Les API

Salsabil ZAGHDOUDI

Promo : data ingé 2024

API c’est quoi ?

API = interface pour un code.

On fait quoi avec ?

Les APIs relient les applications entre elle.

Pourquoi les APIs ?

Une application c’est un code qui fonctionne sur un serveur.

L’application en back-end contient plusieurs applications.

Les APIs établissent la connexion entre ces applications.

Ex: API Uber Drive

Collection de différentes applications

Données nécessaires:

- Une application pour le paiement.
- Une application pour calculer votre position actuelle.
- Une application qui gère l'avis que vous laissez à la fin du trajet.

Focus sur l'appli des avis

Cette application a besoin de données :
1. Qui a pris la course ?
2. Qui était le conducteur ?
3. Combien de départs ont eu lieu ?

Ces données sont ce qui se connecte à l'API.

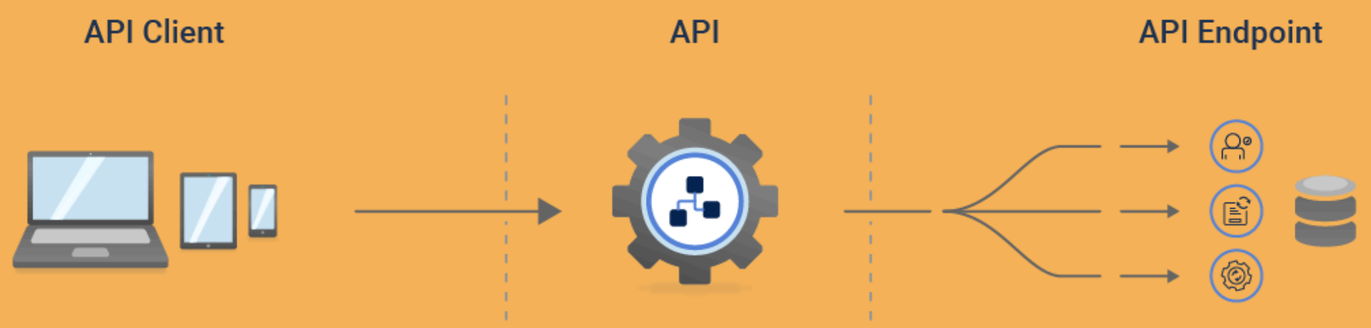
UI vs API ?

UI : User Interface connectée en front end, accessible à l'utilisateur.

API : Même fonctionnement que les UI mais en back end pour les développeurs.

API Endpoints ?

Un point de terminaison API, est une URL désignée à laquelle un client peut envoyer des demandes pour accéder à un service web ou à une application et interagir avec eux.



Focus sur les Endpoint API

Définition:
Un endpoint d'API est une URL qui représente un point de contact entre un client (ex. : une application) et un serveur d'API. Le client envoie des requêtes à cet endpoint pour accéder aux fonctionnalités ou aux données fournies par l'API.

Focus sur les Endpoint API

Endpoint comment ça marche ?

Les endpoints relient les clients et les serveurs. Chaque endpoint correspond à une ressource.

Exemple API REST

- GET /authors → liste des auteurs
- POST /authors → créer un auteur
- GET /authors/:id → infos d'un auteur spécifique
- PUT /authors/:id → modifier un auteur
- DELETE /authors/:id → supprimer un auteur

Focus sur les Endpoint API

Chaque requête inclut :

- Méthode HTTP (GET, POST...)
- Paramètres (ex : ?category=AI pour filtrer)
- Headers (ex Authorization)
- Corps de la requête (pour POST/PUT)

Le serveur traite la requête, exécute l'action demandée, puis renvoie une réponse avec un code de statut (200, 404, etc.) et éventuellement des données.

Les 5 éléments clés d’un endpoint d’API

1.URL de l’endpoint
C’est l’adresse unique pour accéder à une ressource (ex. /users, /articles).

