

DESCRIPTION D'UNE SITUATION PROFESSIONNELLE

PARCOURS	SISR <input checked="" type="checkbox"/>	SLAM <input type="checkbox"/>
-----------------	---	--------------------------------------

Lieu de réalisation	Coordonnées entreprise ou centre de formation	Logo
Période de réalisation	Du : 08/06/2022	Au : 15/06/2022
Modalité de réalisation	SEUL <input checked="" type="checkbox"/>	EN EQUIPE <input type="checkbox"/>

Intitulé de la mission	Paramétrage d'un testeur d'applications cypress
Description du contexte de la mission	La mission a pour but de mettre en place un testeur pour les applications métiers

Contraintes & Résultat	Ressources fournies / contraintes techniques / Résultats attendu
	Choix des moyens de réalisation libre. Le testeur doit être généraliste pour toutes les applications.
Productions associées	Liste des documents produits et description
	X

Modalités d'accès aux productions	Identifiants, mots de passe, URL d'un espace de stockage et présentation de l'organisation du stockage
	<p style="text-align: center;"> https://rocksurvie.github.io/ https://github.com/rocksurvie/documentsBTS </p> <p>Les documents sont placés dans des répertoires distincts en fonction du lieu de réalisation et sont classés par année pour les missions réalisées en entreprise.</p>

Description détaillée de la situation professionnelle retenue et des productions réalisées
en mettant en évidence la démarche suivie, les méthodes et les techniques utilisées
<p>Cypress est un framework JS de tests end-to-end. C'est un outil open source permettant de mettre facilement en place ces tests d'applications</p> <h3>Installation</h3> <p>Dans un premier temps, télécharger le zip de cypress sur le site officiel. Se positionner ensuite dans un répertoire. Cloner le repository git</p>

```
git clone liengit
```

lancer cypress avec la commande :

```
./Cypress
```

Sélectionner dans le dossier que vous avez créé le projet souhaité

Sélectionner e2e

à savoir sur l'installation

En complément de l'installation standard, il est nécessaire de rajouter un bout de code pour accéder à l'iframe.

Ce bout de code doit être ajouté dans support/commands.js ce qui est fait dans le fichier commiter dans le projet git.

```
// support/commands.js
```

```
Cypress.Commands.add('iframe', { prevSubject: 'element' }, $iframe => {  
  return new Cypress.Promise(resolve => {  
    $iframe.ready(function() {  
      resolve($iframe.contents().find('body'));  
    });  
  });  
});
```

Cypress permet d'enregistrer graphiquement les tests qui seront rejoués à la demande.

Il est important d'écrire les cahiers de tests en fonction de l'application et du paramétrage de celle-ci.

Il faut écrire des tests significatifs, il ne faut pas seulement augmenter la couverture du code, mais tester la vraie fonctionnalité, sinon c'est une perte de temps.

L'interface graphique

cliquer sur e2e

Sélectionner son test

Le test est lancé.

Les informations à savoir :

pour se rendre à la page du site

```
cy.visit('lien application')
```

lorsque l'on change de page toujours tester l'url:

```
cy.url().should('include', '/listeDossier') // Test de l'URL
```

le fichier "fixtures" permet de placer les fichiers de paramétrage JSON

le fichier "e2e" contient les tests

Ligne de code utile

cliquer sur un bouton avec id et sans

```
cy.get('#cypress_bouton_login').click()
```

```
cy.contains('Valider').click()
```

cliquer sur un sous menu

```
cy.contains('Nouveau dossier').click() // Clic sur le menu principal pour afficher le sous-menu
```

```
cy.get('ul.nav-sub').find('li').contains('C1021B').click({force:true}) // Clic sur l'élément de sous-menu souhaité
```

insérer une information

```
cy.get('#SimulAssure_0_frais_intermediation').type('10')
```

parcourir une iframe

```
cy.get("iframe").iframe().contains("Suivant").click()
```