



# Page object models

## Introduction

Large test suites can be structured to optimize ease of authoring and maintenance. Page object models are one such approach to structure your test suite.

A page object represents a part of your web application. An e-commerce web application might have a home page, a listings page and a checkout page. Each of them can be represented by page object models.

Page objects **simplify authoring** by creating a higher-level API which suits your application and **simplify maintenance** by capturing element selectors in one place and create reusable code to avoid repetition.

## Implementation

Page object models wrap over a Playwright [Page](#).

**Sync**   **Async**

---

models/search.py

```
class SearchPage:
    def __init__(self, page):
        self.page = page
        self.search_term_input = page.locator('[aria-label="Enter your search term"]')

    def navigate(self):
        self.page.goto("https://bing.com")

    def search(self, text):
        self.search_term_input.fill(text)
        self.search_term_input.press("Enter")
```

Page objects can then be used inside a test.

**Sync**

**Async**

test\_search.py

```
from models.search import SearchPage
```

```
# in the test
```

```
page = browser.new_page()
```

```
search_page = SearchPage(page)
```

```
search_page.navigate()
```

```
search_page.search("search query")
```