

Selenium Page Object Model Guide PDF Page

Selenium Page Object Model Guide PDF Page: Content + <head> (Uncodemy)

FINAL URL: <https://uncodemy.com/selenium-page-object-model-guide-pdf>

Publish this page at exactly this URL (non-www, no trailing slash). Canonical, og:url and sitemap entry must match it.

Prepared: 26 October 2026 · **Primary keyword:** selenium page object model guide pdf · **Search volume:** not checked (keyword tool out of credits); verify in Search Console.

Read first (5 flags):

- Your course PDF title, page count and size were not shared.** Download cards are fill-in slots ([CONFIRM]). Card 1 is your course PDF; cards 2 and 3 only if you have more files.
- One unique file per button.** The same course PDF repeated on many pages is fine as a resource, but each page's text must stay unique and useful without the file.
- Free or gated?** Not stated. The page says "Download", not "Free". If a form is used, add a consent checkbox and Privacy Policy link (DPPD Act 2023).
- No made-up stats or testimonials;** no placement or salary claims without a named, dated source.
- Author and reviewer are placeholders;** use real named trainers and one real "Last reviewed" date.

PART 1: Page plan

Item	Decision
Final URL	https://uncodemy.com/selenium-page-object-model-guide-pdf
Intent (anti-cannibalisation)	SELENIUM POM IN JAVA. The Playwright version is on a separate page.
Why not thin	Explainer-style sections, FAQs and cross-links make the page useful even without a download
Schema	WebSite, Organization, Breadcrumb, CollectionPage, FAQPage (only FAQs visible on the page). PDF <code>ItemList</code> provided commented, to enable after titles are final
Share image	https://uncodemy.com/image/logo-og.jpg for now; custom 1200x630 card recommended

PART 2: Page content (copy-paste ready)

SEO Title (47 chars):

Selenium Page Object Model Guide PDF | Uncodemy

Meta Description (128 chars):

Download the Selenium Page Object Model guide PDF from Uncodemy: structure, base page, locators and a worked Java login example.

H1: Selenium Page Object Model Guide PDF Downloads **Breadcrumb:** Home > Selenium Page Object Model Guide PDF **Final URL:** <https://uncodemy.com/selenium-page-object-model-guide-pdf>

Byline: Prepared by the Uncodemy trainer team [CONFIRM: author name and role] · Reviewed by [CONFIRM: reviewer name and role] · Last reviewed: [DATE]

Quick Answer

This page offers Uncodemy's Selenium Page Object Model guide as a PDF download. The Page Object Model is a design pattern in which each page or component of the application is represented by a class that holds its locators and actions. The guide shows the structure, a worked Java example and the benefits and pitfalls.

Download Selenium Page Object Model Guide PDF

[CONFIRM: free or gated, and whether details are needed before download]

PDF 1 (main): [CONFIRM TITLE: your Software Testing course PDF]
[CONFIRM: one line on what it covers and who it is for] · [N] pages · [X] MB · PDF · Updated [DATE]
[Download PDF] → <https://uncodemy.com/pdf/selenium-page-object-model-guide.pdf> (your course PDF, uploaded under this file name)

PDF 2 (optional): [CONFIRM TITLE]
Only if you have a second PDF that genuinely fits this topic. **[Download PDF]** → <https://uncodemy.com/pdf/selenium-page-object-model-guide-2.pdf>

PDF 3 (optional): [CONFIRM TITLE]

Same rule. **[Download PDF]** → <https://uncodemy.com/pdf/selenium-page-object-model-guide-3.pdf>

Each button must open its own file. Do not point several buttons at one file. Delete unused optional cards.

What is the Page Object Model?

The Page Object Model, or POM, is a design pattern for UI test automation. Each page of the application, or each significant component such as a header or a modal, gets a class. The class stores the locators of the elements on that page and exposes methods that describe what a user can do there, such as login or searchFor. Tests then call these methods and contain assertions and flow, but no locators. The result is that when the interface changes, you update one page class instead of dozens of tests, and tests read like descriptions of user behaviour. Without POM, locators are copied through many tests and maintenance becomes the main cost of automation.

