Creating Easy to Maintain Selenium Test Code

Jason Roberts
.NET DEVELOPER

@robertsjason www.dontcodetired.com

Overview

An overview of Page Object Models

Which elements should you add to a POM?

Create a new POM class

Navigate to a POM page

Refactor tests to use POMs

Navigate to other POMs

Encapsulate explicit waits in POMs

Form filling and submission with POMs

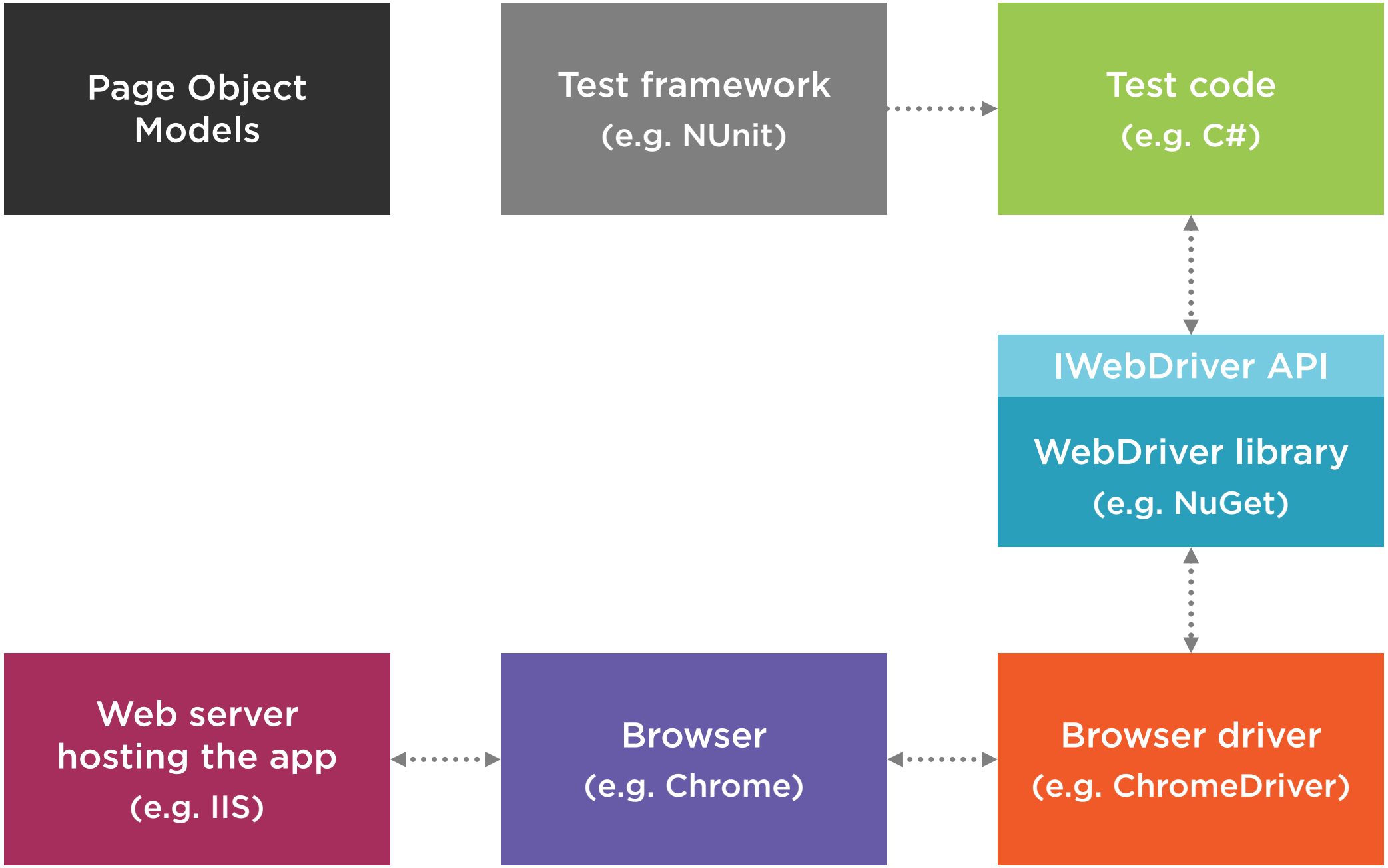
Test form validation with POMs

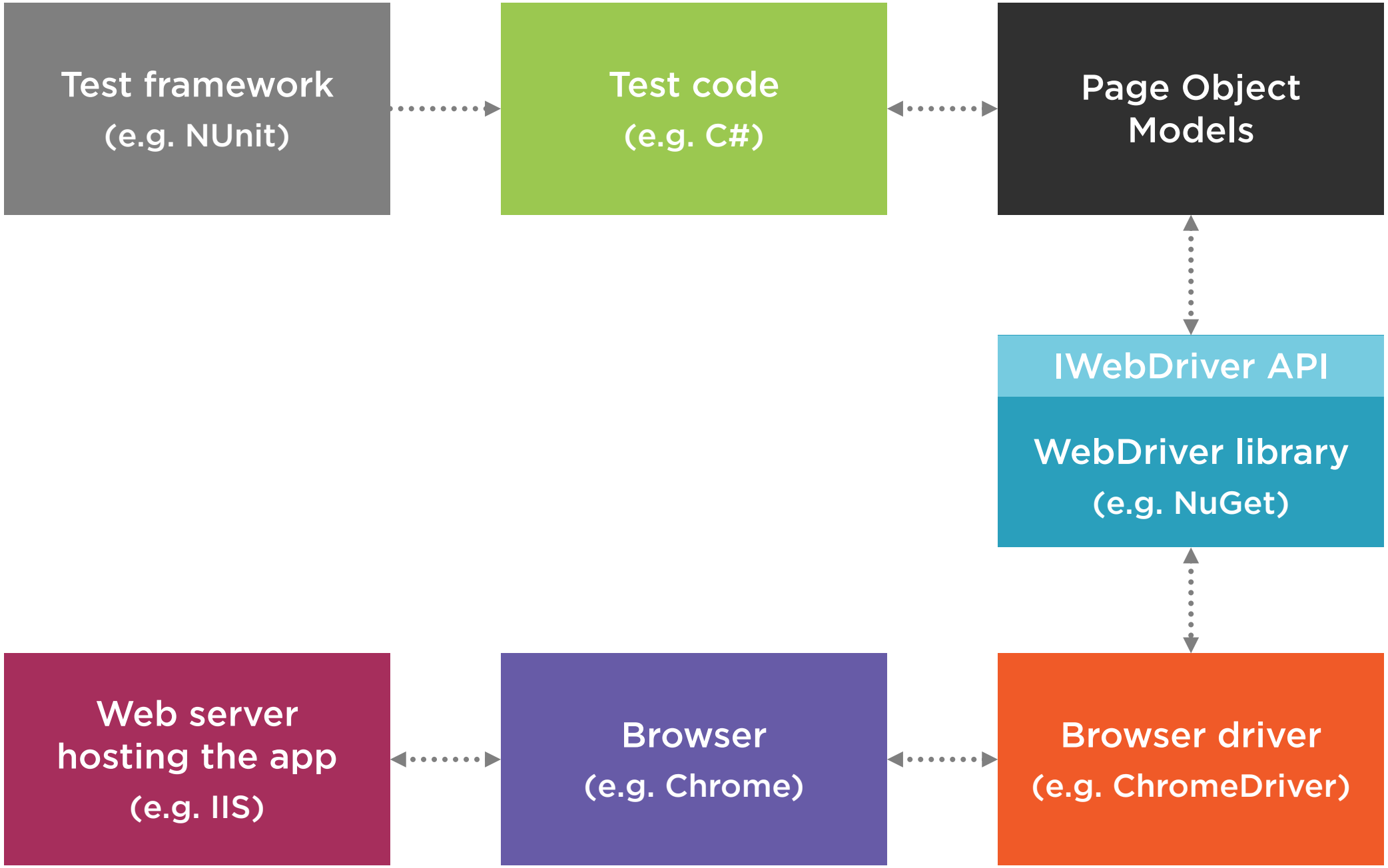
Reusing web driver instances

Page Object Model considerations

Page Object Model

Provides test code with a logical view of the user interface while abstracting away the low level details of the user interface implementation.