2.Méthodes HTTP
Définissent l’action à faire, GET, POST, PUT, DELETE

3. Paramètres et requêtes
On peut filtrer ou cibler une ressource via des paramètres dans l’URL (ex. ?id=5) ou envoyer des données dans le corps de la requête (avec POST, PUT).

4.Réponses
Le serveur renvoie un code de statut (ex. 200 OK, 404 Not Found)
une réponse au format JSON ou XML avec les données demandées.

5. Authentification et sécurité
Certains endpoints nécessitent un jeton d’accès, une clé API ou un système comme OAuth2 pour s’assurer que seul un utilisateur autorisé accède aux données.

Endpoints - Ex pratique

Supposons que vous ayez une API pour une bibliothèque en ligne. Voici quelques endpoints possibles :

- **GET** https://api.library.com/books : Récupère la liste de tous les livres.
- **GET** https://api.library.com/books/{id} : Récupère les informations d'un livre spécifique en utilisant son identifiant.
- **POST** https://api.library.com/books : Ajoute un nouveau livre à la bibliothèque.
- **PUT** https://api.library.com/books/{id} : Met à jour les informations d'un livre existant.
- **DELETE** https://api.library.com/books/{id} : Supprime un livre de la bibliothèque.

Type des APIs

API privées

- Internes à l'entreprise.
- Connecter des systèmes et des données au sein de l'entreprise.

API publiques

- Ouvertes au public et peuvent être utilisées par n'importe qui.
- Peuvent être soumis ou non à des autorisations et à des coûts.

API partenaires

- Accessibles qu'aux développeurs externes autorisés pour faciliter les partenariats interentreprises.

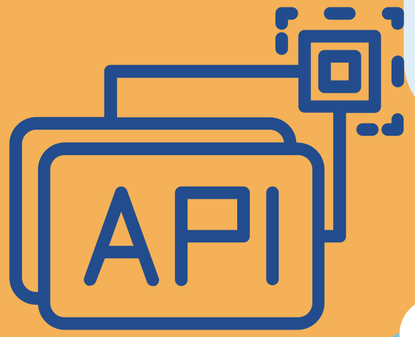
→

API SOAP :

Ces API utilisent le protocole d'accès simple aux objets. Le client et le serveur échangent des messages à l'aide de XML. Il s'agit d'une API moins flexible qui était plus populaire dans le passé.

API Websocket :

L'API Websocket est une autre API web moderne qui utilise des objets JSON pour transmettre des données. Elle prend en charge la communication bidirectionnelle entre les applications clientes et le serveur. Le serveur peut envoyer des messages de rappel aux clients connectés, ce qui le rend plus efficace que l'API REST.



API RPC :

Ces API sont appelées appels de procédure à distance. Le client exécute une fonction (ou une procédure) sur le serveur, et le serveur renvoie le résultat au client.

API REST :

Il s'agit des API les plus populaires et les plus souples que l'on trouve aujourd'hui sur le web. Le client envoie des demandes au serveur sous forme de données. Le serveur utilise ces données pour lancer des fonctions internes et renvoie les données de sortie au client. Nous allons examiner les API REST plus en détail.

Focus sur les API REST :

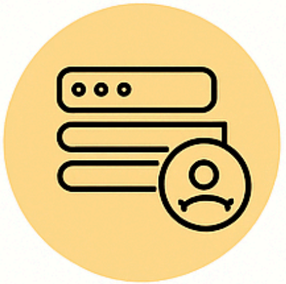
- REST signifie Representational State Transfer.
- REST définit un ensemble de fonctions telles que : GET, PUT, DELETE, etc. pour accéder aux données du serveur par le client.
- Les clients et les serveurs échangent des données à l'aide du protocole HTTP.

Focus sur les API REST :

- **La principale caractéristique de l'API REST est l'absence d'état. : Absence d'État (Statelessness)**
- **Stateless** = sans mémoire
- Chaque requête est indépendante :
  - Le serveur ne se souvient pas des échanges précédents.
  - Le client doit tout renvoyer à chaque fois : identifiant, données, etc.
  - **Avantages** : plus rapide, plus simple à gérer, plus facile à faire évoluer.

