priserne flytter sig, versioner udfases, kvoter strammes. Kaldes leverandøren direkte fra

halvtreds steder, bliver et skifte til en omskrivning. Kaldes den fra ét modul, teamet selv ejer, bliver skiftet en ændring i det modul. Indpakningen koster lidt indirektion og køber en udvej.

Semantic Versioning giver løftet et ordforråd: hovednummeret stiger, når det offentlige API ændres, så det bryder med tidligere versioner, og en klient kan derfor se på versionen alene, om en opgradering er sikker. Over HTTP er der flere steder at markere et brud, og RFC 9205, IETF's vejledning til protokoller bygget på HTTP, nævner blandt andet en ny linkrelation, en ny medietype og et nyt headerfelt.

Sådan ser det ud i praksis

API'erne bag flere af projekterne på dette site er formet af ét spørgsmål: hvad hører hjemme i API'et, og hvad hører hjemme et andet sted?

På NextMariner, en maritim platform til professionelt netværk, virksomhedsanmeldelser og sponsoreret uddannelse, omfattede grundlæggerens arkitekturbidrag at holde Go-API'et bevidst tyndt. Det er bygget med Huma oven på Fiber, og det router, validerer forespørgslen, anvender sikkerhedsfiltreringen og orkestrerer, men rummer ingen forretningslogik overhovedet. Reglerne bor i PostgreSQL-funktioner, så en regelændring sker i ét lag, i SQL. Grænserne blev skrevet ned og håndhævet i review. Det gør API'et til en kedelig adapter i den gode betydning af ordet.

Et beslægtet projekt tog også formningen af svarene ud af API'et. PostgreSQL-funktioner samler hvert svar som JSON, og Go-handleren modtager det som `json.RawMessage` og sender det uberørt videre gennem en lille `QueryJSON`-hjælper. De DTO'er, mappere og svarstrukturer, et lagdelt API plejer at samle op, faldt bort, handlerkoden blev markant kortere, og frontenden blev koblet fri af Go. Prisen er værd at nævne: svarenes form er nu defineret i SQL, så det er dér, en reviewer læser kontrakten.

Et API, som alle kan nå, må også kunne forsvare sig. Et offentligt register over virksomheder og fagfolk ligger bag 22 rate limiters, hver bygget til den sti, den beskytter, med deres tilstand i Redis, så en grænse gælder på tværs af instanser. Forsvinder Redis, fejler de lukket og afviser trafikken. Rundt om dem står Cloudflare Turnstile, 351 linjers middleware til idempotens, så en gentaget skrivning ikke bliver til to, en oprindelseslås og en skræddersyet udfordring

på selve registret. Prisen står der ligeud: et Redis-problem bliver et problem, brugerne kan se, og det blev foretrukket frem for et Redis-problem, der ender på regningen.

Hvor det går galt

Fejl i et API viser sig typisk sent, når klienterne og datamængden er vokset.

Et omdøbt felt knækker en klient, som ingen kendte til. Når et API først er offentliggjort, afhænger nogen af hver synlig detalje: feltnavne, rækkefølgen i en liste, ordlyden af en fejlbesked. Omdøb `customer_name` til `name`, og en partners integration knækker ved næste forespørgsel. Ændr i stedet ved at tilføje: læg det nye felt ved siden af det gamle, markér det gamle til udfasning, følg med i logfilerne, til det gamle felt står ulæst hen, og fjern det først i en ny hovedversion.

Et listeendpoint bliver langsommere hver måned. Det returnerer hver eneste række, som forespørgslen rammer, hvilket gik fint med halvtreds poster ved lanceringen og giver timeout ved to hundrede tusind. Paginér fra første udgivelse med en standardstørrelse pr. side og et hårdt loft, klienten ikke kan overskride. Til store eller travle samlinger er en cursor bedre end et offset: med et offset skubber en række, der indsættes mellem to forespørgsler, en anden række over på næste side, hvor klienten så læser den to gange.

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

En kunde læser en anden kundes ordre. API'et tjekker, at den, der kalder, er logget ind, og henter så `/v1/orders/42` alene ud fra id'et, så man ved at ændre tallet i adressen kan bladre gennem andres data. Autentificeringen svarede på, hvem der kalder; intet spurgte, om den, der kalder, ejer ordre 42. Afgræns hvert opslag efter ejer inde i selve forespørgslen, og hav en test, der logger ind som én konto og beder om en anden kontos poster.

Svaret indeholder kolonner, som ingen havde tænkt sig at offentliggøre. Et endpoint serialiserer en databaserække eller et ORM-objekt, som det står, så en hashet adgangskode, et internt flag eller en leverandørs indkøbspris går ud til alle klienter, og en kolonne, der tilføjes næste år, offentliggøres samme dag, den oprettes. Den samme fejl på input lader en forespørgsel sætte `role` eller

`is_admin` blot ved at have feltet med. Giv hvert endpoint eksplicitte input- og outputtyper, der opregner de tilladte felter, så det bliver en bevidst beslutning at offentliggøre et felt.

Læs mere

- RFC 9110: HTTP Semantics — de metoder, statuskoder og headere, som ethvert web-API er bygget af, defineret af IETF.
- RFC 9205: Building Protocols with HTTP — bedste praksis for at designe en applikation oven på HTTP, også for hvordan den udvikler sig.
- RFC 9457: Problem Details for HTTP APIs — en standardiseret JSON-form til fejlsvar, så klienter læser hver fejl på samme måde.
- RFC 6585: Additional HTTP Status Codes — hvor 429 Too Many Requests og headeren `Retry-After` til begrænsede klienter er defineret.
- Semantic Versioning 2.0.0 — reglerne for at signalere en inkompatibel API-ændring gennem et versionsnummer.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er et API?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/api/