Engineer

API

🎼 la manière documentée dont un programme en appelle un autre

Ce que c’est

Une API, abréviation de l’anglais application programming interface, est l’ensemble publié d’opérations qu’un logiciel propose à d’autres : nom de chaque opération, entrées acceptées, résultats et manière de signaler les problèmes. Le pluriel anglais est APIs. L’appelant connaît cette surface et rien de ce qui se trouve derrière. Le programme distant peut donc être réécrit, déplacé sur une autre machine ou divisé en trois sans interrompre l’appelant tant que la surface est maintenue.

Le mot couvre plusieurs couches. L’API d’une bibliothèque est l’ensemble des fonctions et types exportés, appelés dans un processus unique. Celle d’un système d’exploitation regroupe ses appels système. POSIX en normalise un socle commun pour qu’un programme utilisant `open` et `read` puisse être compilé sur plusieurs noyaux de type Unix. Une API Web est proposée sur un réseau, presque toujours par HTTP. C’est aujourd’hui le sens le plus courant : un serveur répondant à des adresses comme `/v1/orders`, généralement avec du JSON dans le corps.

Les API Web suivent quelques styles. Celles orientées ressources, souvent dites de style REST, attribuent une adresse à chaque objet et utilisent les méthodes HTTP pour le lire et le modifier. Celles de style RPC, comme gRPC, publient des procédures nommées dont arguments et résultats sont déclarés à l’avance. GraphQL propose un point d’entrée et un système de types : le client écrit une requête nommant exactement les champs souhaités. Les webhooks inversent le sens : le fournisseur appelle une adresse enregistrée par le client lorsqu’un événement survient.

Trois termes voisins sont souvent confondus. Un protocole, comme HTTP, définit le transport des messages ; une API est ce qu'un service particulier dit dans ce protocole. Un SDK est une bibliothèque dans un langage donné qui enveloppe une API pour appeler une fonction plutôt qu'assembler des requêtes. La mise en œuvre est le code derrière l'API, que l'appelant n'a jamais besoin de lire.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'une API ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. La partie invisible pour les utilisateurs : où résident les données, quelles règles s'appliquent et comment les autres systèmes peuvent y accéder.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 6 pipelines ETL/ELT conçus et exploités — 6 pipelines ETL sur trois moteurs de bases de données

6 pipelines ETL/ELT conçus et exploités Contactez-nous →

Pourquoi c'est important

L'API est la partie du système sur laquelle d'autres construisent, ce qui en fait une promesse survivant au code sous-jacent. Réécrire un gestionnaire prend un après-midi. Modifier son résultat peut casser une application déjà installée sur les téléphones, un import nocturne partenaire ou un script que personne ne se souvient avoir écrit.

La première décision concerne donc le public. Une API privée, utilisée seulement par l'interface de l'équipe, peut changer dans la même demande de fusion que son unique client. Une API publique évolue par annonce et obsolescence progressive. Chaque champ publié devra être pris en charge aussi longtemps que quelqu'un le lit.

La deuxième décision est la rigueur du contrat. Une API stricte déclare type et limite pour chaque entrée, rejette l'inconnu et répond à chaque échec par un code de statut précis. Elle intercepte les erreurs à l'entrée et facilite documentation, tests et génération de clients. Une API souple accepte ce qui arrive et fait au mieux. Cela semble accueillant le premier mois, puis chaque accident de mise en œuvre devient un comportement dont quelqu'un dépend. La rigueur coûte du temps de conception initial ; la souplesse coûte davantage plus tard, à un moment que personne ne choisit.

La troisième décision concerne les API d'autrui. Prestataire de paiement, service cartographique ou modèle de langage constituent une API appelée sans contrôle : prix changeants, versions retirées, quotas durcis. Appeler directement un fournisseur depuis cinquante endroits transforme son remplacement en réécriture. L'appeler depuis un module maîtrisé par l'équipe limite le changement à ce module. L'enveloppe coûte un peu d'indirection et offre une porte de sortie.

Semantic Versioning donne un vocabulaire à cette promesse : le numéro majeur augmente si l'API publique devient incompatible, pour que la version seule indique si une mise à niveau est sûre. Plusieurs endroits peuvent signaler une rupture sur HTTP. La RFC 9205, recommandations de l'IETF pour les protocoles construits dessus, cite notamment une nouvelle relation de lien, un nouveau type de média ou un nouvel en-tête.

À quoi cela ressemble en pratique

Les API de plusieurs projets de ce site ont été façonnées par une question : qu'est-ce qui appartient à l'API, et qu'est-ce qui doit résider ailleurs ?