Without POM versus with POM

Aspect	Without POM	With POM
Locators	Repeated in many tests	Defined once in the page class
Maintenance	Many files change when the UI changes	One page class changes
Readability	Low-level driver calls in tests	Tests read as user actions
Reuse	Copy and paste	Methods reused across tests
Separation	Mixed locators, actions and checks	Clear layers

Typical project layers

Layer	Responsibility
Base page or base test	Shared driver handling, waits and helpers
Page classes	Locators and user actions of one page
Test classes	Scenarios and assertions
Utilities	Configuration, data readers, screenshots
Test data and resources	Files and environment settings

A base page with an explicit wait helper

A small shared parent class for page objects. Illustrative Java using Selenium 4.

```
import java.time.Duration;
import org.openqa.selenium.*;
import org.openqa.selenium.support.ui.*;

public abstract class BasePage {
    protected final WebDriver driver;
    private final WebDriverWait wait;

    protected BasePage(WebDriver driver) {
        this.driver = driver;
        this.wait = new WebDriverWait(driver, Duration.ofSeconds(10));
    }

    protected WebElement visible(By locator) {
        return wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
    }

    protected void type(By locator, String text) {
        WebElement el = visible(locator);
        el.clear();
        el.sendKeys(text);
    }

    protected void click(By locator) {
        wait.until(ExpectedConditions.elementToBeClickable(locator)).click();
    }
}
```

A login page object and a test

The page object hides locators and returns the next page after a successful action.

```
public class LoginPage extends BasePage {
    private final By email = By.id("email");
    private final By password = By.id("password");
    private final By loginButton = By.id("login");
    private final By error = By.cssSelector(".error-message");

    public LoginPage(WebDriver driver) { super(driver); }

    public LoginPage open(String baseUrl) {
        driver.get(baseUrl + "/login");
        return this;
    }

    public DashboardPage loginAs(String user, String pass) {
        type(email, user);
        type(password, pass);
        click(loginButton);
        return new DashboardPage(driver);
    }
}
```

```

    public String errorText() {
        return visible(error).getText();
    }
}

// In the test class:
@Test
public void validLogin() {
    DashboardPage dashboard = new LoginPage(driver)
        .open(baseUrl)
        .loginAs("test.user@example.com", "your-test-password");
    Assert.assertTrue(dashboard.isDisplayed());
}

```

PageFactory and its status

Selenium offers PageFactory with the `@FindBy` annotation to initialise elements declared as fields. It reduces boilerplate and can cache or lazily locate elements. Many teams prefer plain By locators with explicit waits, as in the example above, because they are simple, work well with dynamic pages and are easy to debug. Check the current Selenium documentation for the status of PageFactory in your version before standardising on it, and apply whichever approach your team picks consistently.

Design rules for good page objects

- Page classes contain locators and actions, not assertions about business rules.
- Methods return the next page object when navigation happens, or the same page when it does not.
- Name methods after user intent, such as `submitOrder`, not `clickButton3`.
- Prefer stable locators such as ids, data-test attributes or accessible names over long XPath chains.
- Use explicit waits in the base class instead of fixed sleeps.
- Model reusable components, such as a navigation bar, as their own classes.

Locator strategy comparison

Locator	Stability	Note
Dedicated test attribute (data-test)	High	Best, agree with developers to add them
ID	High if not auto-generated	Fast and clear
Name or accessible label	Medium to high	Tied to meaning, not layout
CSS selector	Medium	Keep short and specific
XPath with positions	Low	Breaks easily with layout changes

Common mistakes

- **Assertions buried in page classes** that hide what a test verifies.
- **Huge page classes** covering several pages; split into components.
- **Fragile XPath locators**.
- **Static driver fields** that break parallel runs.
- **Fixed sleeps** in page methods.

