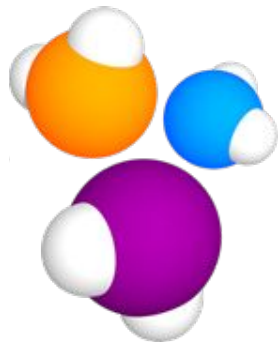




Appium Meet-Up Singapore

Android, iOS and Web testing in a single framework

Tech stack (for this talk)



Scenario: Multiple platforms



Scenario: Multiple platforms



Just a few examples



There are thousands more!

Scenario: Multiple platforms



Handling tests for different platforms **separately is not the most efficient way.**

- (possibly) different test-cases (while actually testing the same)
- (possibly) different frameworks
- (possibly) different integration into the build pipeline
- ...

How can we be more efficient?

What can be reused across platforms?

Selenium



```
WebDriver driver = new FirefoxDriver();
```

```
driver.get("http://www.google.com");
```

```
WebElement element = driver.findElement(By.name("q"));
```

```
element.sendKeys("Appium Meet-up");
```

```
element.submit();
```

```
...
```



Appium



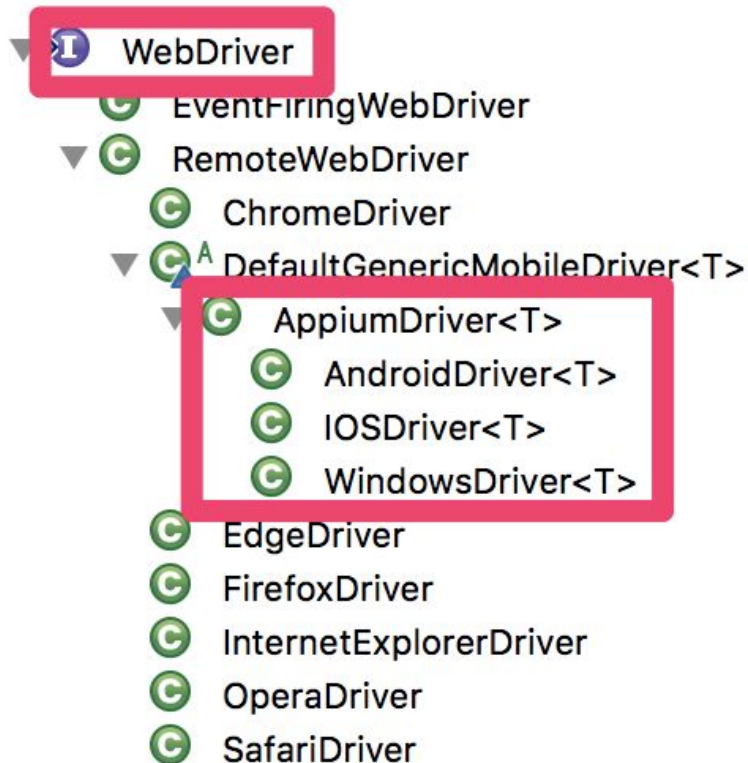
```
AppiumDriver driver = driver = new AppiumDriver(new  
URL("http://127.0.0.1:4723/wd/hub"), capabilities);
```

```
IOSElement element = driver.findElement(By.name("search"));  
element.sendKeys("Appium Meet-up");  
element.submit();
```

```
...
```



Appium and Selenium



Feature files



@web

Scenario Outline: Filter by tags

Given I am on the homepage

When I go to the tags page

And I filter for "<tag>"

And I select the tag "<tag>"

And I select the first question

Then the question is tagged with "<tag>"

Examples:

| tag |

| selenium |

| appium |



Step definitions



```
public class HomeSteps extends BaseSteps {  
    private HomePage home;  
  
    @Given("^I am on the homepage$")  
    public void homepage() {  
        home.load();  
    }  
  
    @When("I search for \"([^\"]*)\"")  
    public void search(String query) {  
        home.search(query);  
    }  
}
```

PageObjects



```
@Component
@Profile(Platform.WEB)
public class HomePage extends BasePage {
    private QuestionsPage questions;
    private TagsPage tags;

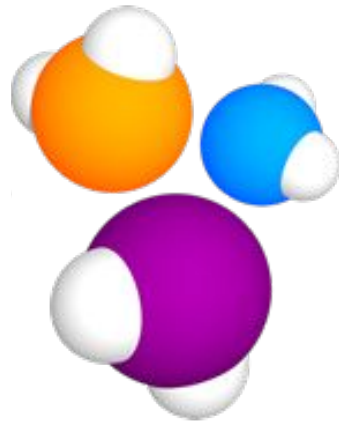
    public QuestionsPage search(String query) {
        $("SEARCH_FIELD").sendKeys(query);
        $("SEARCH_BUTTON").should(appear).click();
        return questions;
    }
}
```

