



CYPRESS

EL YASSINI SAFOUAT
EL BOUJADAINI MOHAMED AMYN



CONTENT



01

DEFINITION

02

LES COMMANDES DE CYPRESS

03

LAB-1

04

LAB-2

05

CONCLUSION

CYPRESS

CYPRESS EST UN FRAMEWORK OPEN-SOURCE DE TEST END-TO-END
(E2E) CONCU POUR FACILITER LE TEST AUTOMATISÉ DES
APPLICATIONS WEB MODERNES. CYPRESS SIMPLIFIE LE TDD.
IL EST ÉCRIT EN JAVASCRIPT ET PRINCIPALEMENT UTILISÉ POUR
TESTER DES APPLICATIONS FRONTALES.

TDD

Traditionnellement, les développeurs créent d'abord leur application, puis écrivent les tests. Avec le TDD, le processus est inversé. Les tests sont écrits avant même que le code de l'application ne soit développé. Cela peut sembler à l'envers, mais cela présente de nombreux avantages.

Imaginons que le testeur connaît déjà les noms de classe que le développeur front-end utilisera.. Grâce à Cypress, le testeur peut anticiper les besoins du développeur et rendre le processus plus fluide.

LES HOOKS ET LES CONTEXTES

- `describe` : Utilisé pour regrouper plusieurs tests liés dans une suite de tests. Fournit un contexte pour organiser et structurer les tests.
- `it` : Définit un test individuel ou une spécification à l'intérieur d'une suite de tests. Représente un scénario de test spécifique.
- `beforeEach` : Exécute du code avant chaque test dans une suite de tests. Utile pour effectuer des actions de configuration préalable nécessaires à l'exécution des tests.
- `afterEach` : Exécute du code après chaque test dans une suite de tests. Utile pour effectuer des actions de nettoyage ou de remise à zéro après l'exécution des tests.

2-Browser Commands

cy.visit Navigue vers l'URL spécifiée.

cy.go("back"): Reviens en arrière dans l'historique de navigation.

cy.go("forward"): Avance dans l'historique de navigation.

cy.reload Recharge la page actuelle.

cy.viewport: Permet de définir ou d'obtenir la taille de la fenêtre d'affichage.

cy.url: Obtient l'URL de la page en cours.

cy.window: Obtient la référence à la fenêtre globale du navigateur.

cy.title Obtient le titre de la page.

.scrollIntoView: Fait défiler la page jusqu'à l'élément spécifié pour le rendre visible dans la fenêtre d'affichage.

.scrollTo: Fait défiler la page jusqu'à la position spécifiée.

LES COMMANDES DE CYPRESS:

LES COMMANDES DE SELECTION

`cy.get(selector)` : Sélectionne un ou plusieurs éléments du DOM en utilisant un sélecteur.

`cy.contains(content)` : Sélectionne un élément qui contient le texte spécifié.

`.first` : Sélectionne le premier élément d'une liste d'éléments.

`.last` : Sélectionne le dernier élément d'une liste d'éléments.

`.next` : Sélectionne l'élément suivant dans la liste d'éléments parents.

`.prev` : Sélectionne l'élément précédent dans la liste d'éléments parents.

`.parents` : Sélectionne tous les parents de l'élément.

`.children` : Sélectionne tous les enfants de l'élément.

`.siblings` : Sélectionne tous les frères et sœurs de l'élément.

`.closest` : Sélectionne l'ancêtre le plus proche qui correspond au sélecteur spécifié.

`.find` : Sélectionne les descendants de l'élément qui correspondent au sélecteur

LES COMMANDES D'ACTION

- click() : Déclenche un événement de clic sur l'élément.
- .dblclick() : Déclenche un événement de double-clic sur l'élément.
- .rightclick() : Déclenche un événement de clic droit sur l'élément.
- .type("foo") : Saisit le texte spécifié dans un champ de saisie.
- .clear() : Efface le contenu d'un champ de saisie.
- .check() : Coche une case à cocher.
- .unchecked() : Décoche une case à cocher.
- .focus() : Donne le focus à l'élément.
- .blur() : Enlève le focus de l'élément.
- .submit() : Soumet un formulaire.
- .trigger('click') : Déclenche manuellement un événement de clic sur l'élément.
- .hover() : Simule le survol de la souris sur l'élément.
- .select() : Sélectionne une option dans un élément <select>.

LES COMMANDES DE CYPRESS:

LES COMMANDES D'ASSERTATION

- .SHOULD() : COMMANDE UTILISÉE POUR APPLIQUER DES ASSERTIONS SUR UN ÉLÉMENT OU UN OBJET CYPRESS. ELLE EST SOUVENT UTILISÉE POUR DÉFINIR DES CONDITIONS QUE L'ÉLÉMENT DOIT SATISFAIRE.
- .AND() : UTILISÉE POUR CHÂÎNER PLUSIEURS ASSERTIONS ENSEMBLE. ELLE PERMET D'AJOUTER DES CONDITIONS SUPPLÉMENTAIRES À UNE ASSERTION.
- .EXPECT() : UNE AUTRE MANIÈRE DE FAIRE DES ASSERTIONS. ELLE EST SOUVENT UTILISÉE AVEC DES CHÂÎNES DE CARACTÈRES ET DES EXPRESSIONS RÉGULIÈRES POUR VALIDER LE CONTENU D'UN ÉLÉMENT.
- .THEN() : PERMET D'AJOUTER DES ACTIONS ASYNCHRONES OU DES OPÉRATIONS SPÉCIFIQUES À LA CHÂÎNE DE COMMANDES. ELLE EST SOUVENT UTILISÉE AVEC UNE FONCTION DE RAPPEL POUR EXÉCUTER DES ACTIONS APRÈS LA RÉOLUTION D'UNE COMMANDE PRÉCÉDENTE.

Autre Commande

`cy.log(message)` : Affiche un message dans la console de Cypress.

`cy.fixture(fixtureName)` : Charge un fichier de données (fixture) pour une utilisation dans les tests.

`.then(callback)` : Permet d'effectuer des actions asynchrones dans la chaîne de commandes, en utilisant une fonction de rappel.

`.wait(time)` : Attend le nombre spécifié de millisecondes avant de poursuivre l'exécution des commandes suivantes.

`.pause()` : Met en pause l'exécution des commandes pour permettre le débogage.

`.screenshot(name)` : Capture une capture d'écran de la vue actuelle et l'enregistre sous le nom spécifié.

`.debug()` : Met en pause l'exécution des commandes et ouvre le navigateur Cypress dans le mode débogage interactif.

`.on`: écouter des événements

VOIR LA VIDEO POUR L'INSTALATION ET POUR 2 LAB