Selenium project framework structure conventions

These conventions ensures your framework remains organized, scalable, and easier to maintain over time.

1. Base Classes

Folder: `src/test/java/base`

File Naming: `BaseTest.java`, `BasePage.java`

Purpose: Handles setup, teardown, and common configurations for test cases (`BaseTest`) and provides common page actions (`BasePage`).

2. Page Object Classes

Folder: `src/test/java/pages`

File Naming: Each class represents a page, so use meaningful names, such as `LoginPage.java`, `HomePage.java`, `ProductPage.java`.

Purpose: Implements the Page Object Model (POM) by encapsulating web elements and actions for each page. Each class contains elements and actions specific to that page.

3. Test Classes

Folder: `src/test/java/tests`

File Naming: Use the name of the feature or page followed by `Tests`, like `LoginTests.java`, `HomePageTests.java`, `CheckoutTests.java`.

Purpose: Contains actual test methods, usually organized by functionality or feature. Each test class uses methods from the relevant Page Object classes to perform actions and assertions.

4. Utilities (Helpers) Classes

Folder: `src/main/java/utilities` or `src/test/java/utilities`

File Naming: Descriptive names based on the utility’s purpose, such as `WaitUtil.java`, `ExcelUtil.java`, `LoggerUtil.java`.

Purpose: Stores reusable utility functions like waits, data handling, logging, and screenshot capturing. Can also include helper functions for handling specific file types (Excel, JSON).

5. Configuration Files

Folder: `config`

File Naming: Use descriptive names, like `config.properties`, `dev_config.properties`, `test_config.properties`.

Purpose: Stores configurations such as environment URLs, browser settings, credentials, and timeouts. Naming can be environmentspecific (e.g., `dev_config.properties`).

6. Driver Files

Folder: `drivers`

File Naming: Name the files according to the browser, such as `chromedriver.exe`, `geckodriver.exe`.

Purpose: Holds browser driver executables for easy access across environments.

7. Data Files

Folder: `testData`

File Naming: Use clear naming for the file contents, like `TestData.xlsx`, `users.json`, `loginData.csv`.

Purpose: Holds external test data files, such as Excel sheets, CSV files, or JSON files.

8. Reports and Logs

Folder: `reports` and `logs`

File Naming:

Reports: `TestReport.html`, `ExtentReport.html`

Logs: `Execution.log`, `Error.log`

Purpose: Stores test execution reports and logs, which help in debugging and tracking test execution.

9. Framework Configuration Files

File Names: `pom.xml` (for Maven) or `build.gradle` (for Gradle)

Purpose: Manages project dependencies and build configurations, like setting up Selenium, TestNG, Extent Reports, etc.

10. Test Suite Configurations

File Names: `testng.xml` (for TestNG), `junitplatform.properties` (for JUnit 5)

Purpose: Configures test suites, groups, parallel execution settings, and test listeners.

/SeleniumFramework Structure

```
│
├── src
│   ├── main
│   │   └── java
│   │       └── utilities
│   │           └── ExcelUtil.java
│   │           └── ConfigReader.java
│   │
│   └── test
│       └── java
│           ├── base
│           │   └── BaseTest.java
│           │
│           ├── pages
│           │   └── LoginPage.java
│           │   └── HomePage.java
│           │
│           ├── tests
│           │   └── LoginTests.java
│           │   └── HomeTests.java
│           │
│           └── utils
│               └── WaitUtil.java
│               └── LoggerUtil.java
│
├── config
│   └── config.properties
│
├── drivers
│   └── chromedriver.exe
│   └── geckodriver.exe
│
├── testData
│   └── TestData.xlsx
│
├── reports
│   └── TestReport.html
│
├── logs
│   └── Execution.log
│
├── pom.xml
└── testng.xml
```