Automation Testing

Page Object Model

simplilearn

A Day in the Life of an Automation Test Engineer

Mark now understands the use of CSS selectors.

Now, he has been asked to test a multiple pages in an application to test Login Page, Registration Page, and Book Flights page..

To achieve this, he will learn the page object model,app actions in cypress which will help him to find a solution for the given scenario.

Learning Objectives

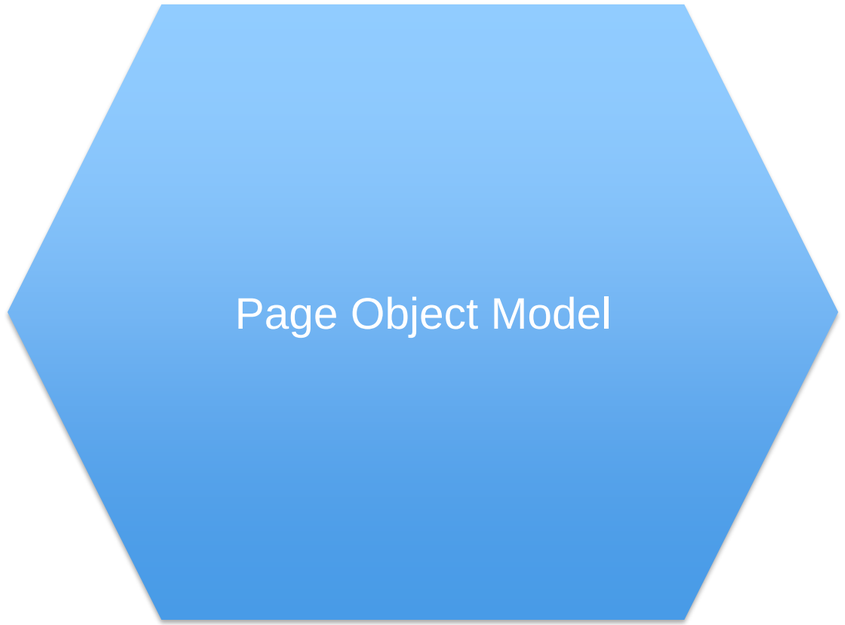
By the end of this lesson, you will be able to:

- Describe Page Object Model
- Describe Page Object issues
- Explain Page Object Pattern
- Comprehend app actions in cypress

Page Object Model: Overview

Page Object Model

Page Object Model is an automated design pattern known for its test maintenance approach and avoidance of code duplication.



Page Object Model Implementation

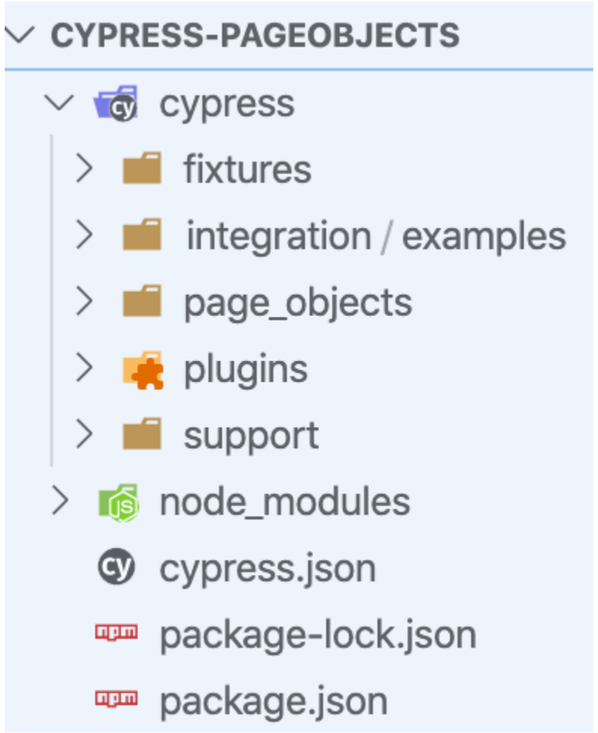
The Page Object Model may be implemented in two ways:

- Page Object Model with Page Factory
- Page Object model without Page Factory

Page Objects

Page Object

A page object is a class that represents a web application page.



Creating the Page Object Class

The class we'll be creating is simply a JavaScript class. Make a new folder in the integration folder and call it page-objects. In the page-objects folder, create a new file called todo-page.js.

