Mini projet GLG101
Framework automatisation des tests

Objectif

Avec un langage de votre choix parmi (Java, Javascript, Python, C# , PHP) et **l'API Selenium WebDriver en version > = 4.0**, vous devez développer un framework de tests automatisés permettant de faciliter l'implémentation de tests fonctionnels pour un public de testeurs fonctionnels très peu développeur.

Applications sous tests

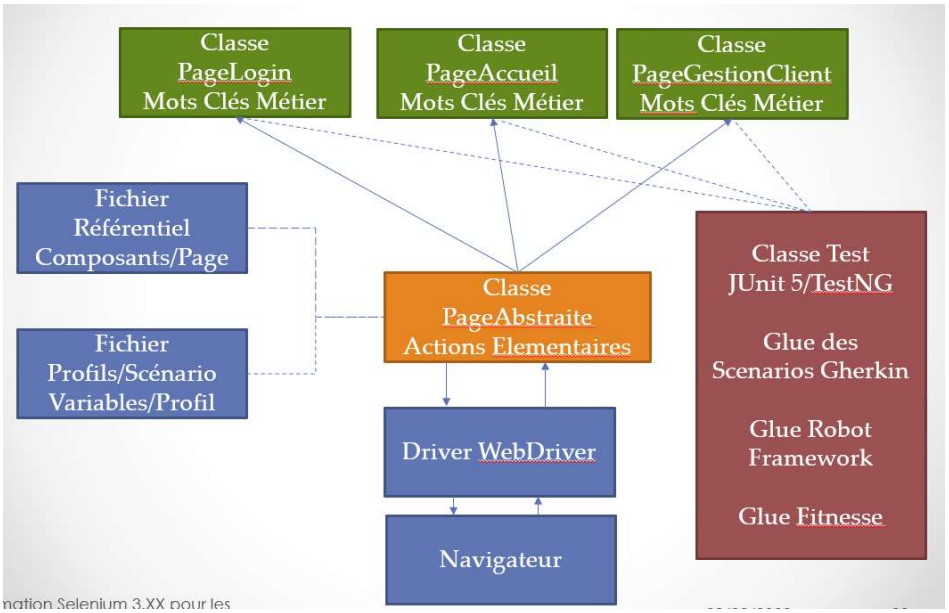
Vous illustrerez l'utilisation de votre framework sur une des deux applications web suivantes :

- https://todomvc.com/examples/react/#/
- https://computer-database.gatling.io/computers

Structure demandée

Votre framework devra permettre :

- De référencer, par leur nom et leur accesseur xpath, tous les composants utilisés dans vos tests dans un fichier (csv, json, ou à votre convenance) appelé référentiel de composants
    - Ex NumSecu : //input[@id='numsecu']
- De proposer un mécanisme pour injecter des données de test stockées dans des fichiers externes dans un format de votre choix (csv, json, excel …)
- De suivre l'architecture orientée objet suggérée par le schéma suivant :



Il faudra donc développer une classe abstraite permettant de faire des actions élémentaires (click , double click , saisie , sélection dans une liste …) sur les éléments de votre référentiel.

Pour chaque page ou état de votre application, vous développerez donc une **classe HERITANT** de votre classe abstraite et vous développerez des méthodes publiques de type « mots clé métier » (Action et Validation) réutilisant les actions élémentaires provenant de cette classe abstraite.

Exemples de méthodes :

- FiltrerListeOrdinateursAvec(String filtre) , ValiderPresenceOrdinateur(String Ordinateur) : boolean
- AjouterNouvelleTache(String todo), ValiderTacheTerminee(String nomTache) : boolean

On rappelle que les mots clé de validation sont utiles pour faire des tests, vous pourrez donc utiliser les bibliothèques d'assertions d'un xUnit ou bien retourner vrai ou faux pour ensuite utiliser une assertion dans un test xUnit

- assertTrue(pageGestionTodo.ValiderTacheTerminee('Faire le mini-projet GGLG101'))

Livrables demandés

On vous demande de fournir pour le lundi 5 juin 2022, par WeTransfer ou sur un référentiel public de type Git, à l'adresse mail suivante : olivier.charles@supavenir.fr, les éléments suivants :

- Une note de version expliquant l'objet du framework, son utilisation, la liste des mots clé d'actions et de validations, et les bugs restants éventuels
- Le code de votre framework avec au moins 10 mots clés (Actions + Vérifications)
- Le code des tests illustrant dans le framework xUnit (Au moins 5 tests, illustrant la totalité de vos mots clés)

Exemple de tests en java :

On admet que les paramètres référentiel et data correspondant à votre référentiel et à vos données de test sont des champs statiques de la classe de test préalablement initialisés dans une méthode @BeforeAll

On a un fichier de données data.csv structuré comme ci-dessous :

scenario	filtre	ordinateur	message
nominal	Apple	Apple Newton	13 computers found
inconnu	xxx	yyy	No computer

On pourra alors écrire les 2 tests suivants qui sont lisibles par des non développeurs.

```
@Test
public void testFiltrageNominalOrdinateurs() {
        data.scenario('nominal') ;
        String filtre = data.var('filtre'); // Apple
        String ordinateur = data.var('ordinateur') // Apple Newton
        String message = data.var('message') // 13 computers found
        PageRechercheOrdinateurs page = new PageRechercheOrdinateurs( referentiel , data ) ;
        page.filtrerListeOrdinateursAvec( filtre ) ;

        // On intègre l'assert JUnit dans les mots clé de validation
        page.validerMessageInformation( message ) ;
        page.validerAbsenceOrdinateur(  ordinateur )) ;
        page.naviguerPageSuivante() ;
        page.validerPresenceOrdinateur( ordinateur ));
}

@Test
public void testFiltrageOrdinateurInconnu() {
        data.scenario('inconnu') ;
        String filtre = data.var('filtre'); // xxx
        String ordinateur = data.var('ordinateur') // yyy
        String message = data.var('message') // No computer

        PageRechercheOrdinateurs page = new PageRechercheOrdinateurs( referentiel , data ) ;
        page.filtrerListeOrdinateursAvec( filtre ) ;

        page.validerMessageInformation( message ) ;
        page.validerAbsenceOrdinateur(  ordinateur )) ;

}
```

Bonus :

On vous demande d'écrire un scénario Gherkin et son code « glue » pour illustrer l'utilisation de vos mots clé sans passer par un xUnit.

Evaluation :

Il sera tenu compte de la conformité avec la structure demandée, les livrables demandés et la facilité d'utilisation de votre framework. L'évaluation portera un poids équivalent à l'examen dans l'évaluation finale du module GLG101

Remarque :

Si vous avez un besoin potentiel en tests fonctionnels web automatisés dans votre entreprise, vous pourrez réutiliser la partie référentiel, données et page abstraite de votre framework. Choisissez donc bien votre langage d'implémentation.

Pour toutes questions complémentaires : olivier.charles@supavenir.fr (objet : Projet GLG101)