
accenture
Latvia ATC
Test Automation Engineering Fundamentals: Java + Selenium WebDriver
Page Object Model
GROW CONFI DENTLY

Why do we need Page Object Model?

accenture
Latvia ATC

During work so far sometime same webElement is used multiple times during the test, which can lead to things getting messy and hard to maintain.
Of course if there is only 1 test class it is possible to have all the elements stored in class variables, but as websites gets more complex, more than just 1 test class is needed. So the next logical step would be saving all the elements separately in class.
Now lets not forget that if we save them all in the same class it again can get quite messy.

Solution to that would be to use Page Object Model.
As name indicates we save each page as a separate class and use Selenium build in PageFactory.

Defining a Page

accenture
Latvia ATC

First of all page needs to be defined.
“how” annotation can be used for defining elements. Code example:

```
public class articlePage {
    @FindBy(how = How.TAG_NAME, using = "h1")
    private WebElement pageHeader;
    @FindBy(how = How.TAG_NAME, using = "h2")
    private List<WebElement> pageSecondaryHeaders;

    public WebElement getPageHeader() {
        return pageHeader();
    }

    public String getPageSecondaryHeaderText(int index) {
        return pageSecondaryHeaders.get(index).getText();
    }
}
```

How annotation

accenture
Latvia ATC

“how” **full list of how annotations:**

- CLASS_NAME
- CSS
- ID
- ID_OR_NAME
- LINK_TEXT
- NAME
- PARTIAL_LINK_TEXT
- TAG_NAME
- UNSET
- XPATH

Calling Page in tests

accenture
Latvia ATC

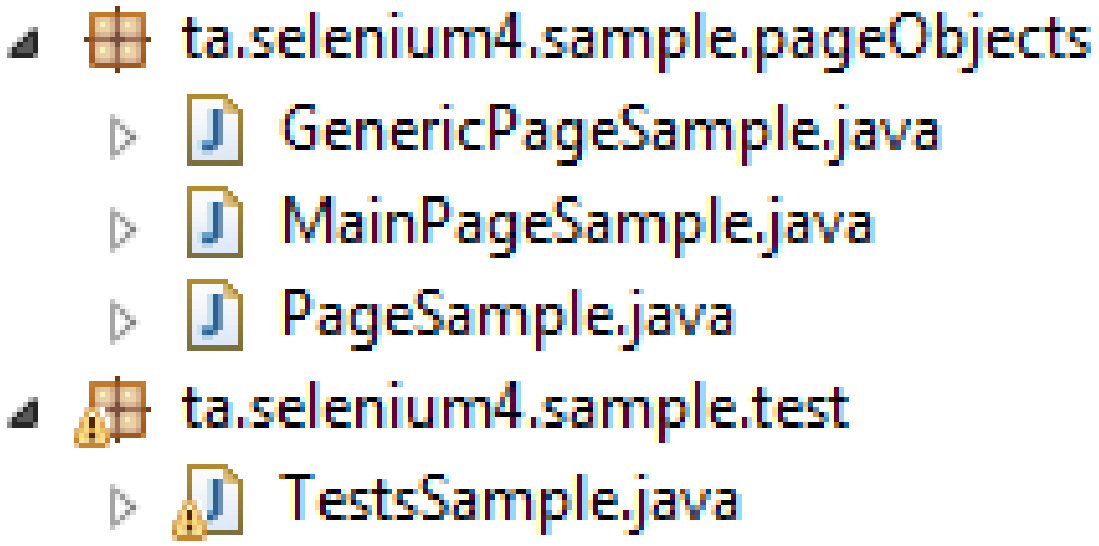
```
public class ArticlePage {
    @FindBy(how = How.TAG_NAME, using = "h1")
    private WebElement pageHeader;

    public String getPageHeaderText() {
        return pageHeader.getText();
    }
}
```

```
import org.openqa.selenium.support.PageFactory;
public class Test1 {
    static ArticlePage article = PageFactory.initElements(driver, ArticlePage.class);

    public void testMainPage() throws Exception {
        assertEquals("In news", article.getPageHeaderText());
    }
}
```

Project structure example:

accenture
Latvia ATC

Usually there is a separate folder for the pageObject and then there is a folder for all the tests.
Additionally classes with Driver set up can be implemented.



