

## Cucumber et l'automatisation des tests

Les projets informatiques s'orientent de plus en plus vers l'agilité et l'attente sur l'automatisation des tests devient de plus en plus forte au sein des équipes.

Acceptance Test Driven Development (ATDD), Behaviour Driven Development (BDD) sont les pratiques de test phares à déployer dans un contexte agile, notamment par la simplification de l'automatisation et son impact sur la robustesse du code développé.

Pour répondre à ces exigences pour l'organisation et les pratiques de l'automatisation, Cucumber devient une solution incontournable, à partir de son framework très populaire décliné sous toute sorte de plates-formes : Java, Ruby...

### 1. PUBLIC CONCERNE

---

- Testeurs fonctionnels « manuels » désireux de s'orienter vers l'automatisation de tests
- Développeurs désireux de s'orienter vers l'automatisation de tests
- Automaticiens de tests débutants

### 2. PREREQUIS

---

- Connaître les principes de l'automatisation des tests – cf. le module « Introduction à l'automatisation »
- Connaître ou pratiquer des automates de test et les langages de scripting.
- Avoir un minimum d'expérience en développement (Java...)

### 3. GAINS POUR LES PARTICIPANTS

---

- Acquérir les compétences pour être autonome sur la création d'un projet d'automatisation Cucumber/java
- Etre en mesure de coder proprement et simplement des suites de tests Cucumber/Java

### 4. DUREE

---

- 2 jours (14 heures)

### 5. REPARTITION

---

- Théorie : 30 %
- Pratique : 70 %

### 6. DEROULEMENT PEDAGOGIQUE

---

#### Introduction et rappels du contexte :

- La fiabilisation automatique des tests d'acceptance, le BDD, l'ATDD ; une demande croissante en contexte agile

- Les objectifs et avantages de Cucumber
- Les limites de Cucumber

### 1ère partie : les bases

- Installer son environnement de dev., avec ou sans Maven
- Créer le projet.
- Etude d'un cas pratique « *la cible commerciale* »
- Rédiger un script Gherkin
- Générer le glue code

### 2ème partie : scripter, tester

- Comprendre les expressions régulières (bases)
- Lancer Cucumber avec et sans le launcher JUnit
- Utiliser le Design Pattern Injection de dépendances avec PicoContainer
- Suite du cas pratique « *la cible commerciale* »

## 7. TRAVAUX PRATIQUES

---

- TP1 : Installer projet de test
- TP2 : Utiliser le langage Gherkin pour la construction des scripts de test
- TP3 : Lancer une suite de test
- TP4 : Créer et modifier du glue Code
- TP5 : Paramétrer un script
- TP6 : Utiliser un plan du scénario
- TP7 : Manipuler l'injection des dépendances
- TP8 : Manipuler annotations et hooks

Exercices avec exemples basés sur un cas pratique réel (cible commerciale d'audience de la télévision).

## 8. ORGANISATION DE LA FORMATION

---

### 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.

### Moyens pédagogiques et techniques

- Accueil des stagiaires dans une salle dédiée à la formation
- Documents supports de formation projetés.
- Descriptions des cas concrets
- PCs équipés des solutions outillées

### Dispositif de suivi de l'exécution de d'évaluation de la formation

- Feuilles de présence.
- Formulaire d'évaluation de la formation.