

Designing Page Object Model

This lesson introduces the Page Object Model and explains with an example why we prefer this model.

WE'LL COVER THE FOLLOWING ^

- What is POM?
- Why do we need POM?
- POM design example

What is POM?

POM (Page Object Model) is a design pattern that is used in selenium test automation for organizing the helper code for maintainability and reducing code duplication. A page object or a page class is basically a representation of all or some of the actions and operations which can be performed on that page of your application and also acts as an interface to a page of your application under test.

Why do we need POM?

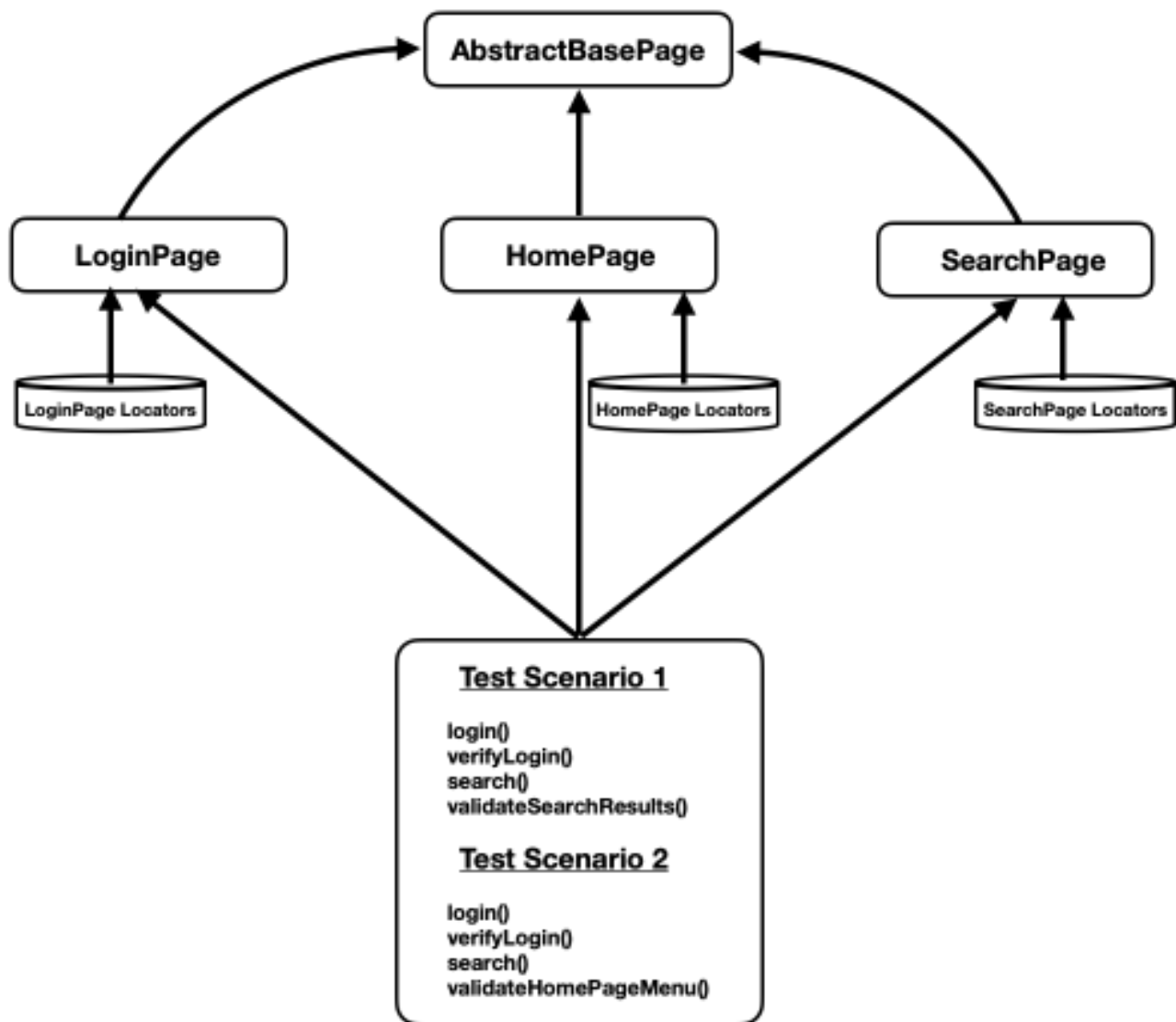
The main advantage of POM is to reduce the maintenance when the locator changes or new functionality is being added to a page. Typically, a test flow will use the page object methods to perform an operation that internally will use locators to operate on that WebElement. The Page Object Model pattern makes the test automation framework easy to use, reliable and robust. The test flow looks clean and short since we reuse the page object methods.

POM design example

We have assumed an application with three screens: **Login**, **Home**, and **Search**. We designed a POM for the same. The design shows the interaction of tests with Page Objects and locators.

In the below diagram, the locator's file will have the locators for that page in a

In the below diagram, the locator's file will have the locators for that page in a key/value format, where the key will be the element's identifier and the value is the location of the element in the page [DOM](#) (XPATH, CSS PATH etc.).



Now that you're familiar with POM, let's learn about loading the components.