Sur NextMariner, plateforme maritime de réseau professionnel, d'avis sur les entreprises et de formations sponsorisées, la contribution de la fondatrice à l'architecture comprenait le maintien d'une API Go volontairement légère. Construite avec Huma sur Fiber, elle route, valide les requêtes, applique le filtrage de sécurité et orchestre, sans aucune logique métier. Les règles sont dans des fonctions PostgreSQL, ce qui limite leur changement à une couche, en SQL. Les frontières ont été écrites et imposées en revue. L'API est ainsi un adaptateur banal, dans le bon sens du terme.

Un projet lié a également retiré le façonnage des réponses de l'API. Les fonctions PostgreSQL assemblent chaque réponse en JSON. Le gestionnaire Go la reçoit en `json.RawMessage` et la transmet intacte via une petite fonction utilitaire `QueryJSON` . DTO, fonctions de conversion et structures de réponse habituellement accumulés ont disparu, les gestionnaires ont considérablement raccourci et l'interface ne dépend plus de Go. La contrepartie mérite d'être nommée : les réponses sont désormais définies en SQL, où la personne chargée de la revue doit lire le contrat.

Une API accessible à tous doit aussi se défendre. Un annuaire public d'entreprises et de professionnels est protégé par 22 limiteurs de débit, chacun adapté au chemin qu'il protège, avec leur état dans Redis pour maintenir la limite entre

instances. Si Redis est inaccessible, ils sont configurés pour refuser le trafic. Cloudflare Turnstile, un verrouillage de l'accès au serveur d'origine, un défi spécifique à l'annuaire et 351 lignes de couche intermédiaire d'idempotence empêchant une écriture répétée complètent les défenses. La page en indique clairement le coût : un problème Redis devient visible des utilisateurs, choix préféré à sa transformation en problème de facturation.

Où les problèmes surviennent

Les échecs d'API apparaissent souvent tard, lorsque clients et données ont grandi.

Un champ renommé casse un client inconnu. Dès la publication, quelqu'un dépend de chaque détail observable : noms de champs, ordre de liste, formulation d'erreur. Renommer `customer_name` en `name` casse l'intégration d'un partenaire à sa requête suivante. Procédez par ajout : nouveau champ près de l'ancien, marquage de l'ancien comme obsolète, surveillance des journaux jusqu'à disparition des lectures, puis suppression seulement dans une nouvelle version majeure.

Un point d'entrée de liste ralentit chaque mois. Il renvoie toutes les lignes correspondantes, ce qui fonctionnait avec cinquante enregistrements au lancement mais dépasse le délai à deux cent mille. Paginez dès la première version avec une taille de page appliquée si le client n'en indique pas et un maximum imposé au client. Pour de grandes collections ou des collections actives, préférez un curseur au décalage : une insertion entre deux requêtes décale sinon une ligne sur la page suivante, où le client la lit deux fois.

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

Un client lit la commande d'un autre. L'API vérifie la connexion puis charge `/v1/orders/42` par son seul identifiant. Modifier le numéro parcourt donc les données d'autrui. L'authentification a identifié l'appelant, mais rien n'a demandé s'il possède la commande 42. Limitez chaque recherche au propriétaire dans la requête elle-même, et conservez un test qui se connecte avec un compte et demande les enregistrements d'un autre.

La réponse contient des colonnes jamais destinées à la publication. Un point d'entrée sérialise telle quelle une ligne de base ou un objet ORM : hachage de mot de passe, indicateur interne ou prix coûtant fournisseur part alors à chaque client. Une colonne ajoutée l'année suivante est publiée dès sa création. La même erreur en entrée permet de définir `role` ou `is_admin` en les plaçant dans le corps. Donnez à chaque point d'entrée des types explicites d'entrée et de sortie listant les champs autorisés, afin que publier un champ devienne une décision.

Pour aller plus loin

- RFC 9110 : HTTP Semantics — méthodes, statuts et en-têtes des API Web définis par l'IETF.
- RFC 9205 : Building Protocols with HTTP — bonnes pratiques actuelles de conception sur HTTP, évolution comprise.
- RFC 9457 : Problem Details for HTTP APIs — format JSON standard des erreurs pour une analyse uniforme des échecs.
- RFC 6585 : Additional HTTP Status Codes — définition de 429 Too Many Requests et de `Retry-After` pour les clients limités en débit.
- Semantic Versioning 2.0.0 — règles de signalement d'un changement incompatible par un numéro de version.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'une API ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

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

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/api/