

Plateforme Selenium 4.x, Développeur



Spoc pratique
Travaux pratiques
continus



Durée
3 jours



Votre animateur
Olivier Charles



Code Kertify : ASD

S'inscrire →

www.kertify.com/inscription

Développer ses tests automatisés avec Web Driver

Cette formation à vocation très opérationnelle vous permettra de mettre en place un environnement de test complet autour de la plateforme Selenium Web Driver et de développer un framework de tests métier à l'aide d'un des langages supportés par le projet Selenium (Java, C#, Python, Javascript).

Les objectifs de la formation

- ◆ Installer et exploiter la plateforme Selenium et ses composants
- ◆ Programmer les API Web driver pour l'exécution des tests
- ◆ Développer le framework "Keyword Driven Testing" et paramétrer les tests automatisés
- ◆ Intégrer des suites de tests automatisées dans les processus d'intégration et de validation

Participants

- ◆ Consultants automatisation de tests
- ◆ Automaticien de tests,
- ◆ Développeurs/testeurs,

Prérequis

- ◆ Bonne connaissance des processus de test
- ◆ Bonne connaissance d'un langage de programmation Orienté Objet

Le programme de notre formation

1. Mise en place de la plateforme de test, de l'environnement de développement

- ◆ Présentation des composants de la plateforme
- ◆ Les spécifications W3C de l'API WebDriver
- ◆ Mise en place d'une architecture complète pour exécuter des tests
- ◆ Mise en place de l'environnement de développement en Java
- ◆ Travaux pratiques. Installation de la plateforme et tests de bon fonctionnement des navigateurs.

2. Exploitation de la plateforme Selenium et de ses composants

- ◆ Structure générale d'un test métier automatisé. Application avec différents langages
- ◆ Utiliser Selenium Server pour lancer des tests sur une machine distante
- ◆ Utiliser Selenium Grid et TestNG pour exécuter des tests sur plusieurs machines en parallèle
- ◆ Travaux pratiques. Préparation, exécution et monitoring des suites de tests. Montage d'une grille de serveurs de tests existants.

3. API WebDriver : programmer efficacement les actions élémentaires de tests

- ◆ Reconnaissance de composants dans une page Web avec des sélecteurs CSS et XPath
- ◆ Interactions avec les composants (sélection, double-clique), avec les fenêtres de l'application et avec le navigateur
- ◆ Synchroniser le test, attendre un événement ou une condition explicite
- ◆ Capturer une image de l'écran en cas d'erreur ou à la demande.
- ◆ API complémentaires (Sikuli, OpenCV)
- ◆ Travaux pratiques. Programmer des actions élémentaires de tests.

4. Vers un framework "Keyword Driven testing"

- ◆ Frameworks de mots clés techniques
- ◆ Alimentation des tests avec des données externes
- ◆ Génération de rapports de tests
- ◆ Conception et développement d'une API de "Keyword driven testing"
- ◆ Référentiel de composants, patron "Page Object Model"
- ◆ Travaux pratiques. Application du Keyword Driven Testing. Gestion de l'obsolescence des mots clés.

5. Intégration et déploiement des tests automatisés

- ◆ Intégration dans Jenkins
- ◆ Intégration avec un référentiel de tests, TestLink ou Squash-TM
- ◆ Intégration avec un gestionnaire d'anomalies, Mantis ou Jira
- ◆ Déploiement des tests sur des conteneurs Docker
- ◆ Travaux pratiques. Mise en œuvre et paramétrage sous Jenkins. Démonstration des ponts vers Squash-TM et Mantis.

6. Synthèse et perspectives

- ◆ Les bonnes pratiques organisationnelles et techniques
- ◆ Mise en œuvre réussie chez des grands comptes : robot Framework et sa librairie Selenium
- ◆ Veille technologique : Katalon Studio Open Source
- ◆ Perspectives futures