carousell

APPIUM WORKSHOP TAIWAN

WIFI : Carousell_Welcome
Password : hellowelcome

Self Introduction

- **Software Engineer in Test @ Carousell TW**
  - Buyer Experience Team
  - QA Team
- **Previously worked @ Trend Micro**
  - PC-Cillin
  - Password Manager

Outline

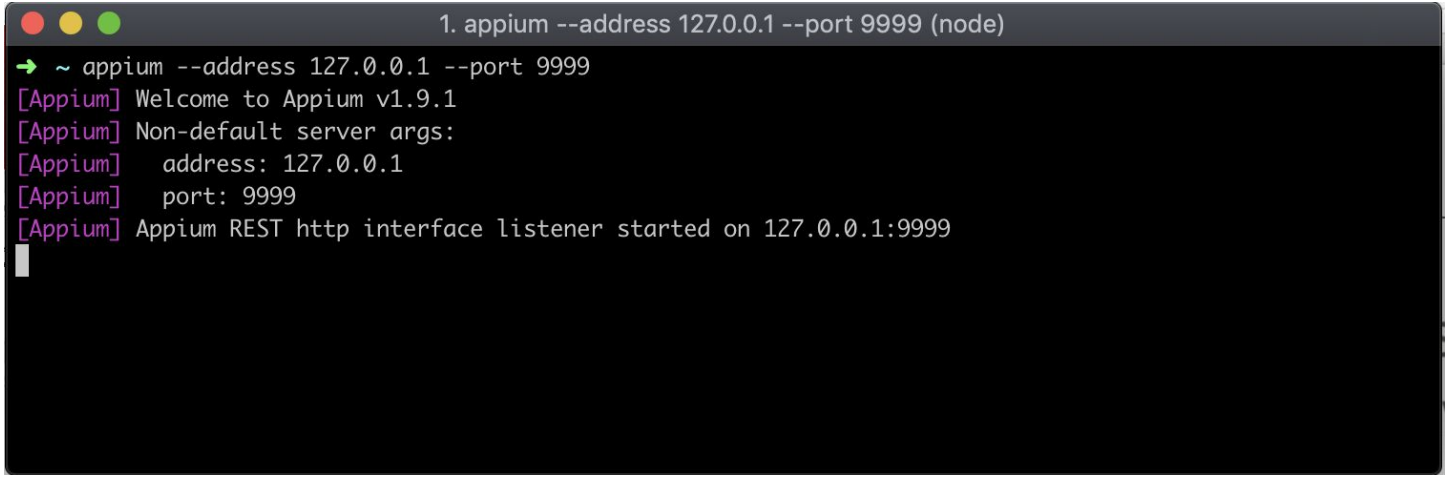
- Tool Introduction
- Set up Appium environment
- Set up Android emulator
- Appium Desktop
- Selector / Locator Strategy
- First Java Project
- Before Test Automation Framework
- Example 1~5
- Appium Pros and Cons
- Final Practice

- **Register your personal account on Carousell**

1. Install Carousell App from App Store / Google Store
2. Open Carousell App
3. Click “**Sign up with Email**” button
4. Enter Email, Username, Password and City
5. Click “**Create my account**”
6. Show Carousell Home page

TOOL INTRODUCTION

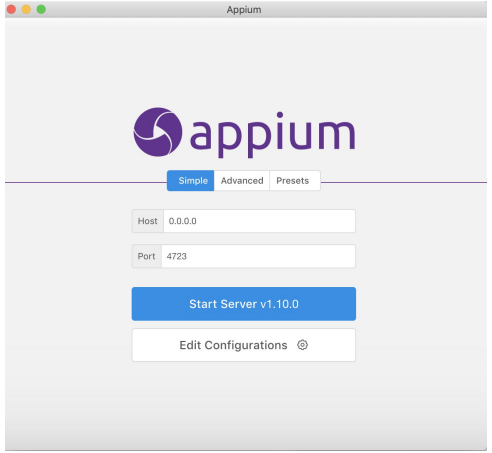
Appium

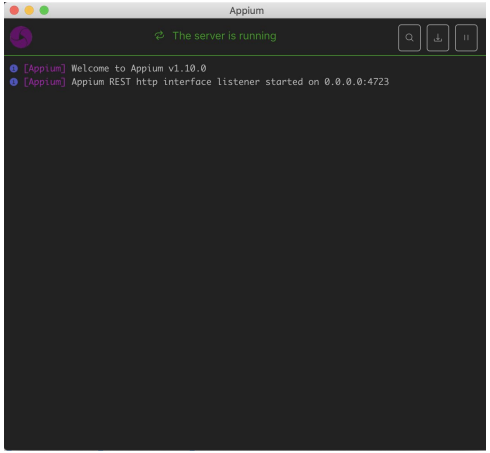


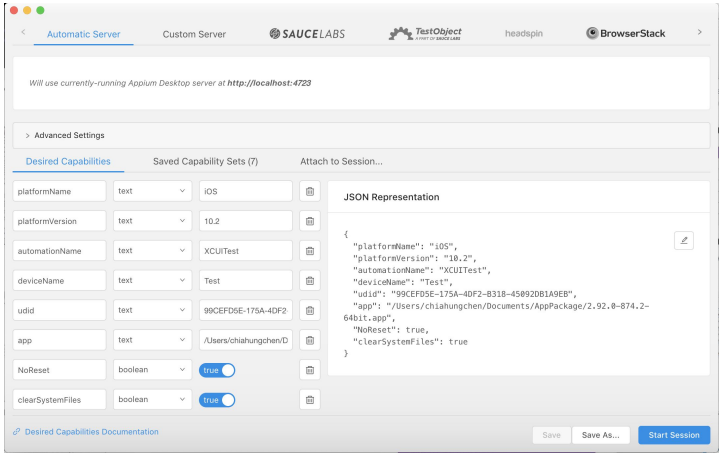
https://www.youtube.com/watch?v=IOSUBda2-g4

Appium Desktop

appium desktop is a point-and-click interface for using appium and inspecting your app's structure







https://www.youtube.com/watch?v=IOSUBda2-g4

Appium and Appium Desktop

Appium CLI	Appium Desktop
Install via NPM	Install by download & DMG (mac) / EXE (win)
Launch from terminal	Launch by clicking
Set parameters as command-line flags	Set parameters via GUI options
Build desired capabilities and start sessions from code only	Build desired capabilities and start sessions from code or GUI
No visual app inspector	Built-in visual app inspector
Runs on Mac, Windows, Linux	Runs on Mac, Windows, Linux

https://www.youtube.com/watch?v=IOSUBda2-g4

Requirements of Appium

- **Appium**

  Appium CLI : Run Appium by command line

  Appium Desktop : Appium GUI console

  Appium-doctor : Diagnose and fix common Node, iOS and Android configuration issues before starting Appium.

- ~~**iOS**~~

- **Android**

  Android Studio - IDE for Android, Tools and Emulators

  Java JDK - Java Development Kit

  (Optional) Gradle - Build Tool for test and dependency management

Other requirements

- **Workshop**

  (Optional) Homebrew : Package management software on Mac

  (Optional) Chocolatey : Package management software on Windows

  Git : A tool / system for version control on your source code history

  Node.js : A JavaScript runtime built on Chrome's V8 JavaScript engine

  (Optional) Intellij IDEA Community : Java IDE

SET UP
APPIUM ENVIRONMENT

Set up Appium environment

- ## Mac

1. Install Package management Tool

   Install Homebrew by link below (https://brew.sh/index_zh-tw)

   brew --version
   brew --help
   brew cask

2. Install Node.js by brew
   brew install node
   node --version

- ## Window

1. Install Package management Tool

   Install Chocolatey by link below (https://chocolatey.org/install#installing-chocolatey)

   Run command prompt as Admin
   choco --version
   choco --help

2. Install Node.js by choco
   choco install nodejs
   refreshenv
   node --version

Set up Appium environment

- **Mac**

3. Install Appium CLI by npm (http://appium.io/)

npm install -g appium
appium --version
appium --help
appium

- **Window**

3. Install Appium CLI by npm (http://appium.io/)

npm install -g appium
appium --version
appium --help
appium

```
Carousells-MacBook-Pro:~ carousell$ appium --version
1.10.0
Carousells-MacBook-Pro:~ carousell$ appium
[Appium] Welcome to Appium v1.10.0
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
```

Set up Appium environment

- **Mac**

4. Install Appium-doctor by npm (https://github.com/appium/appium-doctor)
   npm install -g appium-doctor
   appium-doctor

- **Window**

4. Install Appium-doctor by npm (https://github.com/appium/appium-doctor)
   npm install -g appium-doctor
   appium-doctor

```
Carousells-MacBook-Pro:~ carousell$ appium-doctor
info AppiumDoctor Appium Doctor v.1.6.0
info AppiumDoctor ### Diagnostic starting ###
info AppiumDoctor  ✔ The Node.js binary was found at: /usr/local/bin/node
info AppiumDoctor  ✔ Node version is 11.4.0
info AppiumDoctor  ✔ Xcode is installed at: /Library/Developer/CommandLineTools
info AppiumDoctor  ✔ Xcode Command Line Tools are installed.
info AppiumDoctor  ✔ DevToolsSecurity is enabled.
info AppiumDoctor  ✔ The Authorization DB is set up properly.
WARN AppiumDoctor  ✖ Carthage was NOT found!
info AppiumDoctor  ✔ HOME is set to: /Users/carousell
WARN AppiumDoctor  ✖ ANDROID_HOME is NOT set!
WARN AppiumDoctor  ✖ JAVA_HOME is NOT set!
WARN AppiumDoctor  ✖ adb could not be found because ANDROID_HOME is NOT set!
WARN AppiumDoctor  ✖ android could not be found because ANDROID_HOME is NOT set!
WARN AppiumDoctor  ✖ emulator could not be found because ANDROID_HOME is NOT set!
WARN AppiumDoctor  ✖ Bin directory for $JAVA_HOME is not set
info AppiumDoctor ### Diagnostic completed, 7 fixes needed. ###
info AppiumDoctor
info AppiumDoctor ### Manual Fixes Needed ###
info AppiumDoctor The configuration cannot be automatically fixed, please do the following first:
WARN AppiumDoctor - Please install Carthage. Visit https://github.com/Carthage/Carthage#installing-carthage for more i
nformation.
WARN AppiumDoctor - Manually configure ANDROID_HOME.
WARN AppiumDoctor - Manually configure JAVA_HOME.
WARN AppiumDoctor - Manually configure ANDROID_HOME and run appium-doctor again.
WARN AppiumDoctor - Add '$JAVA_HOME/bin' to your PATH environment
info AppiumDoctor ###
info AppiumDoctor
info AppiumDoctor Bye! Run appium-doctor again when all manual fixes have been applied!
info AppiumDoctor
Carousells-MacBook-Pro:~ carousell$
```

