

Mis à jour le 31/01/2025

S'inscrire

Formation Vitest : Automatisez vos tests

2 jours (14 heures)

PRÉSENTATION

Notre formation Vitest vous permettra d'améliorer la qualité de votre code et de faciliter l'automatisation de vos tests unitaires. Vous pourrez facilement tester vos applications Java, Kotlin et Android pour garantir la qualité de votre code.

Dans cette formation, destinée aux développeurs, vous apprendrez à utiliser les principales fonctionnalités de Vitest.

Telles que la création de tests unitaires, l'exécution de tests et la génération de rapport.

Vous découvrirez également comment intégrer Vitest avec d'autres technologies telles que des frameworks de développement et des outils d'intégration continue pour améliorer la qualité de votre code et accélérer la livraison de vos applications.

Cette formation vous présentera l'ensemble des fonctionnalités de Vitest, avec un accent particulier sur les bonnes pratiques de [tests unitaires](#), de validation de code et d'intégration continue.

Vous apprendrez également comment configurer vos tests unitaires pour des environnements spécifiques, tels que le développement Android ou les applications basées sur Spring.

Comme dans toutes nos formations, celle-ci vous présentera la dernière version stable : [Vitest 0.29](#).

Objectifs

- Configurer et utiliser Vitest dans un projet JavaScript.
- Écrire des tests unitaires, asynchrones et d'intégration.
- Utiliser des mocks pour simuler des modules, des fonctions et des dates.

- Travailler avec les snapshots pour valider des réponses complexes d'interface utilisateur ou d'objets.
- Organiser des tests en utilisant des hooks et des méthodes de nettoyage

Public visé

- Développeurs
- Architectes techniques
- Chefs de projets

Pré-requis

- Avoir des bases en JavaScript et en HTML/CSS

Programme de notre formation Vitest

Introduction aux tests

- Pourquoi tester ?
- Types de tests : unitaires, d'intégration, E2E
- Présentation des différents frameworks de test (Jest, Mocha, Vitest)
- Avantages de Vitest : Rapide, intégré avec Vite, support natif des tests asynchrones

Mise en place de Vitest

- Installation et configuration de Vitest dans un projet JavaScript
- Structure d'un projet de tests
- Configuration de Vitest (fichier vitest.config.ts)

Les fonctions describe(), test() et it()

- Qu'est-ce que describe ? Structure logique des tests
- test() et it(), comment les utiliser pour définir des scénarios de tests
- Exercice pratique : créer une première suite de tests pour une fonction simple

La fonction expect() et les matchers

- Présentation de expect()
- Différents matchers de Vitest : toBe(), toEqual() toHaveBeenCalled(), etc
- Exercice : Tester des fonctions simples avec des assertions

Les tests asynchrones

- Tester des fonctions asynchrones avec `async/await`
- Utilisation de `vitest` pour gérer les promesses et les délais
- Exercice : Tester une fonction asynchrone retournant une promesse

Préparation et nettoyage

- Les hooks de préparation et nettoyage : `beforeAll()`, `afterAll()`, `beforeEach()`, `afterEach()`
- Pourquoi et quand les utiliser dans un projet JavaScript
- Exercice : Configurer des hooks pour initialiser et nettoyer des données

Les fonctions simulées (Mock Functions)

- Introduction aux fonctions simulées dans `Vitest`
- Utilisation de `vi.fn()` pour créer des fonctions simulées
- Exercice : Simuler une fonction de callback dans un test

Les modules simulés (Mocking Modules)

- Comment mocker des modules externes (par exemple, `axios`, `localStorage`, etc.).
- Utilisation de `vi.mock()` pour remplacer un module dans les tests
- Exercice : Mocker un appel API avec `axios`

Les dates simulées (Mocking Dates)

- Comment simuler des dates avec `Vitest`
- Utilisation de `vi.setSystemTime()` pour fixer une date et l'utiliser dans les tests
- Exercice : Tester un module qui utilise des dates et heures

Snapshots pour les tests d'interface utilisateur

- Introduction aux snapshots pour valider des réponses complexes (ex: interfaces utilisateur ou objets)
- Gérer et mettre à jour les snapshots avec `Vitest`
- Exercice : Tester une interface utilisateur dynamique et capturer un snapshot

Tests d'intégration dans un projet JavaScript

- Tester un module ou un service dans un projet JavaScript avec plusieurs dépendances
- Exercice : Tester un formulaire dynamique avec plusieurs composants et états

Tests end-to-end avec `Vitest`

- Simuler un scénario complet d'utilisation dans une application JavaScript
- Mocker des services externes (par exemple, envoyer des emails, appeler des API externes)
- Exercice : Tester un flux utilisateur complet sur une page web

Clôture de la formation

- Révision des concepts abordés
- Questions/réponses et conseils pour aller plus loin

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.