

Onboarding Cypress

1) Qu'est-ce que Cypress ?

Cypress est un framework de test Javascript en E2E (end to end) qui a été pensé pour concurrencer Selenium mais qui permet d'écrire des tests plus vite (de façon très intuitive).

2) Pourquoi être passé sur Cypress ?

Nous passons sur Cypress afin de réduire le temps de la création des tests comparé à Selenium, de les dé-isoler des devs mais aussi de pouvoir les utiliser continuellement sur tous les environnements

3) Comment ?

En formatant l'ancien Sanity check (s'en inspirer) afin de l'optimiser (ce qui peut passer par refaire des process qui étaient un peu plus "bricolés", en utilisant pleinement cucumber afin de créer le moins de phrases possible mais les plus simples afin ne pas se perdre dans un dictionnaire trop long

4) Ce que l'on a à gagner ?

Un sanity check qui est plus simple à lire pour l'arrivée potentielle de nouveaux testeurs, mais aussi pour les anciens qui pourront ajouter des cas et le maintenir plus simplement

Et enfin, un sanity check qu'il sera possible de lancer plus souvent afin de pouvoir avoir des tests d'utilisation basique de l'application le plus souvent possible pour laisser passer le moins de bugs possible.

5) Le set-up de cypress

Tout d'abord pour utiliser cypress tu auras besoin de nodejs dans la version 16

<https://nodejs.org/en/blog/release/v16.16.0/>

Il faut ensuite installer yarn avec la commande : `npm install --global yarn`

Puis tu pourras faire la commande **yarn add cypress --save-dev** dans le dossier pasq-twistr

Il te faudra aussi git

Installation de git sous windows :

<https://git-scm.com/download/win>

Afin de récupérer le fichier il faut aller sur le github de twistr-front:

<https://github.com/adeo/pasq-twistr>

Installation de git sous Linux :

`$sudo apt install git-all`

Cliquer sur "code" et choisir ssh, copier la clef ssh qu'il vous est donnée puis utiliser la commande

git clone <Clef ssh>

Une fois tout ça fait, en allant dans le fichier e2e avec la commande :

cd /e2e/

tu pourras ouvrir l'UI de cypress avec la commande

yarn cypress open

6) Utilisation de Cucumber et Cypress

Cucumber permet d'écrire des tests de façon un peu plus "humaine" afin qu'il soient plus compréhensibles et simple à retenir

un phrase comme "I am authenticated with profile "raf" in "uat" env" est une phrase générique qui explicite que l'on veut se connecter sur un environnement et l'on passe le profil souhaité et l'environnement souhaité entre guillemets (façon de passer des arguments avec cucumber)

du côté cypress, la fonction traitant la phrase ressemble à cela :

```
When("I am authenticated with profile {string} in {string} env", (type: string, environnement: string) => {
  cy.visit(`https://twistr.${environnement}.adeo.com/`);
  env = environnement;
  cy.get("[id=username]").invoke("attr", "value", Cypress.env(`${type}_id`));
  cy.get("[id=my_password]").invoke("attr", "value", Cypress.env(`${type}_password`));
  cy.get("[id=my_sign_on_button]").click();
  cy.url().should("include", "twistr.uat.adeo.com");
});
```

Nous avons la déclaration de la phrase qui vient activer la fonction dans notre cas "I am authenticated with profile {string} in {string} env" suivi de la déclaration du type des variables type (la première de la phrase en partant de la gauche) puis environnement (la deuxième)

les commandes cypress commencent toujours avec "cy" puis un point avant d'utiliser la procédure ou fonction que l'on souhaite utiliser

voici les procédures les plus utilisées en cypress :

cy.get() : cette procédure prend une string en paramètre et nous permet de choisir ce le composant que nous souhaitons sélectionner par exemple :

cy.get("[id=username]") sélectionne l'élément ayant pour id "username"

cy.visit() qui prend une **string** en argument et qui nous **redirige vers l'url** donnée en argument

cy.type() qui prend une **string** comme argument et remplit le champs sélectionné avec l'argument passé (Il permet aussi d'effectuer des actions comme appuyer sur entrer avec cy.type({enter}))

Liste des arguments spéciaux qui peuvent être passés dedans :

<https://docs.cypress.io/api/commands/type#Arguments>

cy.click() afin de **cliquer** sur l'élément sélectionné

cy.contains() qui prend une **string** en argument et sélectionne l'élément qui a pour value l'argument donné

cy.wait() qui prend un **int** en argument et met en pause le test pour le nombre de millisecondes données en argument

7) Liens vers les docs

Liens vers la doc officielle de cypress :

<https://docs.cypress.io/guides/overview/why-cypress>

Liens vers la doc officielle de Cucumber :

<https://cucumber.io/docs/installation/>

Liens vers une doc Cucumber simplifiée et en français :

<https://jira.adeo.com/confluence/display/TWI/Cucumber>

Liens vers le Dico de données :

<https://jira.adeo.com/confluence/pages/viewpage.action?pageId=285262405>