📘 Locator & Framework Usage Guide

A comprehensive guide to creating and using locators, handling static data, and working with page objects and tests in the Playwright POM Framework.

🗂️ Framework Folder Structure Overview

```
/project-root
├── config
│   └── qa.config.ts
├── data
│   └── admin
│       ├── faqs.json
│       ├── generate.json
│       ├── testData.json
│       └── users.json
├── locators
│   └── admin
│       ├── AdminPageLocators.ts
│       ├── AgreementsPageLocators.ts
│       ├── CommonLocators.ts
│       ├── ContentPageLocators.ts
│       ├── FaqsPageLocators.ts
│       ├── SchedulePageLocators.ts
│       └── ...
│   └── index.ts
├── pages
│   └── admin
│       ├── AdminPage.ts
│       ├── BasePage.ts
│       ├── FaqsPage.ts
│       └── ...
├── tests
│   └── admin
│       └── faqs
│           └── faqManagement.spec.ts
├── utils
│   ├── generateAndStore.ts
│   ├── getTestData.ts
│   ├── getCredentials.ts
│   └── ...
```

🧱 Locator File Naming Convention

- Use PascalCase and suffix with `Locators.ts`
- Place them in their respective module folders inside `/locators`

Example: `FaqsPageLocators.ts`

```
export const FaqsPageLocators = {
  buttonCreateFaqs: '[data-testid="createFaq"]',
  selectUserGroupDownArrow: '[data-testid="userGroupDownArrow"]',
  textFieldEnterHeader: '[data-testid="headerInput"]',
  selectStatusDownArrow: '[data-testid="statusDownArrow"]',
  tableHeaders: (text: string) => `//th[contains(text(),'${text}')]`,
  getEllipsisButtonByHeaderText: (header: string) => `//tr[td[contains(text(),'$
{header}')]]//button[contains(@class,'ellipsis')]`,
};
```

🕐 Export All Locators in `locators/index.ts`

```
export * from './admin/MicrosoftLoginLocators';
export * from './admin/LoginPageLocators';
export * from './admin/AdminPageLocators';
export * from './admin/UserGroupLocators';
export * from './admin/CommonLocators';
export * from './admin/FaqsPageLocators';
```

📑 Static Test Data Files

Saved under `data/admin` as JSON:

Example: `data/admin/faqs.json`

```
{
  "faqManagement": {
    "pageTitle": "System Settings",
    "systemSettingLabel": "FAQ",
    "expectedFaqsBreadcrumb": ["System Settings", "FAQs"],
    "breadcrumbText": "System Settings",
    "faqTableHeaders": ["Platforms", "User", "Header", "Last Update", "Status",
```

```
"Action"]
  },
  "faqFilterIterations": [
    {
      "optionPlatformValue": "Patient App",
      "optionUserGroupValue": "Admin Group",
      "headerValue": "Login Header",
      "optionStatusValue": "Active"
    }
  ]
}
```

Use `getTestData('admin/faqs', 'faqManagement')` to access.

🔗Locator Types with Prefix Naming

Prefix	Element Type	Example
`button`	Button	`buttonCreateSchedule`
`textField`	Input / Text Field	`textFieldDoctorName`
`select`	Dropdown / Select	`selectHospitalDropdown`
`radio`	Radio Button	`radioDateToday`
`checkbox`	Checkbox	`checkboxAcceptTerms`
`toggle`	Toggle Switch	`toggleStartAMPM`
`tab`	Tab	`tabSchedule`
`modal`	Modal / Dialog	`modalDeleteConfirmation`
`table`	Table	`tableScheduleList`
`row`	Table Row	`rowDoctorSchedule`
`col`, `cell`	Table Column / Cell	`cellDoctorName`, `colStatus`
`label`	Label / Text	`labelScheduleTitle`
`icon`	Icon	`iconEllipsisMenu`
`section`	Section / Panel	`sectionFilters`
`panel`	Panel	`panelDetails`
`breadcrumb`	Breadcrumb	`breadcrumbSystemSettings`
`link`	Link / Anchor	`linkCreateNewSchedule`

Prefix	Element Type	Example
`tooltip`	Tooltip	`tooltipSlotDurationInfo`
`tag`	Badge / Tag	`tagStatusActive`
`card`	Card	`cardDoctorInfo`
`get`	Dynamic / Function	`getRowByText(text)`

🧰How to Use Locators in Page Classes

```
import { FaqsPageLocators } from '@locators';
import { assertVisible, cmdClick, cmdFill, cmdSelectMatMultiOptions } from
'@commands';

await assertVisible(this.page, FaqsPageLocators.selectUserGroupDownArrow);
await cmdSelectMatMultiOptions(this.page,
FaqsPageLocators.selectUserGroupDownArrow, 'mat-option span', ['Admin Group'],
true, 2);
await cmdFill(this.page, FaqsPageLocators.textFieldEnterHeader, 'Login Header');
await cmdClick(this.page, FaqsPageLocators.selectStatusDownArrow);
```

Use `@commands` wrappers (`cmdClick` , `cmdFill` , etc.) to enforce uniformity and reporting.

🗜️How to Write Iterative Test Cases with Data

```
import { FaqsPage } from '@pages';
import { getTestData } from '@utils';

const faqFilterIterations = getTestData('admin/faqs', 'faqFilterIterations');

faqFilterIterations.forEach(({ optionPlatformValue, optionUserGroupValue,
headerValue, optionStatusValue }) => {
  test(`Verify filters for ${optionPlatformValue}`, async () => {
    const faqsPage = new FaqsPage(page);
    await faqsPage.FillSearchFilters(optionPlatformValue, optionUserGroupValue,
headerValue, optionStatusValue);
    await faqsPage.clickClearAllButton();
    await faqsPage.verifyClearFilledSearchFilter();
  });
});
```

How to Create Page Class and Use in Test

```
// pages/admin/FaqsPage.ts
import { FaqsPageLocators } from '@locators';
import { BasePage } from './BasePage';
import { assertVisible, cmdClick, cmdFill, cmdSelectMatOption,
cmdSelectMatMultiOptions } from '@commands';

export class FaqsPage extends BasePage {
  async FillSearchFilters(platform: string, userGroup: string, header: string,
status: string) {
    await assertVisible(this.page, FaqsPageLocators.selectUserGroupDownArrow);
    await cmdSelectMatOption(this.page,
FaqsPageLocators.selectPlatformsDownArrow, 'mat-option span', undefined, true,
[platform]);
    await cmdSelectMatMultiOptions(this.page,
FaqsPageLocators.selectUserGroupDownArrow, 'mat-option span', [userGroup],
true, 2);
    await cmdFill(this.page, FaqsPageLocators.textFieldEnterHeader, header);
    await cmdClick(this.page, FaqsPageLocators.selectStatusDownArrow);
    await cmdSelectMatOption(this.page, 'mat-option span', status);
  }

  async clickClearAllButton() {
    await cmdClick(this.page, FaqsPageLocators.buttonClearAll);
  }

  async verifyClearFilledSearchFilter() {
    await assertVisible(this.page, FaqsPageLocators.selectUserGroupPlaceHolder);
  }
}
```

🧪How to Call Page Method in Test

```
// tests/admin/faqs/faqManagement.spec.ts
import { FaqsPage } from '@pages';
import { test } from '@playwright/test';

const faqsPage = new FaqsPage(page);

await faqsPage.FillSearchFilters('Doctor Portal', 'Admin Group', 'Login
Header', 'Active');
```

```
await faqsPage.clickClearAllButton();
await faqsPage.verifyClearFilledSearchFilter();
```