Additional methods in Page

accenture
Latvia ATC

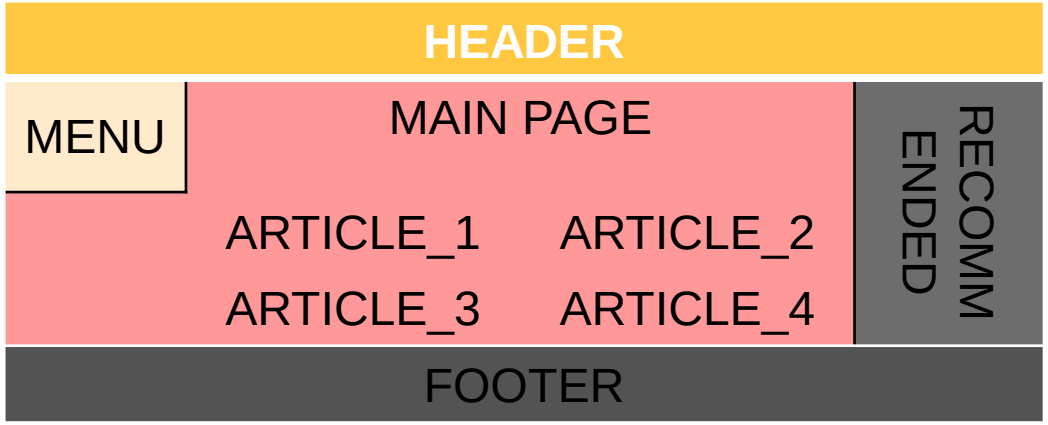
Not only locators can be defined in Page, but also actions, which can be performed on this Page.

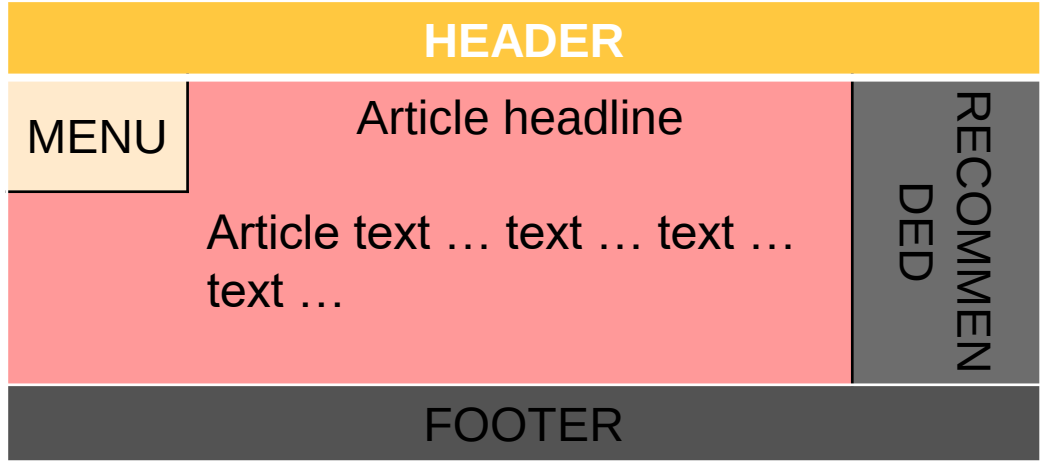
```
public class ArticlePage {
    @FindBy(how = How.NAME, using = “search")
    private WebElement searchField;
    @FindBy(how = How.NAME, using = “go")
    private WebElement searchButton;

    public void searchForText(String text) {
        searchField.clear();
        searchField.sendKeys(text);
        searchButton.click();
    }
}
```

Inheritance

accenture
Latvia ATC

A lot of page have common elements, like Header, Menu, Footer, which would repeat for every page.





Inheritance

accenture
Latvia ATC

In order not to define same thing in all of the Page Classes – one common or general class can be created and then inherited, e.g.:

```
public class GeneralPage {
    @FindBy(how = How.CSS, using = “.header")
    private WebElement header;
    public WebElement getHeader() {...}
}
```

```
public class ArticlePage extends GeneralPage { ... }
```

```
public class Test1 {
    static ArticlePage article = PageFactory.initElements(driver, ArticlePage.class);

    public void testMainPage() throws Exception {
        assertEquals("In news", article.getHeader().getText());
    }
}
```

A few additional things

accenture
Latvia ATC

The model is called Page Object, however that doesn't mean ONLY pages can be used as classes. If there is a logical block of elements which can be separated in a class, than that class can be created.

Other example if you need to test only part of functionally that part can be separated in a classes, without mapping every single element on page.

QUESTIONS

accenture
Latvia ATC