Common component objects

Component	Typical methods
Header or navigation bar	<code>openMenu</code> , <code>search</code> , <code>goToCart</code>
Modal dialog	<code>confirm</code> , <code>cancel</code> , <code>readMessage</code>
Data table	<code>getRowCount</code> , <code>findRowByText</code> , <code>sortBy</code>
Date picker	<code>selectDate</code> , <code>clearDate</code>

Refactoring existing scripts to POM

1. Pick one flow, such as login, and copy its locators into a new class.
2. Move the actions into methods named by intent.
3. Replace the old script lines with the new calls.
4. Run the test and compare results.
5. Repeat flow by flow rather than rewriting everything at once.

When POM may be more than you need

- A throwaway script for one-time use.
- A very small project with one or two pages.
- In these cases keep the code simple, and introduce page objects when duplication starts to hurt.

Related Uncodemy resources

- [Playwright Page Object Model Guide PDF](#)
- [Automation Testing TestNG Guide PDF](#)
- [Automation Testing Cucumber Guide PDF](#)
- [Automation Testing CI/CD Guide PDF](#)

Learn with a trainer [use only the line that is true]

- **Software Testing course:** [CONFIRM URL]. [CONFIRM URL] Link only if this course page is live; otherwise use <https://uncodemy.com/all-courses>

(One soft call to action: course link or free demo. No pop-up.)

About these PDFs

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to [CONFIRM: feedback email] and the trainer team will review it.

Frequently Asked Questions

What is the Page Object Model in Selenium?

It is a design pattern where each page or component is a class holding its locators and user actions, so tests stay clean and maintenance is centralised.

What are the advantages of POM?

Less duplication, easier maintenance, more readable tests, better reuse and clear separation between test logic and page details.

Should assertions go inside page objects?

Generally no. Keep assertions in tests so it is clear what each test verifies, and let page objects expose state, such as text or visibility.

What is PageFactory?

PageFactory is a Selenium support class that initialises elements declared with the FindBy annotation. Many teams use plain By locators with explicit waits instead.

Can POM be used with Cucumber or TestNG?

Yes. Step definitions in Cucumber and test methods in TestNG both call page object methods.

Are these PDFs free to download?

[CONFIRM: state clearly whether the PDFs are free, or whether a short form is needed first.]

The last FAQ contains [CONFIRM], so it is NOT in the FAQ schema. Fill it and keep it on the page, or delete it.

PART 3: <head> section (copy-paste ready)

