Tests Unitaires et d’Intégration avec Vitest

Lien : https://innov-systems.com/formation/tests-unitaires-et-dintegration-avec-vitest

DURÉE
3 jours (21h)

RÉFÉRENCE
TWC244

CATÉGORIE
Frameworks Ember, Backbone, Node, Ionic, Sencha, Typecript, Ecmascript, Polymer

OBJECTIFS DE LA FORMATION

À l'issue de cette formation, vous serez capable de :

- Comprendre les principes fondamentaux du testing moderne dans les environnements front-end
- Installer, configurer et exploiter efficacement Vitest dans un projet Vite, React ou Vue
- Maîtriser l’écriture de tests unitaires, d’intégration et asynchrones
- Utiliser les mocks, spies et snapshots pour fiabiliser les tests
- Mettre en place une stratégie de tests performante et maintenable sur le long terme
- Intégrer les bonnes pratiques de performance, couverture et debugging dans le workflow de développement

POUR QUI ?

- Développeurs front-end et full-stack souhaitant structurer leurs tests avec Vitest
- Lead développeurs et technical managers désirant standardiser les tests au sein de leur équipe
- Développeurs souhaitant migrer depuis Jest ou Mocha vers un environnement plus moderne et rapide

Innov Systems

Programme détaillé

1 / COMPRENDRE LA PLACE DE VITEST DANS L'ÉCOSYSTÈME DU TESTING

- Pourquoi adopter Vitest : philosophie, positionnement et bénéfices
- Vitest vs Jest, Mocha, Jasmine : comparatif technique et performance
- Concepts fondamentaux : test runner, assertions, spies, mocks, snapshots

2 / INSTALLATION ET CONFIGURATION INITIALE

- Installation de Vitest dans un projet Vite (npm, pnpm, yarn)
- Configuration de base : vitest.config.ts et vite.config.js
- Découverte de l'environnement CLI : options, modes et intégration CI/CD
- Structurer les dossiers et fichiers de test selon les bonnes pratiques

3 / PREMIERS TESTS ET STRUCTURATION DU CODE

- Écriture d'un test unitaire simple avec expect()
- Groupement logique avec describe() et gestion des hooks (beforeEach, afterAll…)
- Bonnes pratiques d'écriture et de nommage
- Exercices guidés : tests de fonctions utilitaires (formatage, calcul, validation)

4 / TESTS ASYNCHRONES ET GESTION DES EFFETS SECONDAIRES

- Tests avec async/await, promises et gestion des rejets
- Simulation de délais, timeouts et comportements réseau
- Tests d'erreurs et de comportements inattendus

5 / MOCKS, SPIES ET SNAPSHOTS

- Introduction à vi.fn(), vi.spyOn() et vi.mock()
- Cas d'usage : mocking d'API, modules et hooks personnalisés

Innov Systems

- Snapshot testing : création, mise à jour et gestion des différences
- Exercices pratiques : simuler un appel API et valider les retours

6 / STRUCTURER UNE STRATÉGIE DE TESTS D'APPLICATION

- Tests unitaires vs tests d'intégration : frontières et complémentarités
- Hiérarchiser la couverture : logique métier, composants, interactions
- Organisation des suites de tests et stratégies de maintenance
- Mise en place d'une exécution parallèle et gestion du cache

7 / TESTER DES COMPOSANTS REACT AVEC VITEST

- Configuration de @testing-library/react avec Vitest
- Rendu virtuel et simulation d'événements utilisateur
- Mocking de hooks, contexte et API
- Exercice : test d'un composant interactif avec gestion d'état

8 / TESTER DES COMPOSANTS VUE AVEC VITEST

- Introduction à @vue/test-utils
- Tester les props, slots et événements personnalisés
- Gestion des dépendances et du DOM virtuel
- Exercice : test d'un composant Vue avec interaction utilisateur

9 / OPTIMISATION, RAPPORTS ET DÉBOGAGE

- Mode watch et exécution conditionnelle (--watch, --changed)
- Générer des rapports de couverture (--coverage)
- Intégration avec ESLint, Prettier et Git Hooks
- Résolution de problèmes fréquents et stratégies de debugging

10 / ATELIER FINAL : MISE EN PRATIQUE COMPLÈTE

Inov Systems

- Mise en place d’un environnement de test complet
- Écriture de tests pour un module utilitaire et un composant React/Vue
- Simulation d’un appel API et test de rendu dynamique
- Analyse des performances et recommandations de bonnes pratiques

Approche pédagogique

- Support Ecrit et Projection
- Exposés Interactifs, Podcasts et Vidéos
- Brainstorming et Jeux de Rôle
- Cas Pratiques et Labs inclus pour leur impact opérationnel
- Test de Validation des Acquis des Connaissances

Prochaines dates programmées

06 au 08 Jan. 2027	Présentiel - Casablanca
10 au 12 Mars 2027	Présentiel - Casablanca
05 au 07 Mai 2027	Présentiel - Casablanca
30 Juin au 02 Juil. 2027	Présentiel - Casablanca
01 au 03 Sep. 2027	Présentiel - Casablanca

Autres dates possibles sur demande. Contactez-nous pour organiser une session intra-entreprise.

Innov Systems

Réservation & Renseignements

Téléphone : +212 522 247 210
Email : contact@innov-systems.com
Web : https//www.innov-systems.com

Innov Systems — Tous droits réservés