Page Object Models
Test framework
(e.g. NUnit)
Test code
(e.g. C#)
IWebDriver API
WebDriver library
(e.g. NuGet)
Web server hosting the app
(e.g. IIS)
Browser
(e.g. Chrome)
Browser driver
(e.g. ChromeDriver)


Test framework
(e.g. NUnit)
Test code
(e.g. C#)
Page Object Models
IWebDriver API
WebDriver library
(e.g. NuGet)
Web server hosting the app
(e.g. IIS)
Browser
(e.g. Chrome)
Browser driver
(e.g. ChromeDriver)

Benefits of Page Object Models

Reduce cost of testing

- Reduce test code duplication
- Insulate tests from UI changes

Improve readability of tests

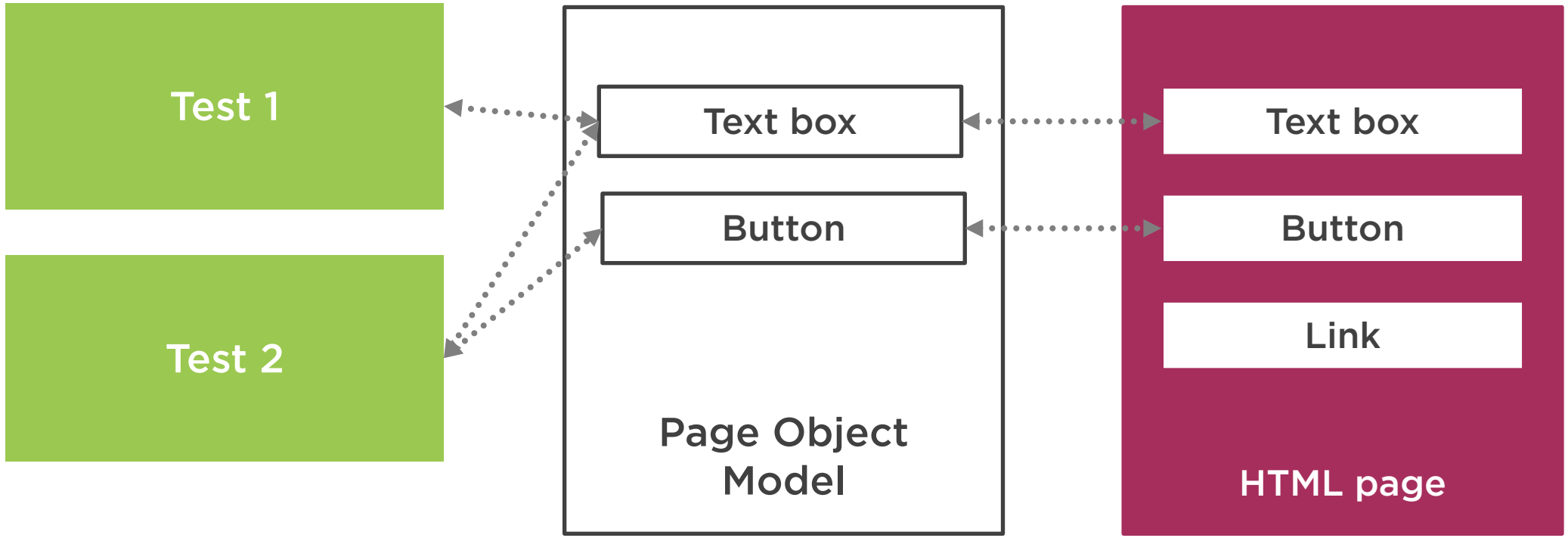
- Concentrate on test intention/logic
- Reduced “clutter” of HTML implementation details