Set up Appium environment

- **Mac**

5. Install Appium Desktop
   (https://github.com/appium/appium-desktop)
   brew cask install appium

6. Install Java JDK
   brew cask install java
   java --version

7. Install Android Studio and Launch
   brew cask install android-studio
   Launch and download components !

8. Install IDE
   brew cask install intellij-idea-ce

- **Window**

5. Install Appium Desktop
   (https://github.com/appium/appium-desktop)
   choco install appium-desktop

6. Install Java JDK
   choco install jdk11
   refreshenv
   java --version

7. Install Android Studio
   choco install androidstudio
   Launch and download components !

8. Install IDE
   choco install intellijidea-community

Set up Appium environment

- ## Mac

9. Set up environment variable

vim ~/.bash_profile

export JAVA_HOME=
$(/usr/libexec/java_home)

export ANDROID_HOME =
/Users/…/Library/Android/sdk

export PATH=
$PATH:$JAVA_HOME/bin:$ANDROID_HOME

Save and restart Terminal !

- ## Window

9. Set up environment variable

Open **advanced system settings**
-> **Environment Variables**
-> **System variable**

Add **ANDROID_HOME** =
“C:\Users\WorkShop\AppData\Local\Android\sdk”

Add %ANDROID_HOME% into PATH

Save and refreshenv !

Set up Appium Environment

- **Mac**

10. Check appium settings

appium-doctor

- **Window**

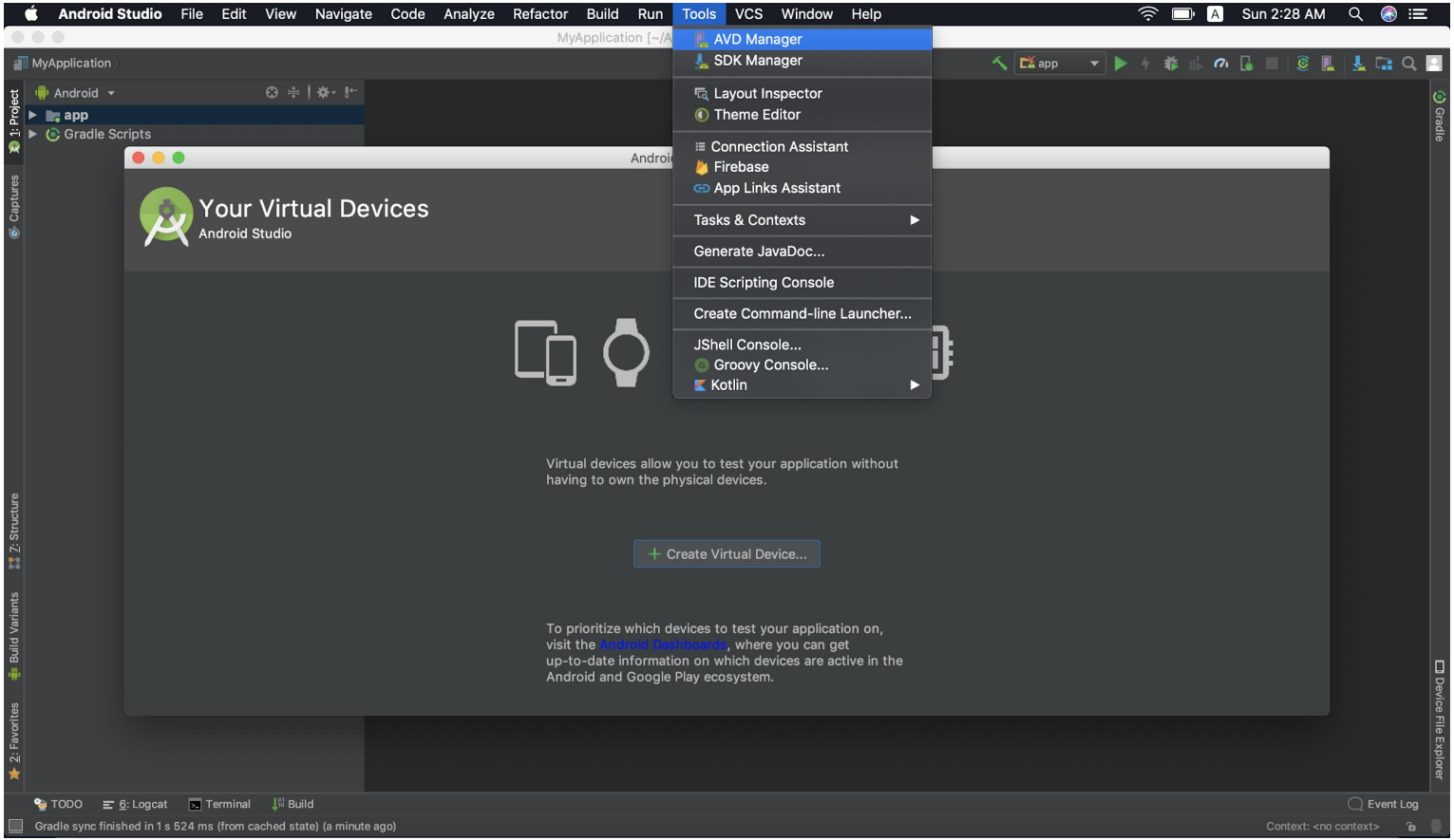
10. Check appium settings

appium-doctor

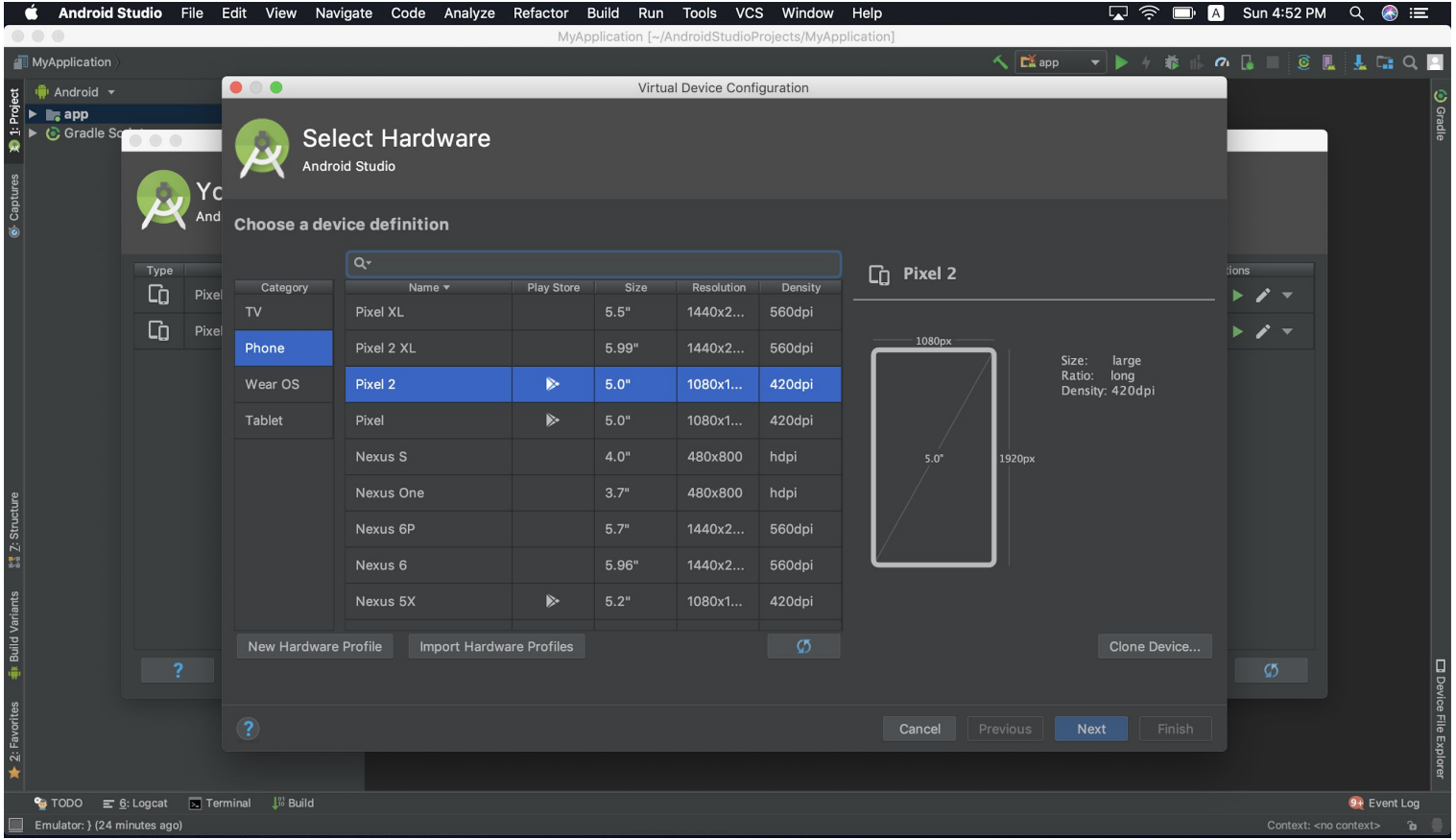
```
1. bash
Last login: Sun Dec 16 01:27:45 on ttys000
Carousells-MBP:~ carousell$ appium-doctor
info AppiumDoctor Appium Doctor v.1.6.0
info AppiumDoctor ### Diagnostic starting ###
info AppiumDoctor  ✔ The Node.js binary was found at: /usr/local/bin/node
info AppiumDoctor  ✔ Node version is 11.4.0
info AppiumDoctor  ✔ Xcode is installed at: /Library/Developer/CommandLineTools
info AppiumDoctor  ✔ Xcode Command Line Tools are installed.
info AppiumDoctor  ✔ DevToolsSecurity is enabled.
info AppiumDoctor  ✔ The Authorization DB is set up properly.
WARN AppiumDoctor  ✖ Carthage was NOT found!
info AppiumDoctor  ✔ HOME is set to: /Users/carousell
info AppiumDoctor  ✔ ANDROID_HOME is set to: /Users/carousell/Library/Android/sdk
info AppiumDoctor  ✔ JAVA_HOME is set to: /Library/Java/JavaVirtualMachines/openjdk-11.0.1.jdk/Contents/Home
info AppiumDoctor  ✔ adb exists at: /Users/carousell/Library/Android/sdk/platform-tools/adb
info AppiumDoctor  ✔ android exists at: /Users/carousell/Library/Android/sdk/tools/android
info AppiumDoctor  ✔ emulator exists at: /Users/carousell/Library/Android/sdk/tools/emulator
info AppiumDoctor  ✔ Bin directory of $JAVA_HOME is set
info AppiumDoctor ### Diagnostic completed, one fix needed. ###
info AppiumDoctor
info AppiumDoctor ### Manual Fixes Needed ###
info AppiumDoctor The configuration cannot be automatically fixed, please do the following first:
WARN AppiumDoctor - Please install Carthage. Visit https://github.com/Carthage/Carthage#installing-carthage for more information.
info AppiumDoctor ###
info AppiumDoctor
info AppiumDoctor Bye! Run appium-doctor again when all manual fixes have been applied!
info AppiumDoctor
Carousells-MBP:~ carousell$
```

