

Comment travailles-tu avec un frontend pour définir un endpoint ?

Je commence toujours par une discussion claire avec le développeur frontend pour comprendre le besoin fonctionnel et l'expérience utilisateur visée. On définit ensemble :

Le format des données attendu (JSON structure, pagination, etc.)

Les verbes HTTP appropriés (GET, POST, PUT, DELETE)

Les éventuelles contraintes (authentification, rôles, délais) Ensuite, je propose un schéma d'URL RESTful cohérent et prévisible, par exemple : `/api/trips/{id}` ou `/api/users/{id}/trips`. Je m'assure que les réponses soient claires, avec des codes HTTP explicites (200, 401, 422, etc.) et des messages d'erreur utiles. Enfin, je teste les endpoints avec Postman et je partage des exemples de requêtes pour faciliter l'intégration côté frontend.

Comment documentes-tu une API pour qu'elle soit simple à utiliser ?

Je documente l'API comme si j'étais le développeur qui va l'utiliser demain.

J'utilise des outils comme Swagger/OpenAPI ou Postman Collection pour générer une documentation interactive.

Chaque endpoint est décrit avec :

Son URL, méthode HTTP, paramètres requis

Exemple de requête et de réponse

Codes d'erreur possibles et leur signification

Je garde la documentation à jour à chaque modification du code. En plus, je rédige un README clair dans le repo GitHub avec les étapes d'installation, les routes principales, et les règles d'accès. Mon objectif : que n'importe quel développeur puisse comprendre et tester l'API en moins de 10 minutes.

Si un client demande une fonctionnalité en urgence, comment gères-tu ça sans casser la stabilité ?

Je commence par évaluer l'impact technique : est-ce une modification isolée ou une partie critique du système ? Ensuite, je propose une solution en deux temps :

Livrer une version minimale fonctionnelle (MVP) qui répond au besoin sans toucher aux composants sensibles.

Planifier une refonte propre si nécessaire, dans un sprint suivant. Pour éviter les régressions, je :

Crée une branche dédiée (hotfix/urgent-feature)

Ajoute des tests automatisés ciblés

Déploie d'abord en staging pour validation rapide

Communique clairement avec le client sur ce qui est livré et ce qui reste à stabiliser Mon objectif est de répondre vite sans compromettre la qualité ni la sécurité du produit.