



import { Page, Locator } from '@playwright/test';

/\*\*

\* Page Object for Login Page

\*/

export class LoginPage {

readonly page: Page;

/\*\*

\* Locator for username input field

\*/

readonly usernameInput: Locator;

/\*\*

\* Locator for password input field

\*/

readonly passwordInput: Locator;

/\*\*

\* Locator for login button

\*/

readonly loginButton: Locator;

/\*\*

\* Locator for remember me checkbox

\*/

readonly rememberMeCheckbox: Locator;

/\*\*

\* Locator for forgot password link

\*/

readonly forgotPasswordLink: Locator;

/\*\*

\* Locator for use custom domain link

\*/

readonly useCustomDomainLink: Locator;

/\*\*

\* Locator for custom domain input field

\*/

readonly customDomainInput: Locator;

/\*\*

\* Locator for custom domain continue button

\*/

readonly customDomainContinueButton: Locator;

/\*\*

\* Locator for custom domain back button

\*/

readonly customDomainBackButton: Locator;

/\*\*

\* Locator for chooser error message

\*/

readonly chooserError: Locator;

/\*\*

\* Locator for chooser label

\*/

readonly chooserLabel: Locator;

/\*\*

\* Locator for edit link

\*/

readonly editLink: Locator;

/\*\*

\* Locator for id list

\*/

readonly idList: Locator;

/\*\*

\* Locator for save edit button

\*/

readonly saveEditButton: Locator;

/\*\*

\* Locator for cancel edit button

\*/

readonly cancelEditButton: Locator;

/\*\*

\* Initializes the page object with the given page

\* @param page The page to initialize the page object with

\*/

constructor(page: Page) {

this.page = page;

this.usernameInput = page.locator('#username');

this.passwordInput = page.locator('#password');

this.loginButton = page.locator('#Login');

this.rememberMeCheckbox = page.locator('#rememberUn');

this.forgotPasswordLink = page.locator('#forgot\_password\_link');

this.useCustomDomainLink = page.locator('#mydomainLink');

this.customDomainInput = page.locator('#mydomain');

this.customDomainContinueButton = page.locator('#mydomainContinue');

this.customDomainBackButton = page.locator('#hint\_back\_domain');

this.chooserError = page.locator('#chooser\_error');

this.chooserLabel = page.locator('#chooser\_label');

this.editLink = page.locator('#edit');

this.idList = page.locator('#idlist');

this.saveEditButton = page.locator('#hint\_save\_edit');

this.cancelEditButton = page.locator('#hint\_back\_edit');

}

/\*\*

\* Enters the username

\* @param username The username to enter

\*/

async enterUsername(username: string): Promise<void> {

await this.usernameInput.fill(username);

}

/\*\*

\* Enters the password

\* @param password The password to enter

\*/

async enterPassword(password: string): Promise<void> {

await this.passwordInput.fill(password);

}

/\*\*

\* Clicks the login button

\*/

async clickLoginButton(): Promise<void> {

await this.loginButton.click();

}

/\*\*

\* Checks the remember me checkbox

\*/

async checkRememberMe(): Promise<void> {

await this.rememberMeCheckbox.check();

}

/\*\*

\* Unchecks the remember me checkbox

\*/

async uncheckRememberMe(): Promise<void> {

await this.rememberMeCheckbox.uncheck();

}

/\*\*

\* Clicks the forgot password link

\*/

async clickForgotPasswordLink(): Promise<void> {

await this.forgotPasswordLink.click();

}

/\*\*

\* Clicks the use custom domain link

\*/

async clickUseCustomDomainLink(): Promise<void> {

await this.useCustomDomainLink.click();

}

/\*\*

\* Enters the custom domain

\* @param domain The custom domain to enter

\*/

async enterCustomDomain(domain: string): Promise<void> {

await this.customDomainInput.fill(domain);

}

/\*\*

\* Clicks the custom domain continue button

\*/

async clickCustomDomainContinueButton(): Promise<void> {

await this.customDomainContinueButton.click();

}

/\*\*

\* Clicks the custom domain back button

\*/

async clickCustomDomainBackButton(): Promise<void> {

await this.customDomainBackButton.click();

}

/\*\*

\* Gets the chooser error message

\*/

async getChooserError(): Promise<string> {

return await this.chooserError.textContent();

}

/\*\*

\* Gets the chooser label

\*/

async getChooserLabel(): Promise<string> {

return await this.chooserLabel.textContent();

}

/\*\*

\* Clicks the edit link

\*/

async clickEditLink(): Promise<void> {

await this.editLink.click();

}

/\*\*

\* Gets the id list

\*/

async getIdList(): Promise<string[]> {

return await this.idList.allTextContents();

}

/\*\*

\* Clicks the save edit button

\*/

async clickSaveEditButton(): Promise<void> {

await this.saveEditButton.click();

}

/\*\*

\* Clicks the cancel edit button

\*/

async clickCancelEditButton(): Promise<void> {

await this.cancelEditButton.click();

}

}