SET UP
ANDROID EMULATOR

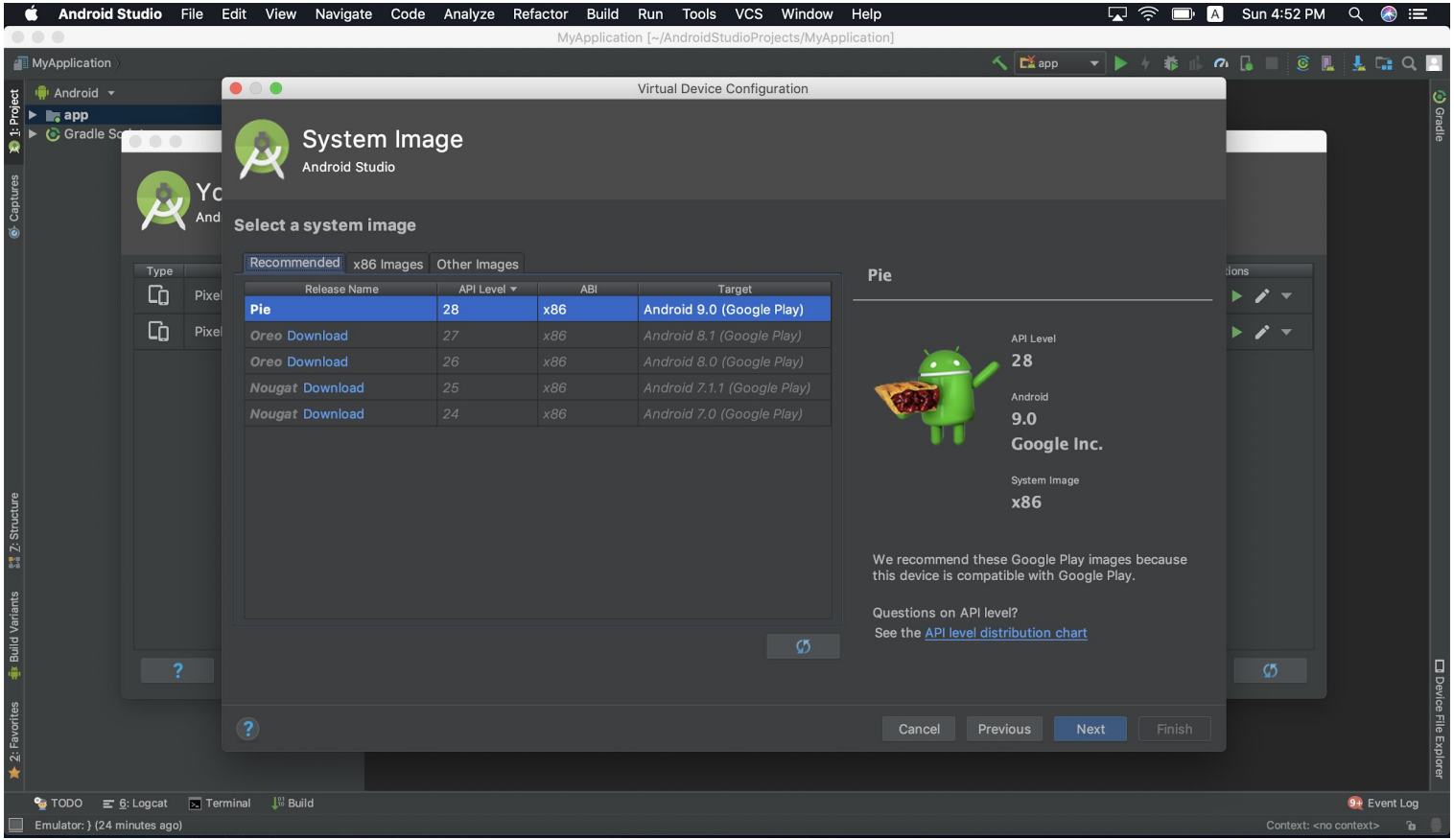
Set up Android Emulator



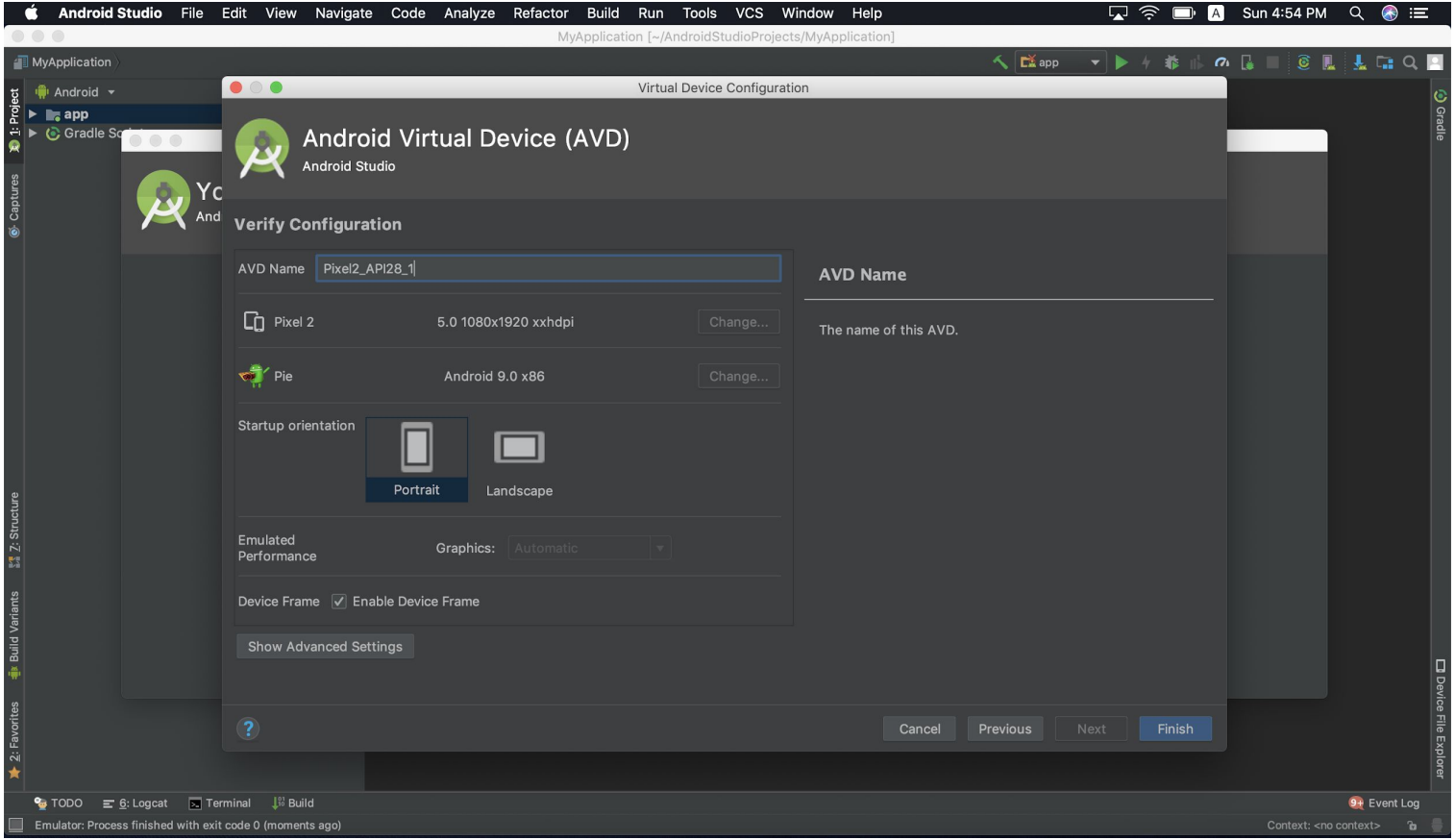
Set up Android Emulator



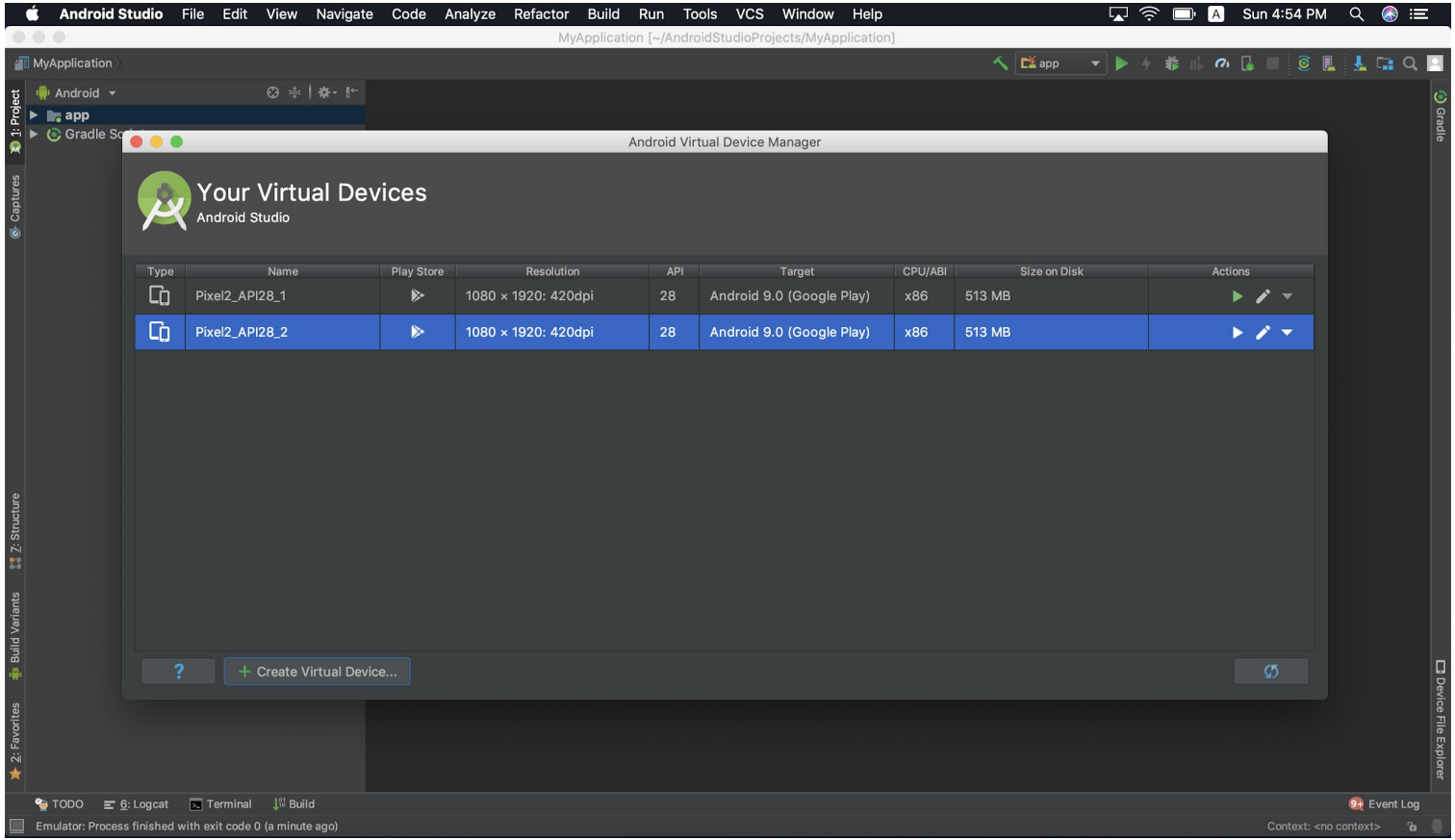
Set up Android Emulator



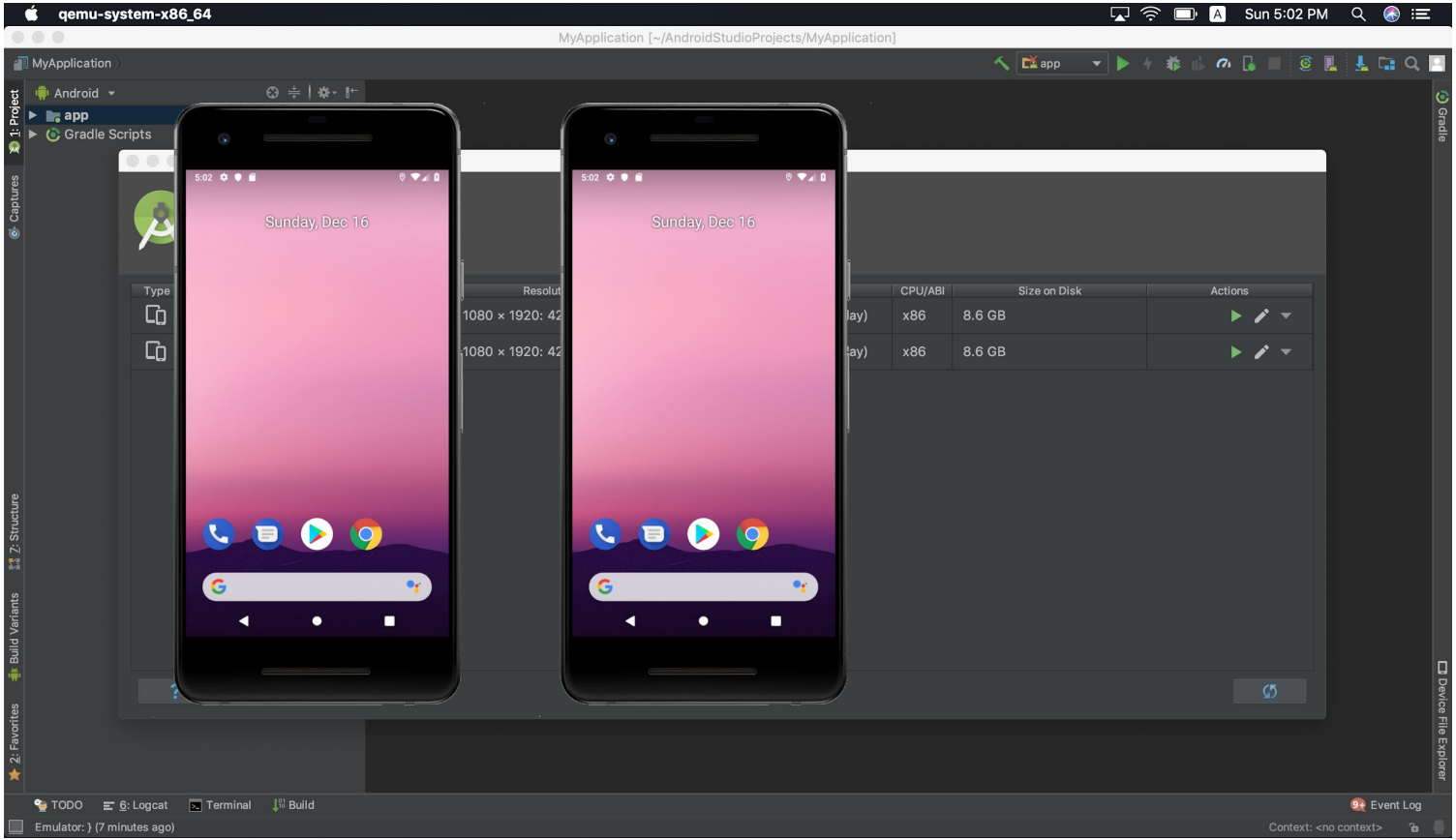
Set up Android Emulator



Set up Android Emulator



Set up Android Emulator



Set up Android Emulator

- **adb** (https://developer.android.com/studio/command-line/adb)

  Check connected android devices : adb devices

```
Carousells-MacBook-Pro:~ carousell$ adb devices
List of devices attached
emulator-5554   device
emulator-5556   device
```

  Add $ANDROID_PLATFORMTOOLS_HOME into PATH

  export ANDROID_PLATFORMTOOLS_HOME=$ANDROID_HOME/platform_tools

```
1. bash
Carousells-MacBook-Pro:~ carousell$ cat ~/.bash_profile
export JAVA_HOME=$(/usr/libexec/java_home)
export ANDROID_HOME=/Users/carousell/Library/Android/sdk
