

1 Concept

Une API Web (***Application Programming Interface***) est une interface qui permet à deux applications de communiquer via le réseau, le plus souvent grâce au protocole HTTP/HTTPS.

👉 Une API expose des **ressources** (données ou services) accessibles par des URL appelées ***endpoints*** et manipulables via des requêtes HTTP.

2 À quoi ça sert ?

- **Accéder à des données distantes** : météo, utilisateurs, capteurs IoT...
- **Piloter des services** : envoi de messages, commandes, actions
- **Faire communiquer un client** : application/automate industriel/script ↔ serveur

3 Architecture client / serveur

- **Client** : application, navigateur, script, automate...
 - **Serveur** : héberge l'API et traite les requêtes
- ① Le client envoie une **requête HTTP** → ② le serveur renvoie une **réponse HTTP**.

4 Anatomie d'une requête

4 parties principales :

① La méthode HTTP

Méthode	Rôles
GET	Lire / Récupérer des données
POST	Envoyer / Créer des données
PUT	Modifier une ressource
DELETE	Supprimer une ressource

② L'URL de l'endpoint

- Adresse de la ressource ciblée.

Exemple :

```
https://api.exemple.com/utilisateurs
```

- Paramètres (uniquement avec la méthode GET)

Exemple : Demande les utilisateurs du département IT qui sont connectés

```
https://api.exemple.com/utilisateurs?department=it&connected=true
```

- '?' : début des paramètres
- param=valeur : couple clé / valeur
- '&' : séparateur entre plusieurs paramètres

⚠ Remarques importantes :

- Les paramètres GET sont visibles dans l'URL
- Ils servent surtout à filtrer, rechercher ou trier des données
- Ils ne doivent pas contenir d'informations sensibles (ex : mot de passe)

③ Les en-têtes HTTP

Informations supplémentaires sur la requête.

Exemples courants :

- **Content-Type** : format des données envoyées (JSON, XML...)
- **Authorization** : authentification (clé API, token)
- **Accept** : format attendu en réponse

④ Le corps de la requête (body)

Contient les paramètres envoyés au serveur avec les méthodes POST et PUT.

► Contrairement à GET, les paramètres ne sont pas dans l'URL, mais dans le corps de la requête.

Paramètres souvent exprimés en JSON

Exemple de body JSON :

```
{  "nom": "Dupont",  "age": 21,  "actif": true}
```

- Ces données sont envoyées au serveur pour :
 - **POST** → créer une nouvelle ressource
 - **PUT** → modifier une ressource existante
- Importance des headers avec POST / PUT

Certains headers sont indispensables. Exemple :

Content-Type: application/json

👉 Indique au serveur comment interpréter les données du body.

5 Anatomie d'une réponse

- Une réponse contient :
 - **Code de statut HTTP**
 - **Headers**
 - **Corps de la réponse (données)**
 - Codes de statut courants :
- | Code | Signification |
|------|-----------------------|
| 200 | OK – succès |
| 201 | Création réussie |
| 400 | Requête incorrecte |
| 401 | Non autorisé |
| 404 | Ressource non trouvée |
| 500 | Erreur serveur |
- Le corps de la réponse contient souvent des données au format JSON, XML ou HTML.
 - Les headers de la réponse peuvent indiquer le format des données (ex : Content-Type: application/json) ou des informations de pagination, de cache, etc.

6 Sécurité et bonnes pratiques

- Pour protéger l'accès à l'API :
- Clé API
 - Token (Bearer Token)
 - HTTPS obligatoire
- ⚠ Ne jamais exposer une clé API en clair dans une application publique.

7 Outils de test d'API

- navigateur : pour les requêtes GET simples
- **curl** : outil en ligne de commande pour envoyer des requêtes HTTP
- **Postman** : interface graphique pour tester les requêtes et visualiser les réponses