

Robot Framework : automatiser les tests

Cours Pratique de 3 jours - 21h

Réf : RBF - Prix 2024 : 2 180€ HT

Robot Framework est un framework générique et open source qui permet d'automatiser facilement les tests d'acceptation. Cette formation vous donnera la maîtrise des bases de cette "trousse à outils", depuis son installation jusqu'à son intégration dans une solution d'intégration continue en passant par les rédactions et automatisations de tests.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Positionner Robot Framework dans l'écosystème des frameworks d'automatisation des tests

Comprendre les bases du fonctionnement de Robot Framework

Installer une plateforme de développement et d'exécution pour Robot Framework

Analyser un test par mot-clé

Assembler des mots-clés de Robot Framework pour concevoir et structurer un test

Organiser les ressources et utiliser les librairies

Utiliser les bibliothèques standard de Robot Framework

Comprendre l'intérêt du CI/CD et l'utilisation de Robot Framework avec GitLab-CI

TRAVAUX PRATIQUES

De nombreux travaux pratiques jalonnent cette formation.

LE PROGRAMME

dernière mise à jour : 11/2021

1) Introduction à l'automatisation

- Le projet d'automatisation.
- Les tests en démarche agile : peer programming, Test-Driven Development (TDD).
- La pratique du TDD.
- La pratique du BDD (Behaviour-Driven Development).
- Les principes du ATDD (Acceptance Test-Driven Development).
- Les frameworks d'automatisation (hybride, KDT...).
- KDT, introduction à Robot Framework.

2) Présentation et installation

- Philosophie de Robot Framework.
- Architecture et concepts.
- Librairies, librairies externes.
- Outils intégrés.
- Installation de Robot Framework dans un environnement Python.

Travaux pratiques : Installation de Python, RIDE, PyCharm/Visual Studio Code. Configuration d'un environnement virtuel Python. Installation de Robot Framework.

PARTICIPANTS

Testeurs/automaticiens, chefs de projets, concepteurs/développeurs.

PRÉREQUIS

Avoir une culture du test logiciel. Une expérience avec un langage de programmation tel que Python ou Java est conseillée.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

3) Syntaxe d'écriture des tests - Partie 1

- Organisation d'un test, vocabulaire et syntaxe, structure d'un test.
- Déclaration, portée et manipulation des variables (extension des variables Python).
- Exécution des tests depuis l'IDE, exécution des tests avec les options CLI, débogage des tests.
- Visualisation et interprétation des résultats.
- Visualisation des rapports de test dans Allure.

Travaux pratiques : Écrire des tests avec les mots-clés de Robot Framework. Exécuter les tests avec/sans les options de la ligne de commande. Produire et analyser les rapports de tests avec les mots-clés. Personnaliser la documentation dans les tests et les mots-clés avec les outils Robot Framework.

4) Syntaxe d'écriture des tests - Partie 2

- Test Setup, Test Teardown.
- Tag et arguments.
- Data-driven testing avec Robot Framework.
- Behavior-driven testing avec Robot Framework.
- Suite de tests.

Travaux pratiques : Modifier les tests pour inclure les concepts vus dans la partie 2.

5) Bibliothèques standards de Robot Framework

- BuiltIn, Collections, Screenshot, Process, Dialogs, OperatingSystem.
- Utilisation de ces bibliothèques dans les tests.

Travaux pratiques : Utiliser ces bibliothèques pour aller plus loin dans les tests avec Robot Framework.

6) Personnalisation des mots-clés dans Robot Framework

- Principe, syntaxe, paramètres.
- Utilisation.

Travaux pratiques : Créer des mots-clés personnalisés. Créer des bibliothèques de mots-clés. Créer des fichiers de ressources et des fichiers de variables.

7) Création de bibliothèques de tests

- Implémentation de bibliothèques en Python.

Travaux pratiques : Écrire des programmes Python implémentant de nouveaux mots-clés.

8) Fonctionnalités avancées

- Lancement des tests en parallèle dans Robot Framework.
- Post-processing.
- Bibliothèques tierces (Selenium, Appium).
- Test des API Rest.

Travaux pratiques : Implémenter ces fonctionnalités pour aller plus loin.

9) CI/CD, intégration/déploiement continu

- Enjeux stratégiques.
- Robot Framework en CI/CD.

LES DATES

CLASSE À DISTANCE
2024 : 15 juil., 28 oct.

PARIS
2024 : 08 juil., 21 oct.