export ANDROID_PLATFORM_TOOLS_HOME=$ANDROID_HOME/platform-tools
export PATH=$PATH:$JAVA_HOME/bin:$ANDROID_HOME:$ANDROID_PLATFORM_TOOLS_HOME
```

Set up Android Emulator

- **emulator** (https://developer.android.com/studio/run/emulator-commandline)

  Show emulators : **emulator -list-avds**

  Launch emulator : **emulator -avd Pixel2_API28_2**

  Add **$ANDROID_EMULATOR** into **PATH**

  export ANDROID_EMULATOR=$ANDROID_HOME/emulator

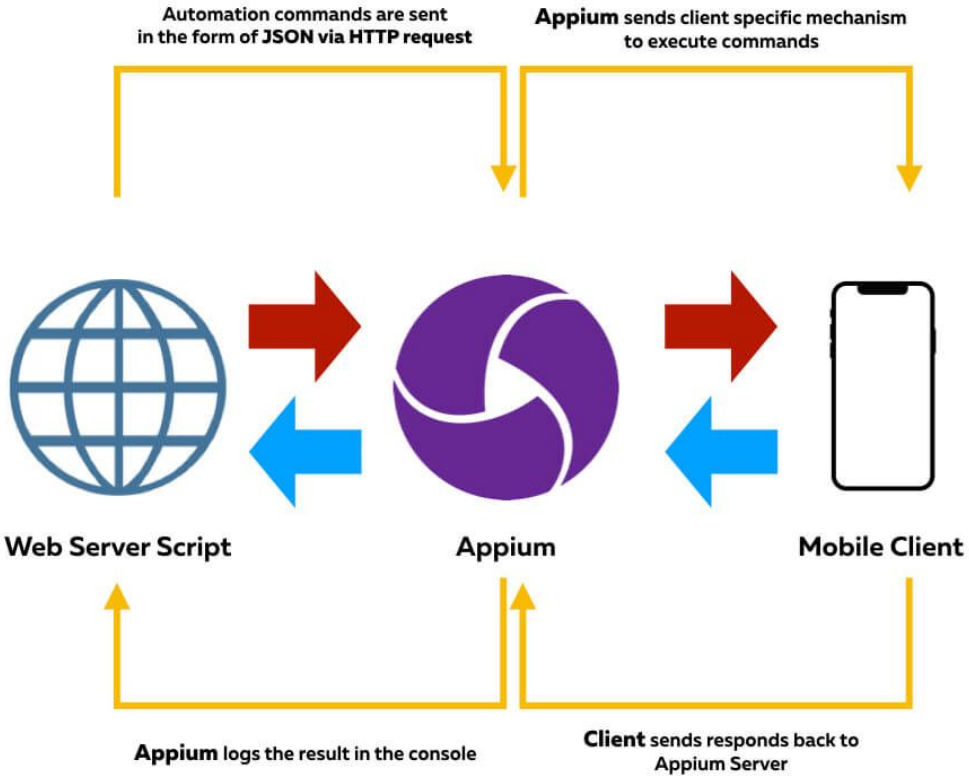
```
1. bash
Carousells-MBP:~ carousell$ cat ~/.bash_profile
export JAVA_HOME=$(/usr/libexec/java_home)
export ANDROID_HOME=$HOME/Library/Android/sdk
export ANDROID_PLATFORMTOOLS=$ANDROID_HOME/platform-tools
export ANDROID_TOOLS=$ANDROID_HOME/tools
export ANDROID_EMULATOR=$ANDROID_HOME/emulator
export PATH=$PATH:$JAVA_HOME/bin:$ANDROID_HOME:$ANDROID_EMULATOR:$ANDROID_PLATFORMTOOLS:$ANDROID_TOOLS
```