Example

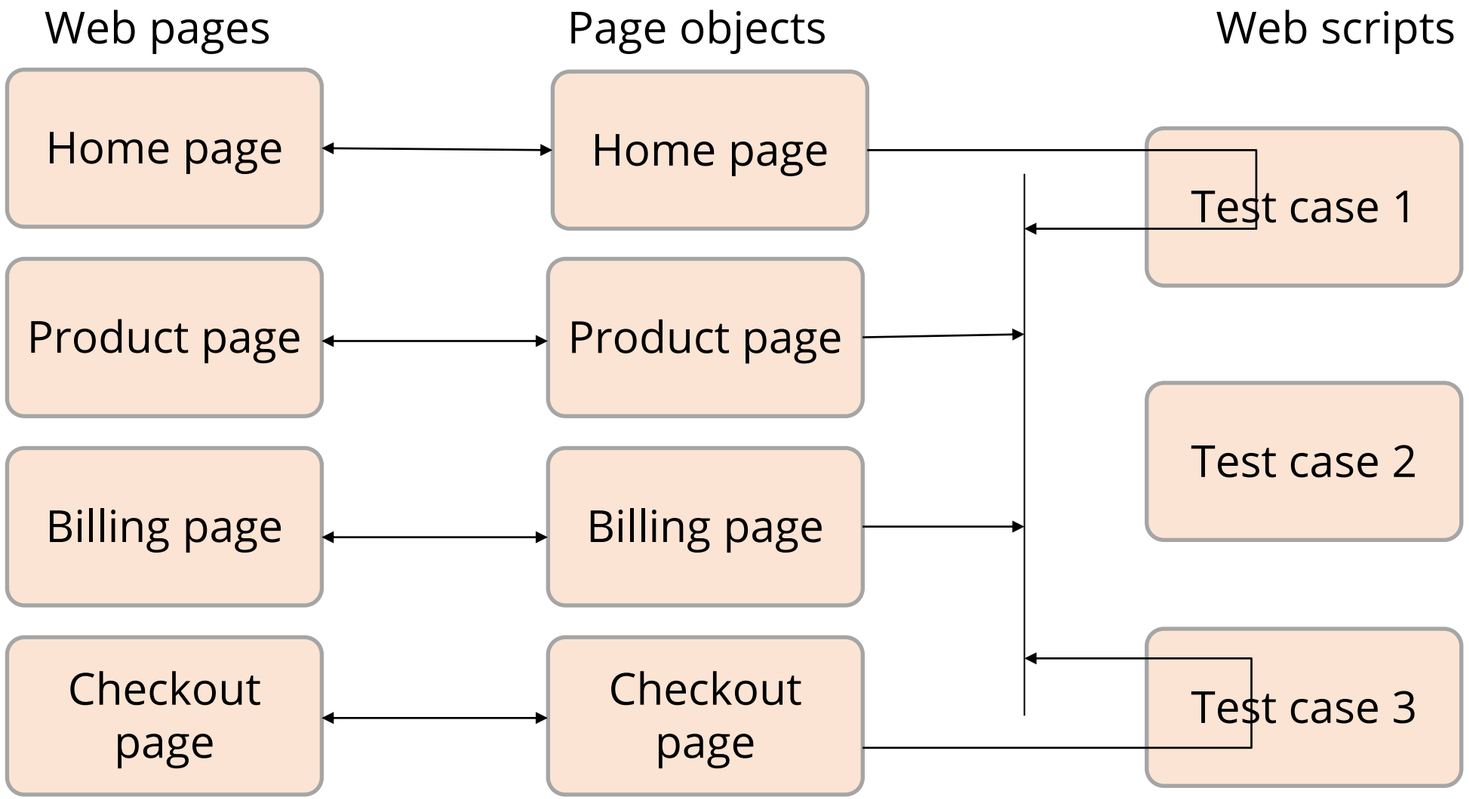
```
export class TodoPage {

    navigateToHome() {
        cy
.visit('https://example.cypress.io/todo')
        }
    }
```