Final URL: <https://uncodemy.com/selenium-page-object-model-guide-pdf>

```
<!-- =====>
UNCODEMY <head> | FINAL URL: https://uncodemy.com/selenium-page-object-model-guide-pdf
Prepared 26 October 2026. Paste into the page <head>; keep your site-wide favicon, fonts, analytics.
BEFORE GO-LIVE: (1) set dateModified in the JSON-LD to the real publish date;
(2) replace og:image with a 1200x630 share card when available;
(3) once PDF titles are final, uncomment the ItemList block at the bottom.
===== -->

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Selenium Page Object Model Guide PDF | Uncodemy</title>
<meta name="description" content="Download the Selenium Page Object Model guide PDF from Uncodemy: structure, base page, locators and a worked Java login
example.">
<link rel="canonical" href="https://uncodemy.com/selenium-page-object-model-guide-pdf">
<meta name="robots" content="index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:-1">
<meta name="author" content="Uncodemy">
<meta name="publisher" content="Uncodemy Edutech Pvt. Ltd.">
<meta name="language" content="en-IN">
<link rel="alternate" hreflang="en-in" href="https://uncodemy.com/selenium-page-object-model-guide-pdf">
<link rel="alternate" hreflang="x-default" href="https://uncodemy.com/selenium-page-object-model-guide-pdf">

<!-- Open Graph -->
<meta property="og:type" content="website">
<meta property="og:site_name" content="Uncodemy">
<meta property="og:locale" content="en-IN">
<meta property="og:url" content="https://uncodemy.com/selenium-page-object-model-guide-pdf">
<meta property="og:title" content="Selenium Page Object Model Guide PDF | Uncodemy">
<meta property="og:description" content="Download the Selenium Page Object Model guide PDF from Uncodemy: structure, base page, locators and a worked Java log
example.">
<meta property="og:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta property="og:image:alt" content="Uncodemy Selenium Page Object Model Guide PDF study material">

<!-- X (Twitter) -->
<meta name="twitter:card" content="summary_large_image">
<meta name="twitter:site" content="@uncodemy">
<meta name="twitter:title" content="Selenium Page Object Model Guide PDF | Uncodemy">
<meta name="twitter:description" content="Download the Selenium Page Object Model guide PDF from Uncodemy: structure, base page, locators and a worked Java lo
example.">
<meta name="twitter:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta name="twitter:image:alt" content="Uncodemy Selenium Page Object Model Guide PDF study material">

<!-- Structured data: one @graph (WebSite, Organization, Breadcrumb, CollectionPage, FAQPage) -->
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@graph": [
    {
      "@type": "WebSite",
      "@id": "https://uncodemy.com/#website",
      "url": "https://uncodemy.com/",
      "name": "Uncodemy",
      "publisher": {
        "@id": "https://uncodemy.com/#organization"
      },
      "inLanguage": "en-IN"
    }
  ]
}
```

```

},
{
  "@type": "EducationalOrganization",
  "@id": "https://uncodemy.com/#organization",
  "name": "Uncodemy",
  "legalName": "Uncodemy Edutech Pvt. Ltd.",
  "url": "https://uncodemy.com/",
  "logo": {
    "@type": "ImageObject",
    "url": "https://uncodemy.com/uc-layout/img/Logo.webp"
  },
  "email": "info@uncodemy.com",
  "identifier": {
    "@type": "PropertyValue",
    "name": "CIN",
    "value": "U85499UP2024PTC195902"
  },
  "contactPoint": [
    {
      "@type": "ContactPoint",
      "telephone": "+91-9818366550",
      "contactType": "admissions",
      "areaServed": "IN",
      "availableLanguage": [
        "en",
        "hi"
      ]
    }
  ],
  "sameAs": [
    "https://x.com/uncodemy",
    "https://www.facebook.com/uncodemyofficial/",
    "https://www.linkedin.com/company/uncodemy/",
    "https://www.instagram.com/uncodemyofficial/"
  ]
},
{
  "@type": "BreadcrumbList",
  "@id": "https://uncodemy.com/selenium-page-object-model-guide-pdf#breadcrumb",
  "itemListElement": [
    {
      "@type": "ListItem",
      "position": 1,
      "name": "Home",
      "item": "https://uncodemy.com/"
    },
    {
      "@type": "ListItem",
      "position": 2,
      "name": "Selenium Page Object Model Guide PDF",
      "item": "https://uncodemy.com/selenium-page-object-model-guide-pdf"
    }
  ]
},
{
  "@type": "CollectionPage",
  "@id": "https://uncodemy.com/selenium-page-object-model-guide-pdf#webpage",
  "url": "https://uncodemy.com/selenium-page-object-model-guide-pdf",
  "name": "Selenium Page Object Model Guide PDF Downloads",
  "description": "Download the Selenium Page Object Model guide PDF from Uncodemy: structure, base page, locators and a worked Java login example.",
  "inLanguage": "en-IN",
  "isPartOf": {
    "@id": "https://uncodemy.com/#website"
  },
  "publisher": {
    "@id": "https://uncodemy.com/#organization"
  },
  "breadcrumb": {
    "@id": "https://uncodemy.com/selenium-page-object-model-guide-pdf#breadcrumb"
  },
  "about": {
    "@type": "Thing",
    "name": "Selenium Page Object Model Guide"
  },
  "primaryImageOfPage": {
    "@type": "ImageObject",
    "url": "https://uncodemy.com/image/logo-og.jpg"
  },
  "dateModified": "2026-10-26"
},
{
  "@type": "FAQPage",
  "@id": "https://uncodemy.com/selenium-page-object-model-guide-pdf#faq",
  "inLanguage": "en-IN",
  "isPartOf": {
    "@id": "https://uncodemy.com/selenium-page-object-model-guide-pdf#webpage"
  },
  "mainEntity": [
    {
      "@type": "Question",
      "name": "What is the Page Object Model in Selenium?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "It is a design pattern where each page or component is a class holding its locators and user actions, so tests stay clean and maintenance
centralised."
      }
    },
    {
      "@type": "Question",
      "name": "What are the advantages of POM?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Less duplication, easier maintenance, more readable tests, better reuse and clear separation between test logic and page details."
      }
    },
    {
      "@type": "Question",
      "name": "Should assertions go inside page objects?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Generally no. Keep assertions in tests so it is clear what each test verifies, and let page objects expose state, such as text or
visibility."
      }
    }
  ]
}