Set up Android Emulator

- Practice

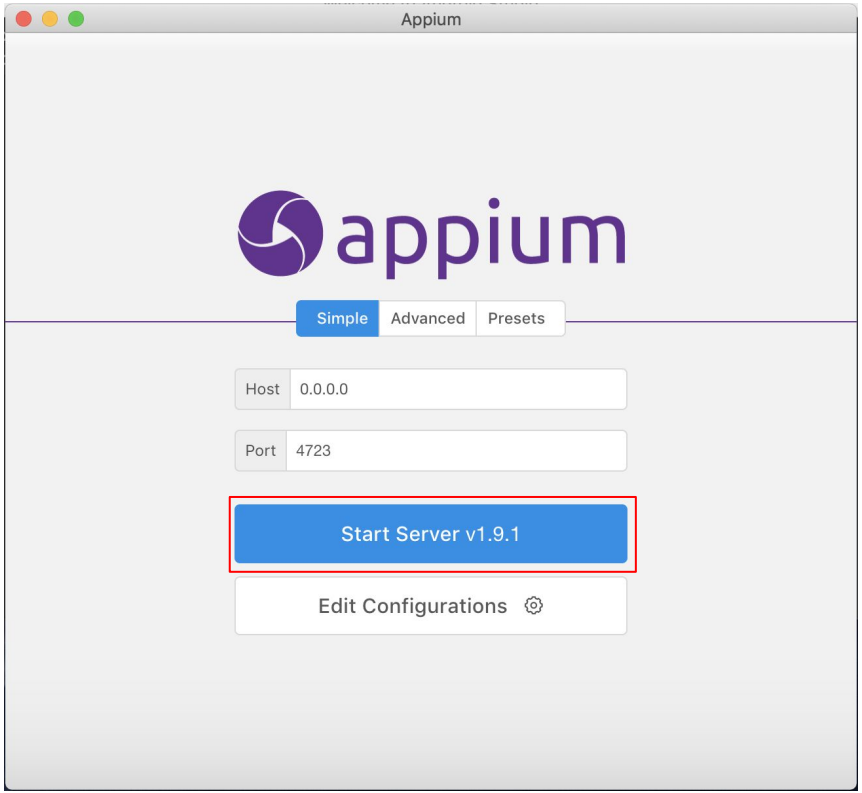
APPIUM DESKTOP

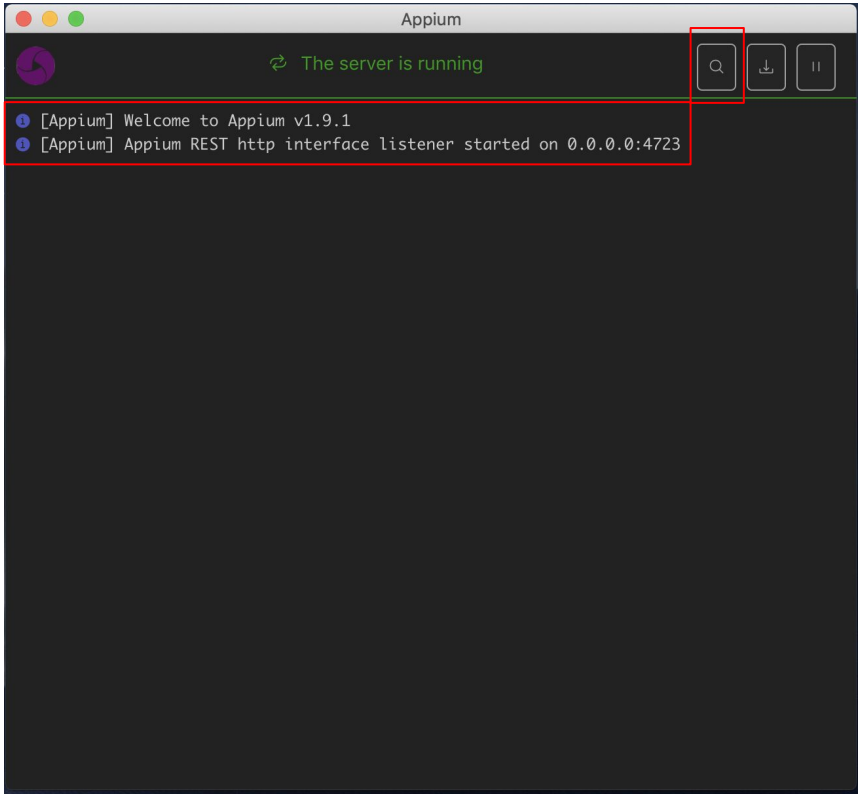
Appium Desktop



https://www.appcoda.com/automated-ui-testing-appium/

Appium Desktop





Appium Desktop

- **Desired Capabilities** (https://appium.io/docs/en/writing-running-appium/caps/)

General

Android Only

Automatic Server | Custom Server | SAUCELABS | TestObject | headspin | BrowserStack

Will use currently-running Appium Desktop server at ***http://localhost:4723***

> Advanced Settings

Desired Capabilities | Saved Capability Sets (7) | Attach to Session...

deviceName	text	emulator-5554
platformName	text	Android
platformVersion	text	8.0.0
fullReset	boolean	true
app	text	/Users/chiahungchen/D
appActivity	text	com.thecarousell.Caro
appPackage	text	com.thecarousell.Caro
newCommandTimeout	number	60000
udid	text	emulator-5554
automationName	text	UiAutomator2
avd	text	Pixel2_API26

JSON Representation

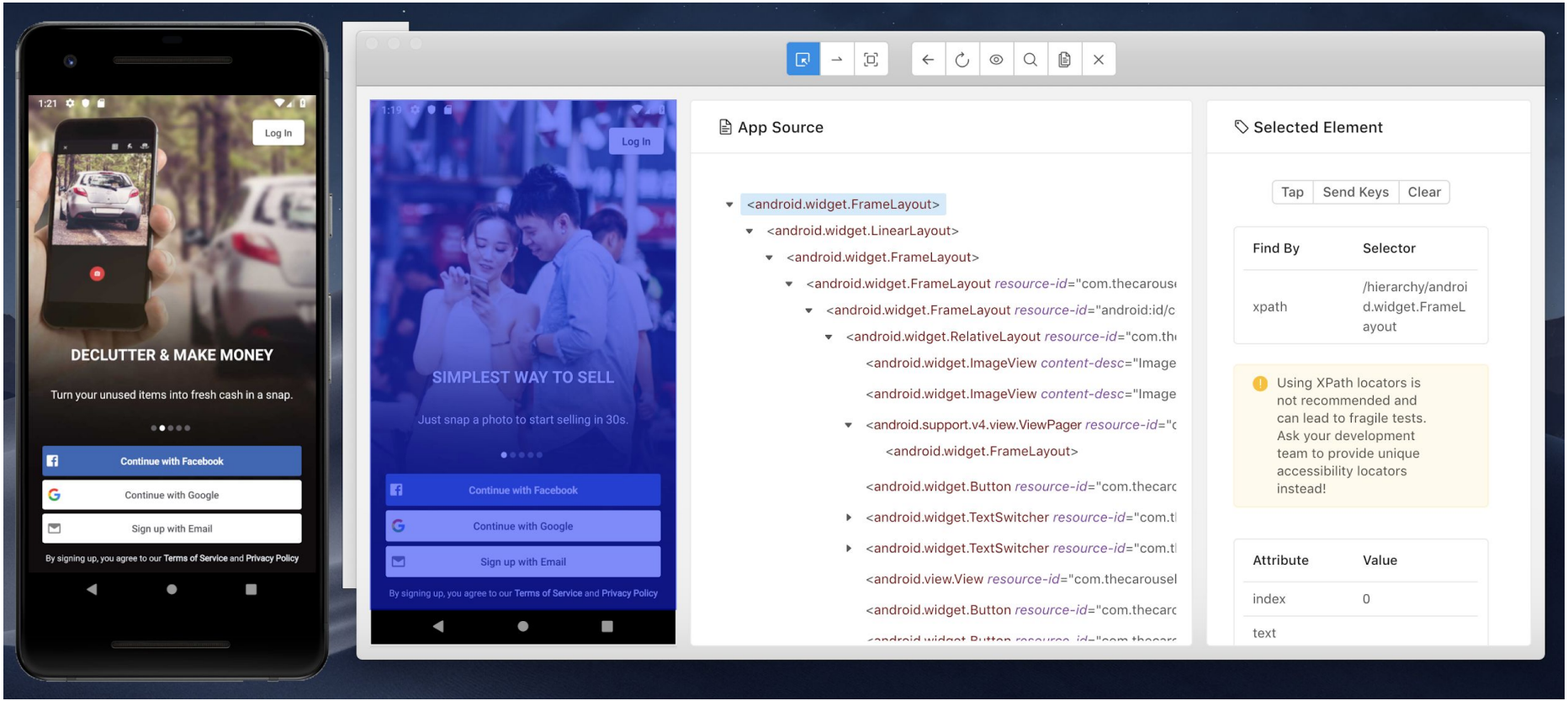
```
{
  "deviceName": "emulator-5554",
  "platformName": "Android",
  "platformVersion": "8.0.0",
  "fullReset": true,
  "app": "/Users/chiahungchen/Documents/AppPackage/2.92.217.196-1384.apk",
  "appActivity": "com.thecarousell.Carousell.activities.EntryActivity",
  "appPackage": "com.thecarousell.Carousell",
  "newCommandTimeout": 60000,
  "udid": "emulator-5554",
  "automationName": "UiAutomator2",
  "avd": "Pixel2_API26"
}
```

