

## Playwright TypeScript POM Framework – Quick Guide

### 1. Overview

- **Framework Type:** Page Object Model (POM) using **Playwright + TypeScript**
  - **Key Features:**
    - Page classes for actions & self-verifying assertions
    - **PageManager** with **lazy loading** of pages
    - **Custom fixture (pm)** injected into tests
    - **HelperBase** provides logging & utility methods
- 

### 2. Folder Structure

```
bash

root/
|
├ tests/           # All test/spec files
├ page-objects/   # Page classes (DatepickerPage, FormLayoutsPage)
├ managers/        # PageManager class (Lazy Loads pages)
├ fixtures/        # customFixtures.ts (injects pm)
├ helpers/         # HelperBase.ts (Logging & utilities)
├ package.json
└ tsconfig.json
```

#### Diagram – Folder Structure

```
css

tests/
page-objects/
├ DatepickerPage
├ FormLayoutsPage
└ NavigationPage
managers/
└ PageManager
fixtures/
└ customFixtures
helpers/
└ HelperBase
```

---

### 3. How Components Work

#### Page Objects

- Extend **HelperBase**
- Contain **actions + internal expect assertions**
- Example:

```
await pm.onDatepickerPage().selectDateInCommonDatePicker(100)
```

#### HelperBase

- Provides utilities:
  - `waitForPageReady()`, `expectVisible()`, `getText()`, `within()`

- Logging: `logInfo()`, `logError()`

## PageManager

- Lazy loads pages **only when requested**
- Example:

```
pm.navigateTo().formLayoutsPage()
```

```
pm.onFormLayoutsPage().submitInlineFormWithNameEmailAndCheckbox('test', 'test@test.com', true)
```

## Custom Fixture

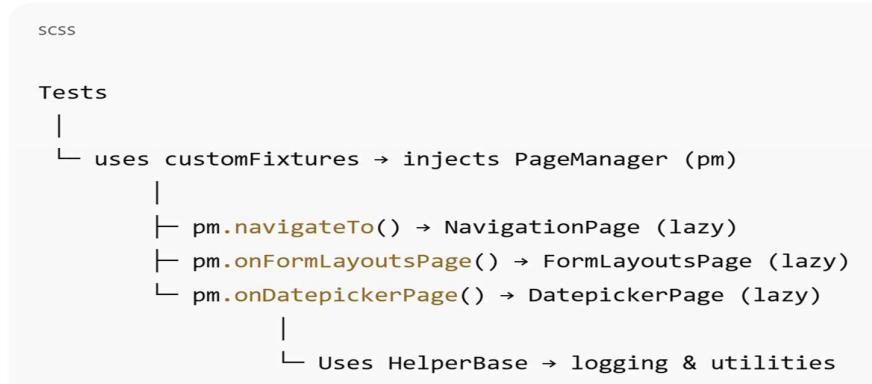
- `customFixtures.ts` injects `PageManager (pm)` into tests:

```
test('Test Example', async ({ pm }) => { ... })
```

---

## 4. Test Flow

### Diagram – Component Flow



### Flow Summary:

1. Test calls `pm` from fixture
  2. `PageManager` lazy loads requested page
  3. Page methods perform actions & internal expect checks
  4. `HelperBase` logs actions and provides utility methods
- 

## 5. How to Run

```
npm install
```

```
npx playwright test
```

```
npx playwright show-report # optional
```

---

## 6. Guidelines

- **Do not put expect in tests** — use page methods
  - **Always access pages via PageManager (pm)**
  - **Use logging from HelperBase** to understand flows
  - **Keep page objects focused** on actions + verifications
- 

 This guide gives a new engineer all they need to understand the framework, folder structure, flow, and how to write tests.