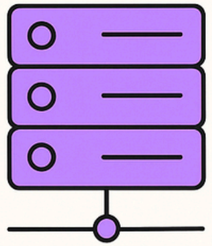
Question : Renvoyer les informations du client à chaque requête ne crée-t-il pas une surcharge ou une redondance dans les échanges ?

Réponse :



Les données envoyées sont souvent légères, généralement composé de headers et de jetons d'authentification (Token).

Standardisation = compatibilité facile le serveur peut traiter chaque appel sans contexte, ce qui rend l'API plus robuste, plus facile à déboguer et plus scalable



Caching facilité

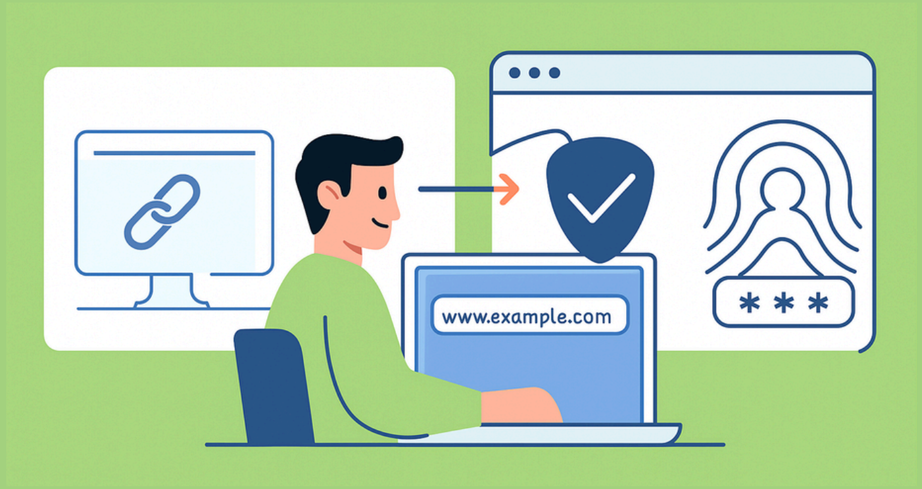
Les requêtes indépendantes facilitent l'utilisation de caches.

Le client garde l'état

Il peut stocker ce qu'il veut en local et le renvoyer à chaque appel

En conclusion :

Le Statelessness signifie que les serveurs ne sauvegardent pas les données des clients entre les requêtes. Les demandes du client au serveur sont similaires aux URL que vous tapez dans votre navigateur pour visiter un site web. La réponse du serveur est constituée de données simples, sans le rendu graphique typique d'une page web.



Qu'est-ce qu'une intégration API ?

Les intégrations API sont des composants logiciels qui mettent automatiquement à jour les données entre les clients et les serveurs.

Ex: la synchronisation automatique des données de la galerie d'images de votre téléphone vers le cloud.

L'intégration d'API permet à deux applications ou systèmes de communiquer entre eux automatiquement, en temps réel, grâce à une API.
Objectif : automatiser l'échange de données entre logiciels, sans intervention humaine.

Exemples concrets

Synchroniser automatiquement les photos de ton téléphone vers le cloud

Connecter un ERP avec un site e-commerce pour suivre les commandes en temps réel

Partager des données de livraison entre un entrepôt et un transporteur

Méthodes les plus utilisées pour sécuriser une API

JSON WEB Token

Définition :
Un JWT est un jeton d'authentification utilisé pour prouver l'identité d'un utilisateur, souvent dans une API stateless. Il est stocké côté client et transmis à chaque requête pour éviter la gestion de session côté serveur.

Structure d'un JWT :
Header.Payload.Signature

- **Header :** indique l'algorithme (ex : HS256)
- **Payload :** contient les données (claims : id, rôle, etc.)
- **Signature :** assure l'intégrité du jeton (calculée à partir du header + payload + clé secrète)

Méthodes les plus utilisées pour sécuriser une API

OAuth 2.0

Définition :
OAuth 2.0 est un protocole d'autorisation (et non d'authentification) permettant à une application d'accéder à des ressources au nom d'un utilisateur, sans exposer son mot de passe.