Desired Capabilities Documentation | Save | Save As... | Start Session

Appium Desktop

```
{
  "deviceName": "emulator-5554",
  "platformName": "Android",
  "platformVersion": "9.0.0",
  "fullReset": true,
  "app": "/Users/chiahungchen/Documents/AppPackage/2.92.217.196-1384.apk",
  "appActivity": "com.thecarousell.Carousell.activities.EntryActivity",
  "appPackage": "com.thecarousell.Carousell",
  "newCommandTimeout": 60000,
  "udid": "emulator-5554",
  "automationName": "UiAutomator2",
  "avd": "Pixel2_API26"
}
```

Appium Desktop

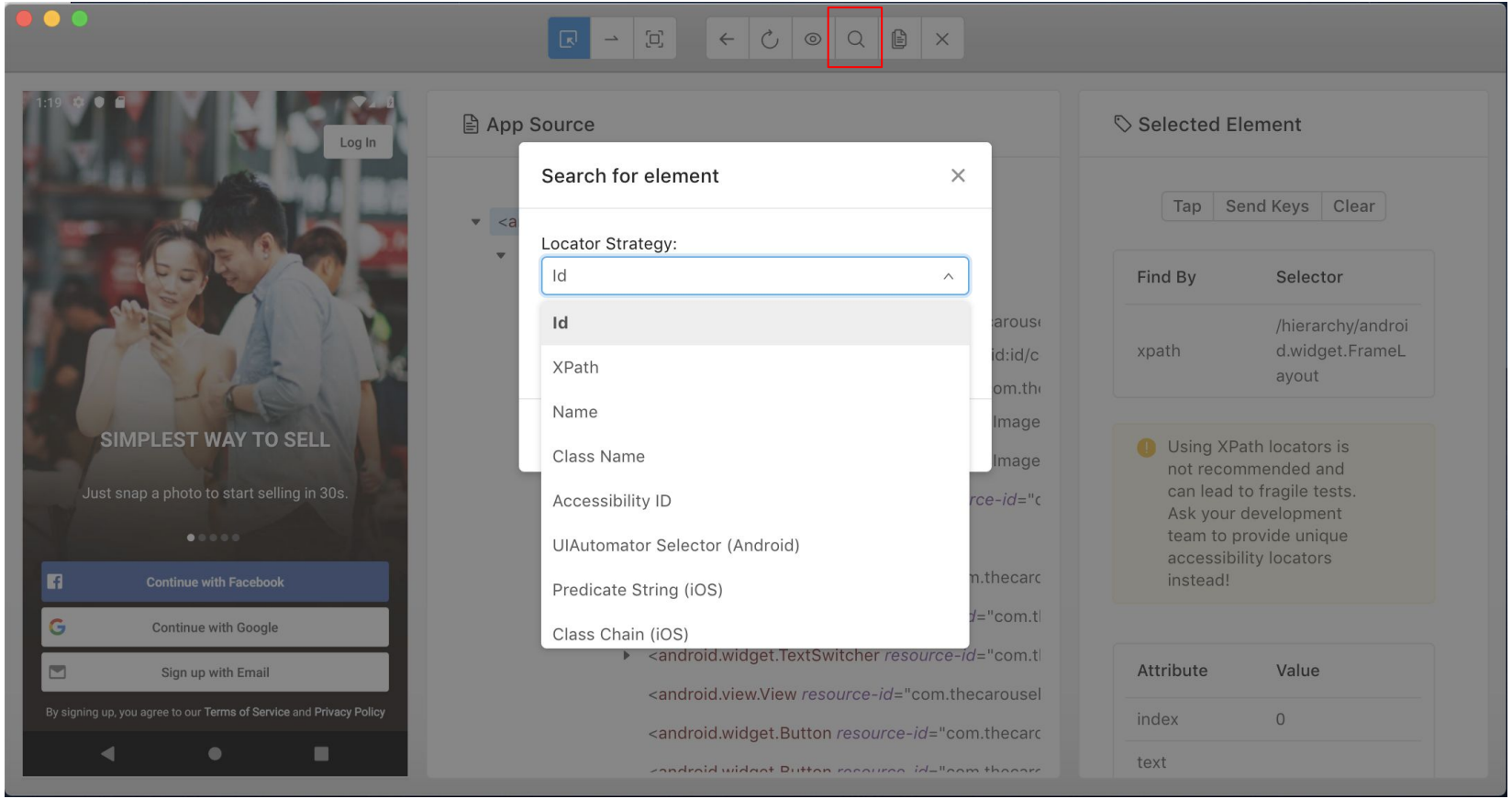


Appium Desktop

- **Practice**

SELECTOR / LOCATOR STRATEGY

Selector / Locator Strategy



Selector / Locator Strategy

- https://appium.io/docs/en/commands/element/find-elements/index.html#selector-strategies

- **ID** : Resource-id=”com.thecarousell.Carousell:id/action_bar_root”

- **Accessibility ID** : Content-desc=”image”

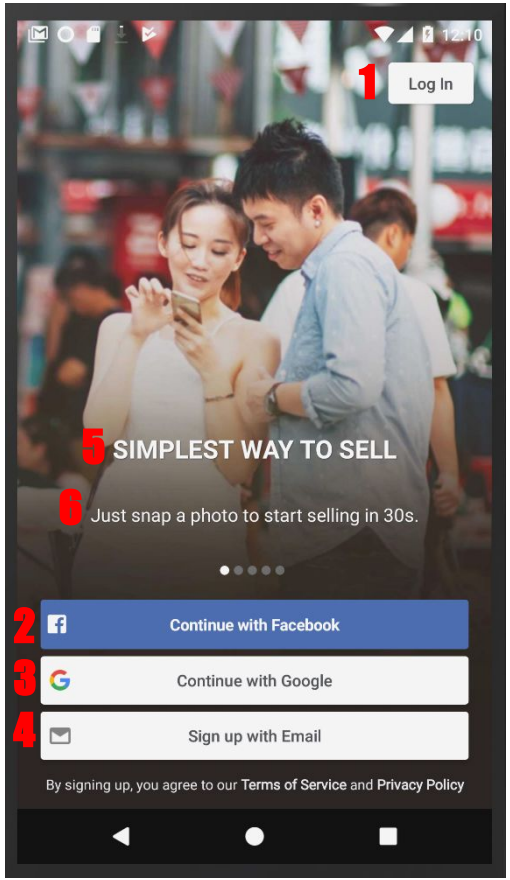
- **Class** : android.widget.TextView

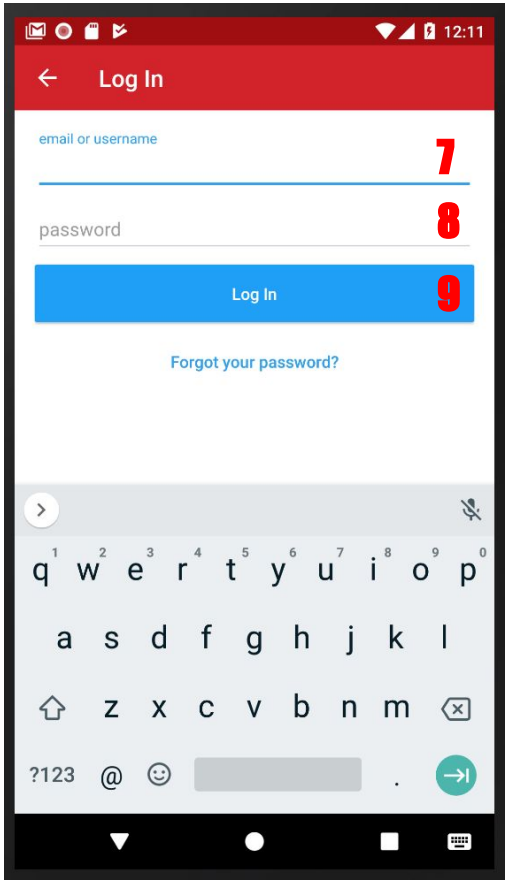
- **XPATH**
  //android.widget.RelativeLayout/android.widget.Button[2]
  //*[@text=”Continue with Facebook”]
  //android.widget.button[contains(@text,”Continue with”)]

- **UIAutomator Selector**
  .className(“android.widget.Button”).instance(1)
  .className(“android.widget.Button”).text(“Continue with Facebook”)
  .className(“android.widget.Button”).textContains(“Continue with”)