```

```

    }
  },
  {
    "@type": "Question",
    "name": "What is PageFactory?",
    "acceptedAnswer": {
      "@type": "Answer",
      "text": "PageFactory is a Selenium support class that initialises elements declared with the FindBy annotation. Many teams use plain By locators w
explicit waits instead."
    }
  },
  {
    "@type": "Question",
    "name": "Can POM be used with Cucumber or TestNG?",
    "acceptedAnswer": {
      "@type": "Answer",
      "text": "Yes. Step definitions in Cucumber and test methods in TestNG both call page object methods."
    }
  }
]
}
}
</script>

<!-- ADD AFTER THE PDF TITLES ARE FINAL (remove this wrapper, fill every [...] value):
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ItemList",
  "@id": "https://uncodemy.com/selenium-page-object-model-guide-pdf#pdfs",
  "name": "Selenium Page Object Model Guide PDF downloads",
  "numberOfItems": 3,
  "itemListElement": [
    {
      "@type": "ListItem",
      "position": 1,
      "item": {
        "@type": "DigitalDocument",
        "name": "[PDF 1 TITLE]",
        "description": "[PDF 1 ONE-LINE DESCRIPTION]",
        "encodingFormat": "application/pdf",
        "url": "https://uncodemy.com/pdf/selenium-page-object-model-guide.pdf",
        "inLanguage": "en",
        "publisher": {
          "@id": "https://uncodemy.com/#organization"
        }
      }
    },
    {
      "@type": "ListItem",
      "position": 2,
      "item": {
        "@type": "DigitalDocument",
        "name": "[PDF 2 TITLE]",
        "description": "[PDF 2 ONE-LINE DESCRIPTION]",
        "encodingFormat": "application/pdf",
        "url": "https://uncodemy.com/pdf/selenium-page-object-model-guide-2.pdf",
        "inLanguage": "en",
        "publisher": {
          "@id": "https://uncodemy.com/#organization"
        }
      }
    },
    {
      "@type": "ListItem",
      "position": 3,
      "item": {
        "@type": "DigitalDocument",
        "name": "[PDF 3 TITLE]",
        "description": "[PDF 3 ONE-LINE DESCRIPTION]",
        "encodingFormat": "application/pdf",
        "url": "https://uncodemy.com/pdf/selenium-page-object-model-guide-3.pdf",
        "inLanguage": "en",
        "publisher": {
          "@id": "https://uncodemy.com/#organization"
        }
      }
    }
  ]
}
}
</script>
-->

```

DEVELOPER CHECKLIST

- ☐ Final URL live at `https://uncodemy.com/selenium-page-object-model-guide-pdf`; canonical, og:url and sitemap identical.
- ☐ Each button: `<a href="/pdf/selenium-page-object-model-guide.pdf" download type="application/pdf">`, its own file, size shown.
- ☐ If a form precedes download: consent checkbox, Privacy Policy link, nothing pre-ticked.
- ☐ One `<h1>`; real HTML tables; code samples in `<pre><code>`.
- ☐ Remove any `<meta name="keywords">` from the template.
- ☐ Test in Google Rich Results Test and `validator.schema.org` after publishing.
- ☐ Add to sitemap and any PDF/resources hub.

STILL NEEDED FROM YOU

#	Item
1	Course PDF title, description, page count, size, and the file URL for each page
2	Free or gated (and form fields if gated)

3	Author, reviewer, publish and last-reviewed dates
4	Course URL for the CTA (lines marked [CONFIRM])
5	Feedback email
6	Confirmation that Uncodemy owns the rights to every PDF