Using Page Object Models

Page Object Pattern

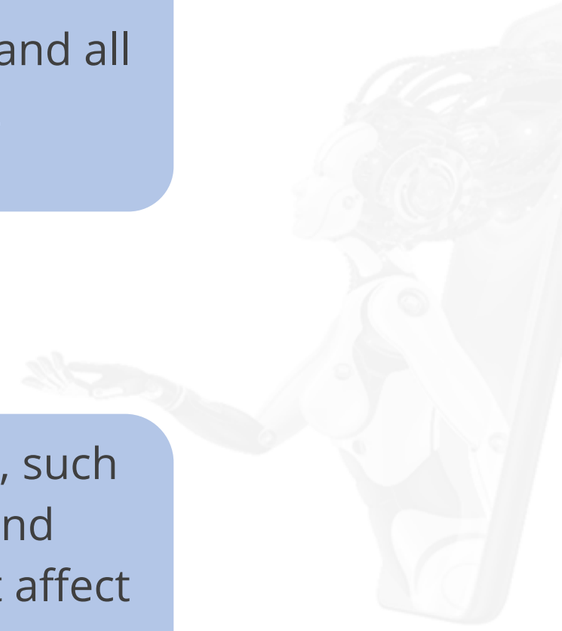
A Page Object Pattern occurs when we divide these pages into separate page files or page objects, and we create page-specific locators and methods in their own file (JavaScript Class) and utilize them directly in test scripts.



Benefits of Using Page Object Pattern

Code reusability- The same page class may be used in several tests, and all locators and methods can be reused across multiple test cases.

Code maintainability- There is a clear distinction between test code, such as functional scenarios, and page-specific code, such as locators and methods. So, if a structural change occurs on the web page, it will just affect the page object and will have no effect on the test scripts.



Page Object Model Framework Architecture

The page object model framework architecture is a tried and true design that may be easily customized using basic approaches.

Implement the Page Object Pattern in Cypress

Visit the ToolQA demo website's My Account page at http://shop.demoqa.com/my-account/.

Then register with a valid username, email address, and password. Check to see if the registration was successful.

Search for a shirt and choose two goods based on the parameters.

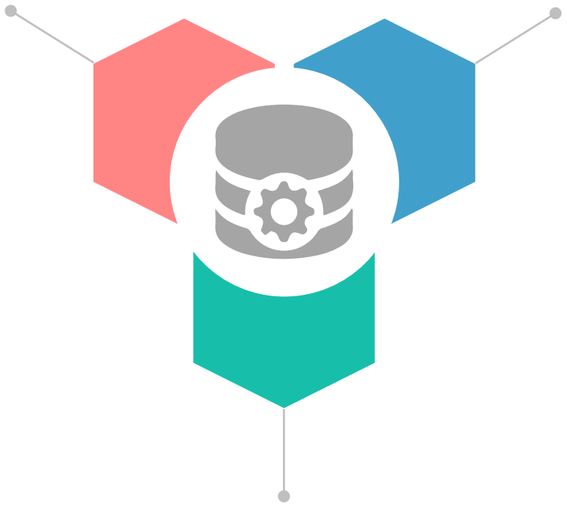
Examine the Checkout page to ensure that the correct product has been put to the cart.

Enter the Billing Data as well as the Login Information.

Place the order button and check to see if the order was successfully placed.