Selector / Locator Strategy

- Practice





FIRST JAVA PROJECT

PREPARATION

- **Mac**

1. Install Gradle

   brew install gradle

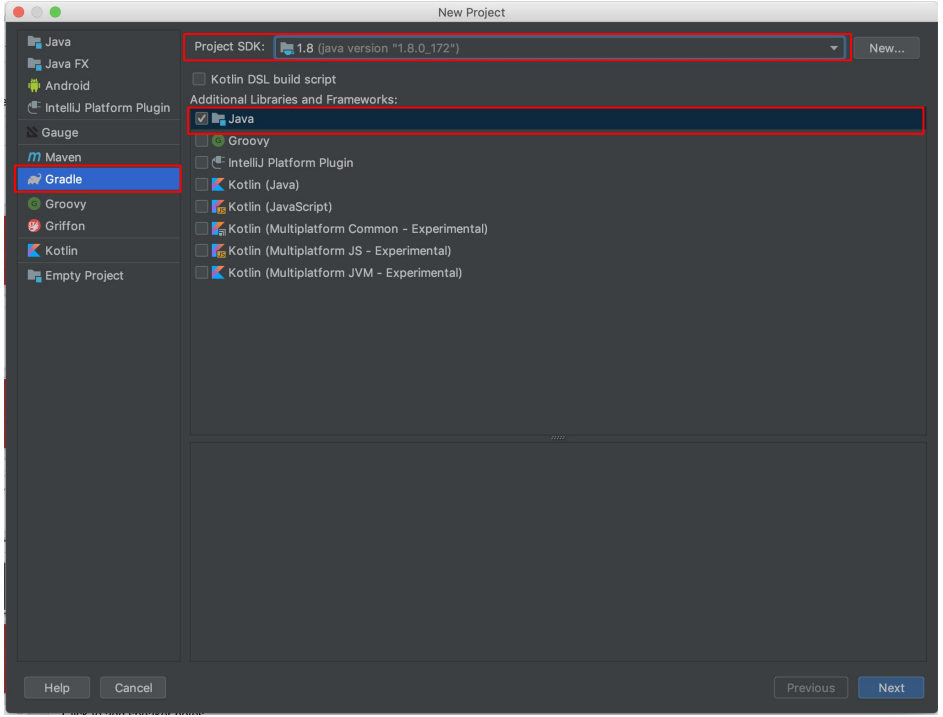
- **Window**

1. Install Gradle

   Run command prompt as Admin
   choco install gradle

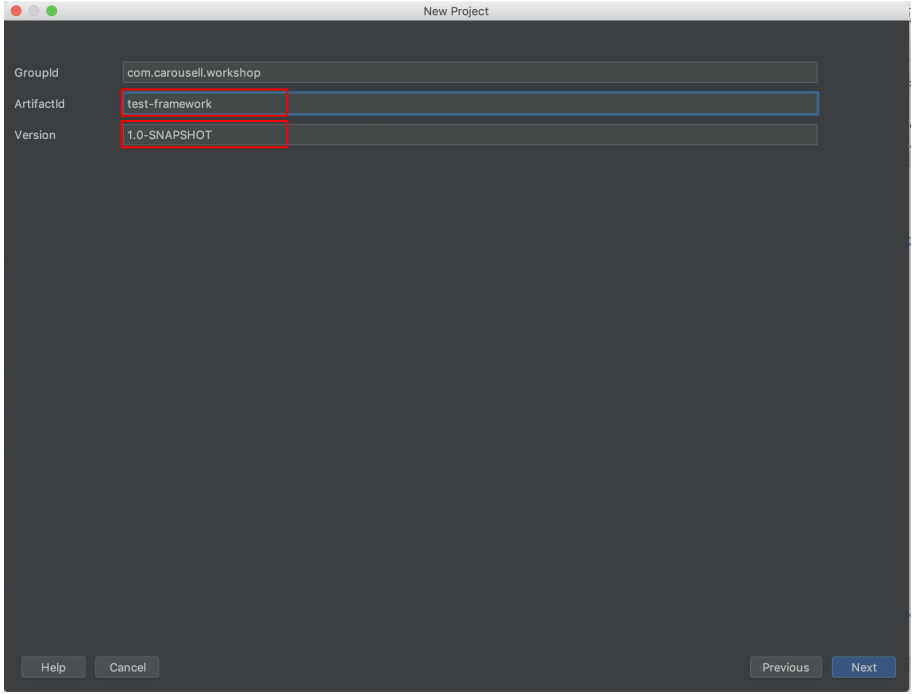
First Java Project

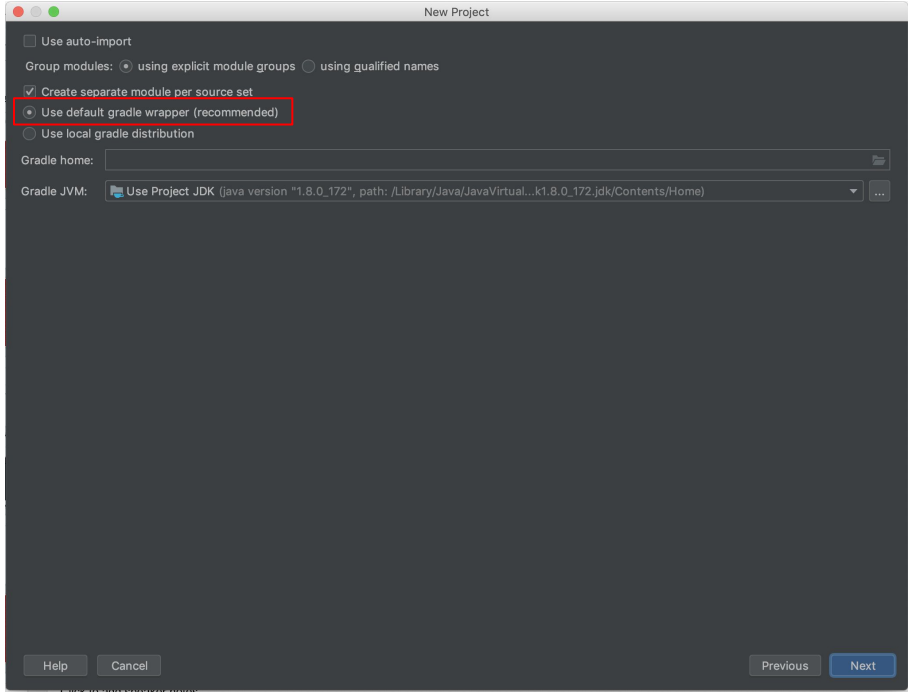
- Create New Project -> Select **Gradle** and **Java SDK**



First Java Project

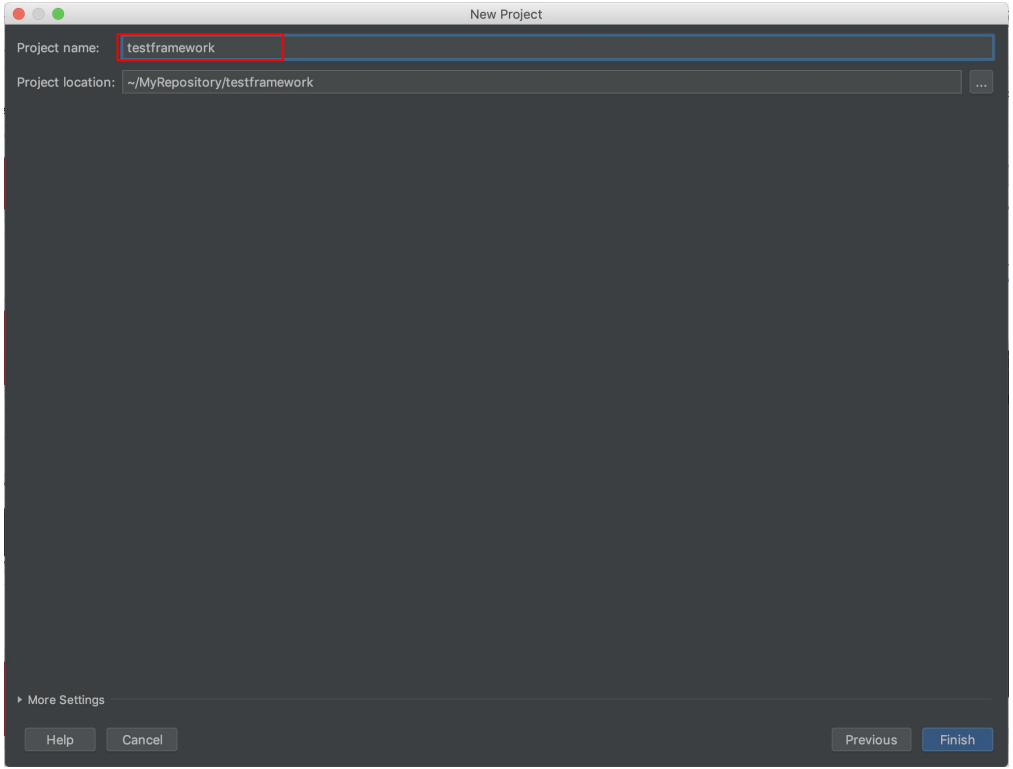
- Set up your **GroupId** and **ArtifactId**



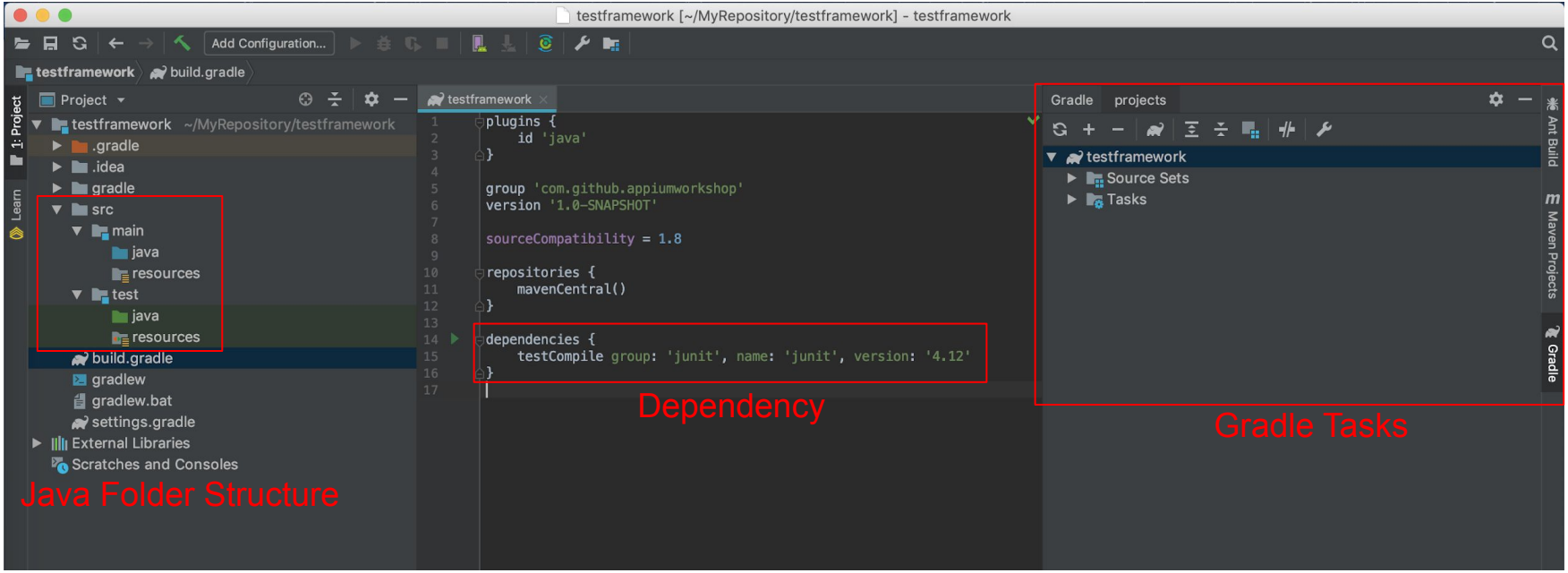


First Java Project

- Set up your **project name**



First Java Project



BEFORE
TEST AUTOMATION FRAMEWORK

PREPARATION

- ## Mac

1. Install Git

   brew install git

2. Clone repository

   git clone https://github.com/ChenChiaHung/PracticeAppium.git

   cd PracticeAppium

- ## Window

1. Install Git

   Run command prompt as Admin
   choco install git

2. Clone repository

   git clone https://github.com/ChenChiaHung/PracticeAppium.git

   cd PracticeAppium

BEFORE AUTOMATION FRAMEWORK

- **Git command** (https://git-scm.com/book/zh/v2)

git clone %SSH or HTTPS%	Clone your repository
git branch	Display branches and current branch
git checkout %branch_name%	Switch branch
git stash	Save changes and revert current branch

BEFORE AUTOMATION FRAMEWORK

- https://github.com/ChenChiaHung/PracticeAppium

PracticeAppium

There are 5 branches for Appium workshop, each branch has different purposes below.