Return/accept fundamental types

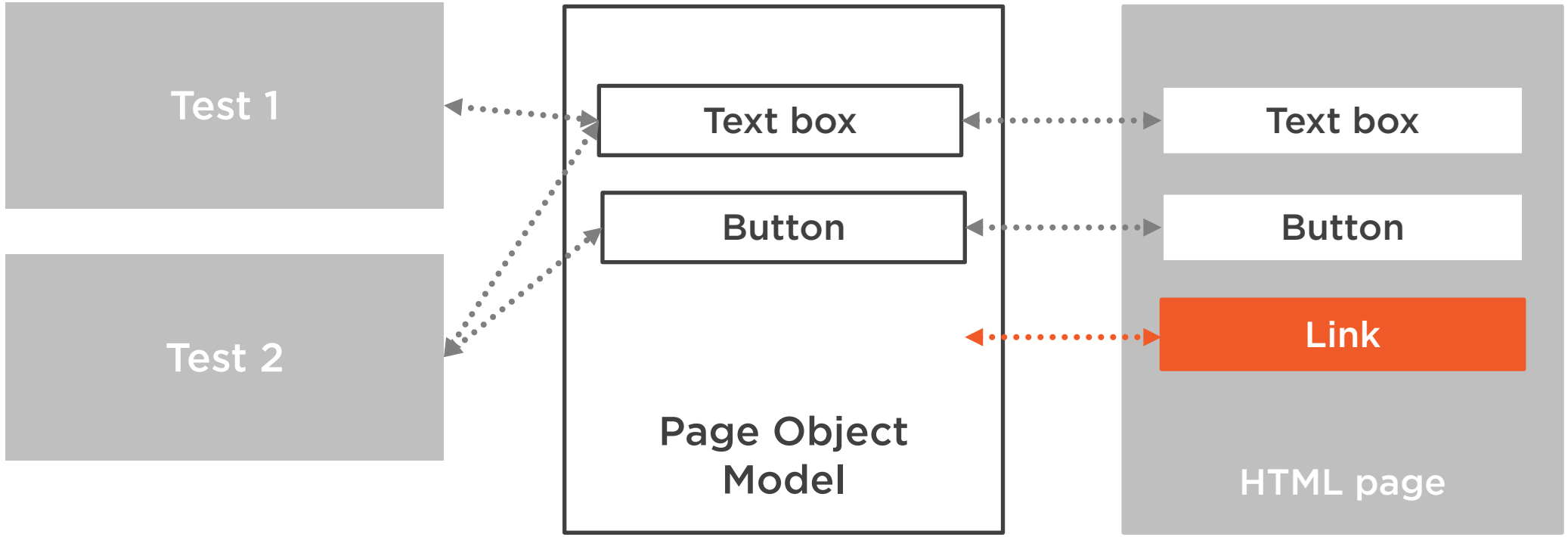
Abstracts away IWebDriver methods

Navigation can return new model

Which HTML Elements Should You Add to a Page Object Model?



Which HTML Elements Should You Add to a Page Object Model?



Only add items to a Page Object Model as they are needed by test code.

Page Object Models Using Selenium Support

```
PM> Install-Package Selenium.Support

using OpenQA.Selenium.Support.PageObjects;


[FindsBy(How = How.Id, Using = "Decision")]
private IWebElement ApplicationDecision { get; set; }

public string Decision => ApplicationDecision.Text;


public ApplicationCompletePage(IWebDriver driver)
{
    Driver = driver;
    PageFactory.InitElements(driver, this);
}
```

Page Object Model Considerations

Page Object Models can become large

- E.g. a dashboard page
- Break down into multiple POMs
- “Page Component Object Models”

No asserts/exceptions in POMs

- Except page load verification
- public void EnsurePageLoaded(...)

Create reusable “user behavior” methods

CreditCardApplicationShould.cs

```
[Fact]
public void BeSubmittedWhenValid()
{
    ...
    applicationPage.EnterFirstName(FirstName);
    applicationPage.EnterLastName(LastName);
    applicationPage.EnterFrequentFlyerNumber(Number);
    applicationPage.EnterAge(Age);
    applicationPage.EnterGrossAnnualIncome(Income)
    applicationPage.ChooseMaritalStatusSingle();
    applicationPage.ChooseBusinessSourceTV();
    applicationPage.AcceptTerms();
    ...
}
```

CreditCardApplicationShould.cs

```
[Fact]
public void BeSubmittedWhenValid()
{
    ...
    EnterApplicationDetails(applicationPage);
    ...
}
```

CreditCardApplicationShould.cs

```
private void EnterApplicationDetails(
    ApplicationPage applicationPage,
    string firstName = "Sarah",
    string lastName = "Smith",
    string frequentFlyerNumber = "123456-A",
    string age = "18",
    string grossAnnualIncome = "50000")
{
    applicationPage.EnterFirstName(firstName);
    applicationPage.EnterLastName(lastName);
    applicationPage.EnterFrequentFlyerNumber(frequentFlyerNumber);
    applicationPage.EnterAge(age);
    applicationPage.EnterGrossAnnualIncome(grossAnnualIncome);
    applicationPage.ChooseMaritalStatusSingle();
    applicationPage.ChooseBusinessSourceTV();
    applicationPage.AcceptTerms();
}

EnterApplicationDetails(applicationPage, lastName:"", age: "17");
```

Treat you browser automation code like production code.

Apply good practices to your browser automation code.

Page Object Models are
just the start.

Readability
Maintainability
Reliability

Summary

- An overview of Page Object Models
- What elements to add to a POM
- Created a new POM class
- Navigated to a POM page
- Refactored tests to use POMs
- Navigated to other POMs
- Encapsulated explicit waits in POMs
- Form filling and submission with POMs
- Tested form validation with POMs
- Reused web driver instance
- Page Object Model considerations

Next:

Course Summary and Further Learning