

PlaywrightOnPowerPlatform

This repository contains end-to-end tests for a TodoMVC demo app using [Playwright](#).

Getting Started

1. Clone the Repository

```
git clone https://github.com/itweedie/playwrightOnPowerPlatform.git
cd playwrightOnPowerPlatform
```

2. Install Dependencies

```
npm install
```

3. Install Playwright Browsers

```
npx playwright install --with-deps
```

Setting Up Environment Variables

To securely manage sensitive information like URLs, usernames, and passwords, create a `.env` file in the root of your project.

Steps to Create the `.env` File

1. Create a file named `.env` in the root directory of your project.
2. Add the following variables to the `.env` file:

```
APP_URL=https://techtweedie.crm11.dynamics.com/main.aspx?appid=6653f9fc-b74b-f011-877a-6045bd0e2fc6
APP_NAME=MDA Playwright Testing
0365_USERNAME=
0365_PASSWORD=
0365_TENANT_ID=
```

Using the `.env` File

The `.env` file is automatically loaded by the `dotenv` package, which is included in the project. These variables can be accessed in your test files using `process.env`. For example:

```
const config = {
  appUrl: process.env.APP_URL || 'default_url',
  appName: process.env.APP_NAME || 'default_name',
  username: process.env.O365_USERNAME || 'default_username',
  password: process.env.O365_PASSWORD || 'default_password',
  tenantId: process.env.O365_TENANT_ID || 'default_tenant_id',
};
```

Important Note

Make sure the `.env` file is added to your `.gitignore` to prevent sensitive information from being committed to your repository:

```
.env
```

For more details on environment variables, see [dotenv documentation](#).

Running Tests

Run All Tests (Headless by Default)

```
npx playwright test
```

Run Tests in Non-Headless Mode

```
npx playwright test --headed
```

Run a Specific Test File

```
npx playwright test tests/example.spec.ts
```

To run a set of test files from different directories, pass in the names of the directories that you want to run the tests in.

To test Portal Locally

```
npx playwright test tests-portal
```

To test MDA Locally

```
npx playwright test tests-mda
```

Run Tests in a Specific Browser

```
npx playwright test --project=chromium  
npx playwright test --project=firefox  
npx playwright test --project=webkit
```

Viewing Test Reports

After running tests, view the HTML report:

```
npx playwright show-report
```

Or open the generated report directly:

- <playwright-report/index.html>

Recording New Tests

You can use Playwright's codegen tool to record new tests:

```
npx playwright codegen https://demo.playwright.dev/todomvc
```

This will open a browser window and generate code as you interact with the page. Save the generated code in the `tests` or `tests-examples` directory.

Running Tests in CI

Tests are automatically run in GitHub Actions on push and pull requests to `main` or `master` ([github/workflows/playwright.yml](https://github.com/playwright-community/playwright/blob/main/.github/workflows/playwright.yml)).

For more details, see [Playwright documentation](https://playwright.dev/docs/running-in-ci).

[https://fakerapi.it/api/v2/custom?](https://fakerapi.it/api/v2/custom?_quantity=1&customfield1=name&customfield2=dateTime&customfield3=phone&something=longText)

[_quantity=1&customfield1=name&customfield2=dateTime&customfield3=phone&something=longText](https://fakerapi.it/api/v2/custom?_quantity=1&customfield1=name&customfield2=dateTime&customfield3=phone&something=longText)