

TD3 - CDP - TESTS

Q1 - En exploitant le langage Gherkin, décrivez les scénarios de test permettant de valider les fonctionnalités relatives à la gestion des ateliers.

→ Voir les fichiers Gherkin dans le dossier “tests”

Q2 - Proposer un langage simple et minimal permettant d’automatiser les tests que vous avez rédigés (inspirez-vous des API permettant de manipuler les navigateurs web).

Pour automatiser nos tests nous pourrions avoir les instructions suivantes :

- get : récupère un élément avec son id / classe / type
- getAll : renvoie une liste de tous les éléments correspondants à un id / classe / type
- go : charge un URL spécifié
- click : permet de cliquer sur un bouton
- setValue : permet de modifier le contenu d’un élément (input, titre, etc...)
- create : crée un élément en fonction du type spécifié
- addChild : ajoute un élément à un autre élément spécifié

Q3 - Exploitez votre langage pour rédiger un test de validation (celui de la validation du scénario permettant la création d’un atelier).

```
document.go("/workshop");  
let nom = document.get(".name");  
let desc = document.get(".description");  
nom.setValue("CDP");  
desc.setValue("Initiation à la conduite de projet");  
let valider = document.get(".save");  
valider.click();  
document.go("/");  
let articles = document.getAll(".article");  
if(articles.length > 0){ SUCCES }  
else{ ERROR }
```

Q4 - Proposez une architecture permettant d’exécuter automatiquement les tests spécifiés avec votre langage. Discutez surtout du diagnostic qui sera rendu après exécution du test.

On pourrait imaginer dans notre architecture un dossier “tests” qui contient nos descriptions en Gherkin et un ou plusieurs fichiers de test automatique associé. Chaque test serait associé à un scénario Gherkin. On pourrait voir pour chaque fichier Gherkin combien de % de tests sont réussis.

Pour cela, il faut comme dans notre exemple de programme définir quand le test est un succès ou non.

Par exemple :

On lance la compilation de notre programme qui lance aussi les tests. Les tests sont validés ou non. Sachant qu’ils sont chacun associés à un scénario on peut donc faire des “statistiques” : combien de % de tests sont un succès pour tel Scénario, puis pour tel Feature, etc...

Cela permettrait de voir là où il reste le plus de travail (% le moins élevé) et là où il en reste le moins (% le plus élevé).