

1.

AZANIR / cypressAllure Public

Watch 1

Fork 10

Star 1

Code Issues Pull requests Actions Projects Security Insights

bb313ea73c cypressAllure / cypress / e2e / pages / loginPage.ts / <> Jump to

Go to file

...

AZANIR update test and add allure

Latest commit de5827a on Jul 15 History

1 contributor

28 lines (23 sloc) 879 Bytes

Raw

Blame



```

1  /// <reference types="cypress" />
2
3  class LoginPage {
4    get signInLink() { return cy.get('.login') }
5    get emailAddressTxt() { return cy.get('#email') }
6    get passwordTxt() { return cy.get('#passwd') }
7    get signInBtn() { return cy.get('#SubmitLogin') }
8    get alertBox() { return cy.get('p:contains("error")') }
9    get alertMessage() { return cy.get('.alert-danger > ol > li') }
10
11   public launchApplication() {
12     cy.visit('/')
13   }
14
15   public login(emailId: string, password: string) {
16     this.signInLink.click()
17     this.emailAddressTxt.type(emailId)
18     this.passwordTxt.type(password)
19     this.signInBtn.click()
20   }
21 }

```

In my opinion the signInLink receiver should be in a separate "MainPage" class in the "pages" folder

emailAdressField(not a "Txt") means exactly the field in which the e-mail fits and passwordField(not a "passwordTxt")

It is better to add "BaseUrl" to the config file

It is best to put it in a separate class in a folder "helper"

2.

```

1  import { LoginPage } from '../pages/loginPage'
2  import { myAccountPage } from '../pages/myAccountPage'
3
4  describe('Login Functionality', () => {
5    beforeEach(() => {
6      LoginPage.launchApplication()
7      cy.fixture('users.json').then(function (data) {
8        this.data = data;
9      })
10   })
11   it('login with valid credentials', function () {
12     LoginPage.login("testautomation@cypress.test.com", "Test@1234")
13     myAccountPage.validateSuccessfulLogin()
14     myAccountPage.logout()
15     myAccountPage.validateSuccessfulLogout()
16   })
17   it('login with valid credentials read data from fixture', function () {
18     LoginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
19     myAccountPage.validateSuccessfulLogin()
20     myAccountPage.logout()
21     myAccountPage.validateSuccessfulLogout()
22   })
23   it('login with invalid email credentials read data from fixture', function () {
24     LoginPage.login(this.data.invalid_credentials.invalid_email.emailId,
25       this.data.invalid_credentials.invalid_email.password)
26     LoginPage.validateLoginError('Authentication failed.')
27   })
28   it('login with invalid password credentials read data from fixture', function () {
29     LoginPage.login(this.data.invalid_credentials.invalid_password.emailId,
30       this.data.invalid_credentials.invalid_password.password)
31     LoginPage.validateLoginError('Authentication failed.')
32   })
33   it('login with wrong email format credentials read data from fixture', function () {
34     LoginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
35     LoginPage.validateLoginError('Invalid email addressssss.')
36   })
37 })

```

In this test, the description does not indicate that it is also checked and the possibility of logging out of the account. It would be better to create Another test to test this functionality

It is better to remove this part of the test description because it does not need to be specified

This check has the following code: this.alertMessage.should('have.text', errorMessage) it would be better to replace it with the following code: this.alertMessage.should('contain', errorMessage) after that the check might look like this: LoginPage.validateLoginError('Invalid email')