

UI Testing Pragmatique avec Cypress

Maîtriser les tests end-to-end avec Cypress

Le testing ou plus particulièrement le testing front-end est indéniablement dans le top 3 des sujets les plus complexes associés au développement d'applications web.

Face à cette complexité, il est tentant de renoncer mais à quel prix ? Certes, il est possible de développer une application sans tests unitaires mais est-ce réellement plus rapide ?

Qu'advient-il de la maintenabilité du code quelques mois ou à peine quelques semaines plus tard ?

Pourrons-nous refactoriser l'application et la déployer régulièrement en toute sérénité au fur et à mesure qu'elle se complexifie et qu'elle vieillit entre nos mains ? Qu'en est-il des tests end-to-end ? Sont-ils suffisants ?

Ce workshop présente à travers des exercices pratiques, des techniques pragmatiques de testing avec Cypress. Vous y découvrirez également comment écrire des tests compréhensibles, maintenables et surtout rentables.

Détails

- Code : DW-CYP
- Durée : 2 jours (14 heures)

Public

- Développeur web

Pré-requis

- Une expérience de testing sera bénéfique afin de bien saisir les bonnes pratiques et astuces exposées lors de cette formation

Objectifs

- Implémenter différents types de test pour vos applications Angular, React, VueJS ou sans framework avec Cypress
- Tester les interactions réseau
- Choisir le meilleur type de tests en fonction du contexte
- Découpler les tests du code afin de réduire les coûts de testing et de maintenance
- Implémenter des tests maintenables et human-readable

Programme

Introduction au testing

- Pourquoi tester ?
- Les différents types de test
- Les stratégies de testing : cornet de glace, pyramide vs rayon de miel

Cypress Core Features

- Développer eXperience
- Rapidité
- Debuggabilité
- Retry-ability
- Cohérence et stabilité

Premier Test

- Mise en place de Cypress
- Mise en place de Cypress avec Nx
- Exercice : Implémenter et lancer un premier test

Fondamentaux Cypress

- Les commandes
- Les assertions
- Exercice : Interagir avec le DOM
- Debugging
- Organisation des tests

- Exercice : Interagir avec les formulaires
- Variables & aliases

Cypress Network Testing

- Avantages et inconvénients des différentes techniques
- Exercice : Intercepter les requêtes
- Exercice : Utiliser des fixtures
- Exercice : Utiliser une sandbox

Retour vers le future

- Exercice : Jouer avec le timer

“Keeping Tests Healthy with DOM Distancing”

- Tests attributes
- Cypress Custom Commands
- Page objects
- Test Harnesses

Test Harnesses

- L'histoire des Test Harnesses
- Comment cela fonctionne
- Exercice : Utilisation des test harnesses
- Exercice : Implémentation de test harnesses

Cypress Component Testing

- Brisons les frontières de l'”isolated testing” avec Cypress Component Testing
- Exercice : Isoler et tester un composant avec Cypress
- Exercice : Réutiliser le test harness

Tester les Interactions

- Stubs & Spies
- Exercice : “Override” les dépendances avec l'injection de dépendance et les spies
- Exercice : Tester la communication à base de props, inputs

& outputs

Visual Regression Testing

- Tester la présentation
- Détecter les régressions visuelles avec Cypress Component Testing & Percy

Automatisation

- Configuration de la CI
- Reporters
- Cypress Dashboard
- Parallelization

Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages