

# Tester avec Jest et Supertest



# Pourquoi les tests unitaires ?

**1**

## **Qualité du code**

Les tests assurent la fiabilité et la robustesse de votre application.

**2**

## **Détection précoce des bugs**

Vous pouvez identifier et corriger les erreurs rapidement.

**3**

## **Refactoring en toute confiance**

Vous pouvez apporter des modifications sans craindre de casser le code existant.

**4**

## **Documentation vivante**

Les tests servent de documentation pour comprendre le fonctionnement de votre code.

# Jest : un framework de test puissant

## Rapide et efficace

Jest est conçu pour être performant, avec une exécution parallèle des tests pour des résultats rapides.

## Assertions intuitives

Les assertions de Jest sont claires et faciles à comprendre, facilitant l'écriture de tests.

## Couverture de code

Jest fournit des outils pour mesurer la couverture de votre code par les tests.

# Supertest : tester les API HTTP

## Syntaxe simplifiée

Supertest offre une API concise et intuitive pour envoyer des requêtes HTTP et vérifier les réponses.

## Compatibilité avec Jest

Supertest peut être facilement combiné avec Jest pour des tests d'intégration complets.

## Validation des réponses

Supertest vous permet de vérifier le statut, les en-têtes et le corps des réponses HTTP.

## Simulation des requêtes

Vous pouvez simuler différents types de requêtes (GET, POST, PUT, DELETE) avec Supertest.

# Configuration de Jest

1

## Installer Jest

Ajoutez Jest à votre projet en tant que dépendance de développement.

2

## Créer `jest.config.js`

Configurez Jest avec les options dont vous avez besoin pour vos tests.

3

## Organiser les tests

Placez vos fichiers de test dans un répertoire dédié, par exemple "tests".



# Écrire des tests avec Jest et Supertest

1

## Importer les dépendances

Importez les modules nécessaires pour accéder à votre application et envoyer des requêtes HTTP.

2

## Définir les tests

Utilisez les fonctions de Jest (describe, it, expect) pour décrire et vérifier le comportement de votre code.

3

## Exécuter les tests

Lancez Jest dans votre terminal pour faire tourner l'ensemble de vos tests.



# Avantages des tests unitaires



## Fiabilité

Les tests assurent la stabilité et la robustesse de votre application.



## Détection précoce

Vous pouvez identifier et corriger les bugs rapidement.



## Refactorisation

Vous pouvez modifier votre code en toute confiance.



## Documentation

Les tests servent de documentation vivante pour votre code.

# Conclusion

Jest	Supertest
Framework de test JavaScript puissant	Bibliothèque pour tester les API HTTP
Exécution rapide et parallèle des tests	Syntaxe simple pour envoyer des requêtes
Assertions claires et concises	Validation des réponses HTTP
Mesure de la couverture de code	Compatibilité avec Jest