

Version 2.1

Process_Codage

Page Object Model framework

POM is a design pattern that creates an object repository for storing all web elements. It is useful in reducing code duplication and improves test case maintenance. It provides flexibility to writing code by splitting into different classes and also keeps the test scripts separate from the locators.

In the Page Object Model, consider each web page of an application as a class file. Each class file will contain only corresponding web page elements. Testers should import those pages to their test classes. Using these elements, they can test and perform operations on the website under test. Considering this few of the important benefits of the Page Object Model are:

- **Code reusability** : The same page class can be used in different tests and all the locators and their methods can be re-used across various test cases.
- **Code maintainability** : There is a clean separation between test code which can be our functional scenarios and page-specific code such as locators and methods. So, if some structural change happens on the web page, it will just impact the page object and will not have any impacts on the test scripts.

Page Object Model in Cypress

Cypress also provides the flexibility to implement the automation framework using the Page Object Pattern.

For the implementation part of the Page Object Model, as a first step, we will need to create a PageClass, it's a class which contains web element's locators and methods to interact with web elements. To achieve this, create a new file (class) under `Cypress/Cypress/pageobjects` directory that will be used in the test files.

For each page created a `goToPage()` function must be coded.

The reason for not creating the page classes under Integration folder is because when you run your test cases, it runs all the test cases present under Integration folder and your page classes have no test cases so it skips these files and then it is shown as Skipped in the report which might confuse people. So its better to keep it somewhere else.

Filename and directory

Filename and directory must use PascalCase

Tree structure

The tree structure is the one you see when browsing the pages. For each page displayed and logical component a file is created.

```
+ pageobjects
--+ HeaderMenu.ts
--+ Settings
----+ SettingsMenu.ts
----+ Organisation
-----+ Users
-----+ UsersPage.ts
```

File naming policy

The classes added in `Cypress/Cypress/pageobjects` must follow this naming rules:

- For pages : xxxPage.ts
- For Menus xxxMenu.ts

Classes, methods and variables

- The classes name must be written in PascalCase.
- The methods and variables name must be written in camelCase.
- The function names should be explicit as much as possible.

Liens utiles

Best Practices Cypress