Benefits of Page Object Model in Cypress

Reusable



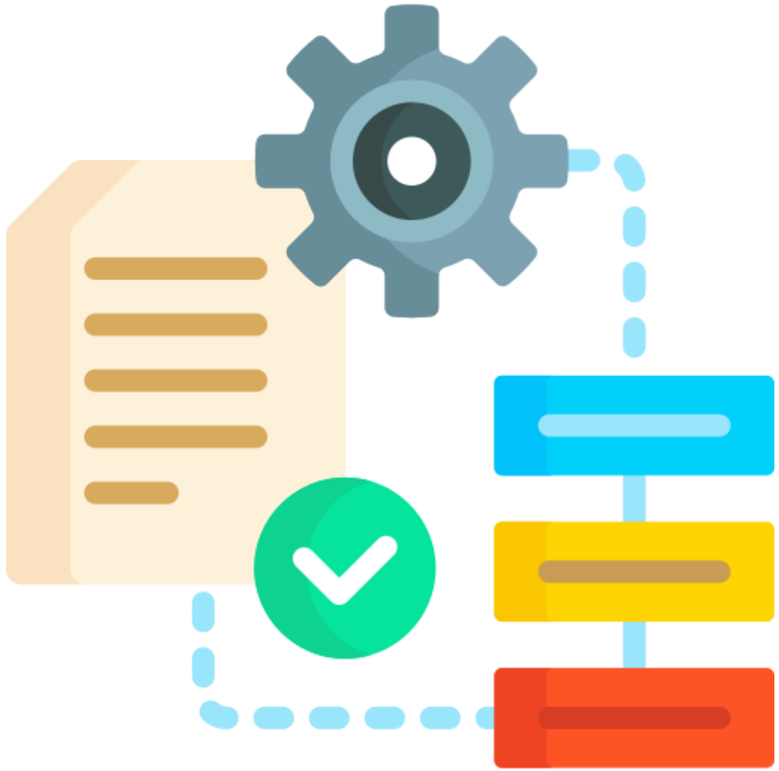
Clean and easily understandable

Accessible

Page Object Issues

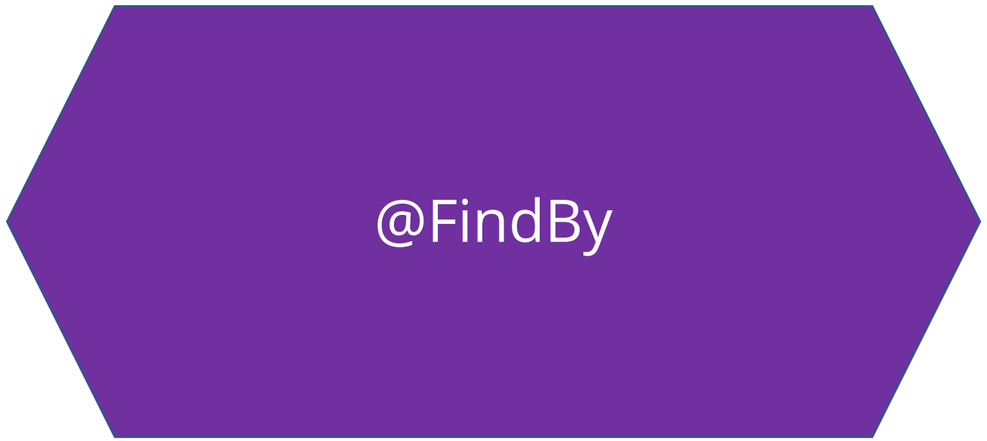
Page Factory

Page Factory is a method of putting the "Page Object Model" into action. The idea of separation of Page Object Repository and Test Methods is followed here. It is a Page Object Model intrinsic notion that is highly efficient.



@FindBy:

In Page Factory, the @FindBy annotation is used to identify and declare web items using various locators.



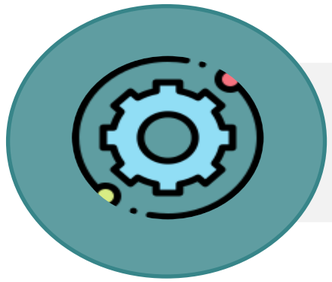
initElements()

The initElements static function of the Page Factory class is used to initialize all web elements found using the @FindBy annotation. As a result, instantiating the Page classes becomes simple.

Page Layer

There are two pages in page layer which can be discussed as follows:

HomePage- The page that appears when a URL is typed and where we enter our login information.



SearchPage- A page that appears following a successful login.

Page Object Model (POM) vs Page Factory

Page Object Model (POM)	Page Factory
It is a design pattern approach.	It is a class provided by Selenium Webdriver.
It is not the best way to complete the tasks.	It is the best way to complete the tasks.
It does not handle exceptions well.	It handles exceptions effectively.
For performing tasks, there is cache storage.	Cache storage is not required.

Creating Page Object Model

Problem Statement:

You are required to create page object model.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to create page object model are:

1. Create page object model

App Actions in Cypress

Application Actions Limitations

Calling too many actions too fast

Actions are restricted

Limitations

PageObjects vs Appactions

PageObjects

- Encapsulated and reused inside each PageObject
- Embedded into each PageObject
- Intuitive usage
- Each page has a single file with the same name as the page
- Some UI changes will fail tests until the PageObjects are updated

Appactions

- Supports redundant selectors on each AppAction
- Embedded into each AppAction
- Usage is not always intuitive
- Pages are utilized on the fly during actions; ONE may need to "Find/Replace" modifications to a page

PageObjects vs Appactions

PageObjects

- The IDE will autocomplete page activities.
- If the pages and their actions are modular enough, tests are quite simple to develop and comprehend.

Appactions

- Users should browse over the current custom commands.js and select which one works best for business.
- Because actions are blackboxes of functionality, there may be a propensity to reinvent the wheel.

Key Takeaways

- Page objects are designed to make end-to-end tests more readable and maintainable.
- The Page Object Model design pattern allows users to create tests that are faster, easier, and cleaner.
- Cypress also includes functionality for implementing the Page Object Pattern and incorporating it into test scripts.
- Custom commands, in addition to test scripts, can incorporate and implement Page Objects.