



Guide – API First avec OpenAPI

Objectif

Avant d'écrire une seule ligne de code backend, vous allez concevoir et documenter votre API à l'aide d'OpenAPI. L'approche API First consiste à définir le contrat d'échange entre le front et le back avant toute implémentation, pour garantir une bonne coordination.

1. Comprendre la philosophie API First

API First, c'est l'idée suivante :

"On commence par définir ce que l'API doit faire avant de coder comment elle le fait."

Cela permet :

- De valider les besoins fonctionnels avant le développement.
- De travailler en parallèle (le front peut commencer dès que la spec est prête).
- D'avoir une source de vérité unique (la spec OpenAPI).
- D'éviter les mauvaises surprises d'intégration à la fin.

2. Qu'est-ce qu'OpenAPI ?

OpenAPI (anciennement Swagger) est un standard de description d'API REST. C'est un fichier `.yaml` ou `.json` qui décrit :

- Les endpoints disponibles (`/users` , `/login` , `/products` ...),
- Les méthodes HTTP utilisées (GET, POST, PUT, DELETE...),
- Les paramètres et corps de requêtes,
- Les réponses attendues,
- Et les types de données échangés.

Ce fichier devient :

- Une documentation vivante,
- Une base de génération automatique (types, clients, mocks, etc.),
- Et un contrat officiel entre front et back.

3. Étapes pour travailler en API First

1 Étape 1 – Créer la spec OpenAPI

- Créez un fichier `openapi.yaml` à la racine du repo backend (ou dans un dossier `/docs`).
- Décrivez vos routes principales : endpoints, entrées, sorties, statuts HTTP.
- Vous pouvez utiliser des outils graphiques comme :

- Stoplight Studio
- Swagger Editor
- Insomnia Designer

2 Étape 2 – Publier la spec

- Rendez-la accessible dans votre repo (ex: `/docs/openapi.yaml`).
- Ajoutez un moyen de visualisation automatique : Swagger UI, Redoc ou Stoplight pour naviguer dans la documentation.

3 Étape 3 – Utiliser la spec côté front

- Le frontend ne code pas "au hasard" : il génère ses types et services HTTP directement depuis la spec.
- Cela évite les divergences entre ce que le back renvoie et ce que le front attend.

4 Étape 4 – Implémenter ensuite le backend

- Le backend code les routes en se conformant strictement à la spec.
- Chaque endpoint doit respecter les schémas de requêtes et réponses définis.
- Si la spec change, on incrémente la version (ex: v1.1) et on documente le changement.

4. Bonus – Mock et tests rapides

Avant que le backend soit prêt, vous pouvez simuler les réponses de l'API grâce à la spec :

- Utilisez **Prism** ou **Swagger Mock Server** pour générer un serveur mock.
- Cela permet au frontend de tester son intégration même sans backend fonctionnel.

5. Livrables attendus pour l'équipe

Pour chaque équipe (back ou front), les éléments suivants doivent exister :

Élément	Description
<code>openapi.yaml</code>	Spécification complète de l'API
Documentation	Swagger UI / Redoc accessible
Postman Collection	Synchronisée automatiquement depuis la spec
Types générés	Types utilisés côté front à partir de la spec
Versioning	Numéro de version et changelog en cas de modification



6. En résumé

L'API First, c'est concevoir avant de coder.

C'est une démarche de collaboration et de rigueur :

- Une communication claire entre les équipes front et back,
- Une documentation fiable et toujours à jour,
- Un gain de temps énorme à long terme.