Fonctionnement :

- L'utilisateur autorise l'application à accéder à ses données.
- Le serveur d'autorisation émet un jeton d'accès (access token).
- L'application utilise ce jeton pour interagir avec le serveur de ressources.

Types de jetons :

- Jeton d'accès : autorisation temporaire
- Jeton d'actualisation : permet de renouveler un jeton d'accès expiré

Méthodes les plus utilisées pour sécuriser une API

OAuth 2.0

Flux principaux :

- **Code d'autorisation (PKCE) :** recommandé pour les apps mobiles ou web
- **Identifiants client :** pour les services automatisés
- **Flux implicite :** obsolète car moins sécurisé

Avantages :

✅ Séparation entre l'application et les identifiants utilisateurs

✅ Moins de risques liés au vol de mots de passe

✅ Contrôle fin des autorisations via les scopes (ex. lecture seule, écriture...)

Méthodes les plus utilisées pour sécuriser une API

API Keys

Définition
Une clé API est un identifiant unique utilisé pour authentifier une application ou un projet lors d'un appel à une API.

Pourquoi l'utiliser ?

- Identifier l'application ou le projet appelant l'API
- Suivre et analyser les usages
- Bloquer le trafic anonyme
- Limiter le nombre de requêtes (quotas)
- Filtrer les journaux d'événements

Exemple d'appel API avec une API Key :

```
GET
https://api.weatherexample.com/v1/forecast?city=Paris
Authorization: ApiKey 12345abcdef67890
```

Méthodes les plus utilisées pour sécuriser une API

HMAC et HTTPS

4. HMAC (Hash-based Message Authentication Code) 🛡️

- ✓ Génére une signature cryptée avec une clé secrète partagée
- ✓ Permet de vérifíer que la requête n'a pas été altérée
- → Utilísé dans des systèmes sensibles (ex.: banques, services critiques)

5. HTTPS obligatoire 🌐

- ✓ Chiffre toutes les données échangées entre ciient et serveur
- ✓ Empêche l'interceptíon et l'espionnage des données

Quelles technologies pour explorer et créer des API ?

Postman

Application qui permet d’interroger et tester des APIs via une interface graphique qui est user friendly

Python

Automatisation des méthodes API dans une appli, c’est beaucoup plus utilisé dans le prod et le dev et graçce à la biblio Requests (Put, POST, Get...)

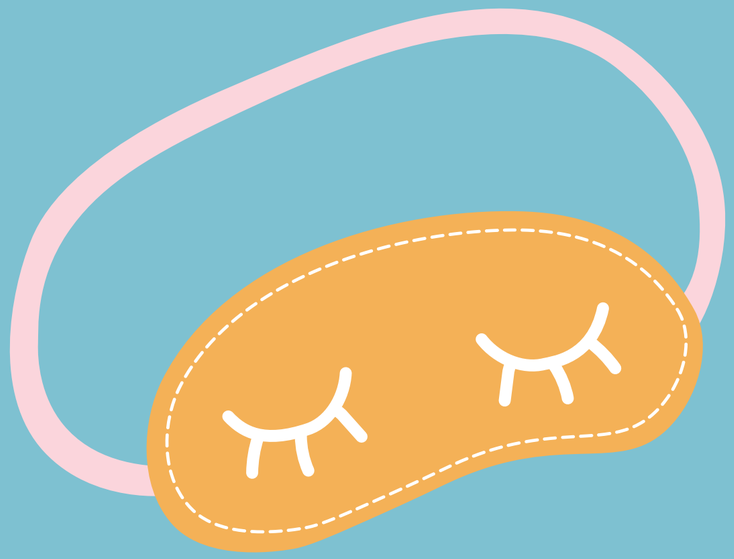
Prérequis pour les APIs ?

1 Concepts de base :

- HTTP Protocol
- REST
- JSON

2 Bases de Python :

- Structure des données
- Les fonctions
- Les librairies



We stop here