Syntactic sugar with Selenide



```
import static com.codeborne.selenide.Selenide.*;
import static com.codeborne.selenide.Condition.*;
@Test
public void userCanLoginByUsername () {
    open("/login");
    $(By.name("user.name")).setValue("johny");
    $("#submit").click();
    $(".loading_progress").should(disappear);
    $("#username").shouldHave(text("Hello, Johny!"));
}
```

<http://selenide.org/documentation/selenide-vs-selenium.html>



Fluent assertions with AssertJ



```
import static org.assertj.core.api.Assertions.*;

assertThat(frodo.getName()).isEqualTo("Frodo");

assertThat(frodo.getName()).startsWith("Fro")
    .endsWith("do")
    .isEqualToIgnoringCase("frodo");

assertThat(fellowshipOfTheRing) .hasSize(9)
    .contains(frodo, sam)
    .doesNotContain(sauron);

assertThat(frodo.getAge()).as("check %s's age", frodo.getName()).isEqualTo(33);
```

<http://joel-costigliola.github.io/assertj>

Autowiring page objects with Spring



```
@Component
```

```
@Profile(Platform.WEB)
```

```
public class HomePage extends BasePage {
```

```
    private QuestionsPage questions;
```

```
    private TagsPage tags;
```

```
    public QuestionsPage search(String query) {
```

```
        $("SEARCH_FIELD").sendKeys(query);
```

```
        $("SEARCH_BUTTON").should(appear).click();
```

```
        return questions;
```

```
    }
```

```
}
```

Autowiring page objects with Spring contd.



```
public class HomeSteps extends BaseSteps {  
    private HomePage home;
```

```
    @When("I search for \"([^\"]*)\"")  
    public void search(String query) {  
        home.search(query);  
    }  
}
```

That's it! No constructor calls! Just declare and use!

Scenario: Multiple platforms



What can be reused across platforms?

- test cases = feature files probably up to 80-90%
- steps
- page objects probably ~50%

What will usually be different?

- UI identifiers

Platform specific page objects



Spring profiles for the rescue!

@Component

@Profile({ Platform.**WEB**, Platform.**ANDROID** })

public class **TagsPage** **extends** BasePage

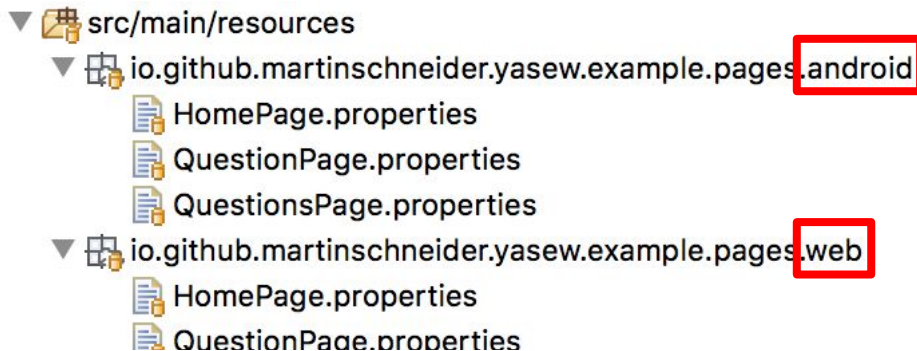
GENERAL settings

platform=web

Platform specific page objects



Identifying UI elements



`SEARCH_FIELD=ID|com.stackexchange.stackoverflow:id/search_src_text`

`ITEM_SCORE=ID|com.stackexchange.stackoverflow:id/question_list_item_score`

`TOOLBAR=ID_com.stackexchange.stackoverflow:id/toolbar`

```
$ ("SEARCH_FIELD") .sendKeys (query) ;
```

```
$ ("SEARCH_BUTTON") .should (appear) .click () ;
```

Try it yourself!



<https://martinschneider.github.io/yasew/>

yasew

Dynamic test framework for web and mobile applications

View on GitHub





Questions



Appium Meet-Up Singapore

Image-based testing with Appium and OpenCV



Questions