Example_01

Use Gradle, Junit4, Log4j2 and Appium java-client on test automation framework

Example_02

Write a simple test scenario with several action controls

Example_03

Implement POM (Page Object Model)

Example_04

Implement an action class for providing unified action control

Example_05

Implement PageFactory

EXAMPLE 1
Basic Test Framework

Basic Test Framework

- **git checkout Example_01**

- Use Gradle, Junit4, Log4j2 on test automation framework
  - Gradle : Build Tool for testing, solving dependency….
    https://gradle.org/
  - Junit4 : Test Framework
    https://junit.org/junit4/
  - Log4j2 : Log feature
    https://logging.apache.org/log4j/2.x/maven-artifacts.html

Basic Test Framework

- Use Intellij IDEA to open “PracticeAppium” folder

Import Project from Gradle

Gradle project: ~/PracticeAppium

Use auto-import

Create directories for empty content roots automatically

Group modules: using explicit module groups using qualified names

Create separate module per source set

Use default gradle wrapper (not configured for the current project)

Use gradle 'wrapper' task configuration Gradle wrapper customization in build script

Use local gradle distribution

Gradle home:

Gradle JVM: Use Project JDK (11, path: /Library/Java/JavaVirtual...jdk-11.0.1.jdk/Contents/Home)

Project format: .idea (directory based)

Global Gradle settings

Help Cancel OK

Basic Test Framework

- Add new artifacts in dependency

myFirstAppiumProject [~/MyRepository/PracticeAppium] - myFirstAppiumProject

```
group 'com.github.chenchiahung'
version '1.0-SNAPSHOT'

apply plugin: 'java'

sourceCompatibility = 1.8

repositories {
    mavenCentral()
}

dependencies {
    compile group: 'junit', name: 'junit', version: '4.12'
    // https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core
    compile group: 'org.apache.logging.log4j', name: 'log4j-core', version: '2.11.1'
    // https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api
    compile group: 'org.apache.logging.log4j', name: 'log4j-api', version: '2.11.1'
}
```

Maven Repository : https://mvnrepository.com/

Basic Test Framework

- Add log4j2.xml in /src/test/resource

myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/test/resources/log4j2.xml [myFirstAppiumProject_test]

PracticeAppium > src > test > resources > log4j2.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>

  <!-- Variable -->
  <Properties>
    <Property name="logfilePath">logs</Property>
    <Property name="patternLayout">%d{HH:mm:ss.SSS} [%level] %C{0}:%M:%L %msg %n</Property>
  </Properties>

  <!-- Configuration -->
  <Appenders>
    <!-- Output log on console -->
    <Console name="console" target="SYSTEM_OUT">
      <PatternLayout>
        <Pattern>${patternLayout}</Pattern>
      </PatternLayout>
    </Console>
    <!--Output log into file -->
    <RollingFile name="rollingFile" fileName="${logfilePath}/info.log" filePattern="${logfilePath}/info-%d{yyyy-MM-dd}-%i.log">
      <PatternLayout>
        <pattern>${patternLayout}</pattern>
      </PatternLayout>
      <Policies>
        <TimeBasedTriggeringPolicy interval="1" modulate="true" />
        <SizeBasedTriggeringPolicy size="10MB" />
      </Policies>
      <DefaultRolloverStrategy max="5"/>
    </RollingFile>
  </Appenders>

  <!-- Set up Loggers -->
  <Loggers>
    <Root level="all">
      <AppenderRef ref="console"/>
      <AppenderRef ref="rollingFile"/>
    </Root>
  </Loggers>

</Configuration>
```

Basic Test Framework

- Add Log into each class

myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/test/java/com/github/chenchiahung/testcases/case01.java [myFirstAppiumProject_test]

PracticeAppium > src > test > java > com > github > chenchiahung > testcases > case01

Project

- PracticeAppium [myFirstAppiumPro
  - .gradle
  - .idea
  - build
  - logs
  - out
  - src
    - main
      - java
      - resources
    - test
      - java
        - com.github.chenchiahung
          - case01
          - case02
          - case03
      - resources
        - log4j2.xml
  - .gitignore
  - build.gradle
  - local.properties
  - settings.gradle
- External Libraries
- Scratches and Consoles

case01.java

```java
package com.github.chenchiahung.testcases;

import ...

public class case01 {

    private static final Logger Log = LogManager.getLogger(case01.class);

    @BeforeClass
    public static void setUpClass() { Log.info("setUpClass"); }

    @AfterClass
    public static void tearDownClass() { Log.info("tearDownClass"); }

    @Before
    public void setUp() { Log.info("setUp"); }

    @Test
    public void Scenario01() { Log.info("Scenario01"); }

    @Test
    public void Scenario02() { Log.info("Scenario02"); }

    @After
    public void tearDown() { Log.info("tearDown"); }
}
```

Basic Test Framework

- JUNIT Annotation

myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/test/java/com/github/chenchiahung/testcases/case01.java [myFirstAppiumProject_test]

PracticeAppium > src > test > java > com > github > chenchiahung > testcases > case01

Project

- PracticeAppium [myFirstAppiumPro
  - .gradle
  - .idea
  - build
  - logs
  - out
  - src
    - main
      - java
      - resources
    - test
      - java
        - com.github.chenchiahung
          - case01
          - case02
          - case03
      - resources
        - log4j2.xml
  - .gitignore
  - build.gradle
  - local.properties
  - settings.gradle
- External Libraries
- Scratches and Consoles

case01.java

```
package com.github.chenchiahung.testcases;

import ...

public class case01 {

    private static final Logger Log = LogManager.getLogger(case01.class);

    @BeforeClass
    public static void setUpClass() { Log.info("setUpClass"); }

    @AfterClass
    public static void tearDownClass() { Log.info("tearDownClass"); }

    @Before
    public void setUp() { Log.info("setUp"); }

    @Test
    public void Scenario01() { Log.info("Scenario01"); }

    @Test
    public void Scenario02() { Log.info("Scenario02"); }

    @After
    public void tearDown() { Log.info("tearDown"); }
}
```

Basic Test Framework

- **Practice**

```
cd PracticeAppium

gradle clean

gradle clean test

gradle clean test -i

gradle clean test -i --tests="com.github.chenchiahung.testcases.case01"

gradle clean test -i --tests="*case01"

gradle clean test -i --tests="*case*.Scenario02"

gradle clean test -i --tests="*case01" --tests="*case03"
```

Basic Test Framework

- Test Report and Test Result

myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/test/java/com/github/chenchiahung/testcases/case01.java [myFirstAppiumProject_test]

PracticeAppium > src > test > java > com > github > chenchiahung > testcases > case01

Project

- PracticeAppium [myFirstAppiumProject] ~/MyRepository/PracticeAp
  - .gradle
  - .idea
  - build
    - classes
    - reports
      - tests
        - test
          - classes
          - css
          - js
          - packages
          - index.html
    - resources
    - test-results
      - test
        - binary
        - TEST-com.github.chenchiahung.testcases.case01.xml
        - TEST-com.github.chenchiahung.testcases.case02.xml
        - TEST-com.github.chenchiahung.testcases.case03.xml
    - tmp
  - logs
  - out
  - src

case01.java

```
1   package com.github.chenchiahung.testcases;
2
3   import ...
6
7   public class case01 {
8
9     private static final Logger Log = LogManager.getLogger(case01.class);
10
11    @BeforeClass
12    public static void setUpClass() { Log.info("setUpClass"); }
15
16    @AfterClass
17    public static void tearDownClass() { Log.info("tearDownClass"); }
20
21    @Before
22    public void setUp() { Log.info("setUp"); }
25
26    @Test
27    public void Scenario01() { Log.info("Scenario01"); }
30
31    @Test
32    public void Scenario02() { Log.info("Scenario02"); }
35
36    @After
37    public void tearDown() { Log.info("tearDown"); }
40  }
41
```

EXAMPLE 2
First Test Automation Script

Example_02

- **git checkout Example_02**

- Use Gradle, Junit4, Log4j2 and Appium java-client on test automation framework

  - Appium java-client
    https://github.com/appium/java-client

- Launch Appium CLI by command
  - appium --address 127.0.0.1 --port 9999

@Before

https://appium.io/docs/en/writing-running-appium/running-tests/

myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/test/java/com/github/chenchiahung/testcases/MyFirstAppiumTest.java [myFirstAppiumProject_test]

PracticeAppium › src › test › java › com › github › chenchiahung › testcases › MyFirstAppiumTest

```
package com.github.chenchiahung.testcases;

import ...

public class MyFirstAppiumTest {

  private static final Logger Log = LogManager.getLogger(MyFirstAppiumTest.class);

  WebDriver driver;

  @Before
  public void setUp() throws MalformedURLException {
    Log.info("setUp - Start");

    URL url = new URL( spec: "http://127.0.0.1:9999/wd/hub");
    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities.setCapability(MobileCapabilityType.PLATFORM_NAME,  value: "Android");
    capabilities.setCapability(MobileCapabilityType.PLATFORM_VERSION,  value: "9.0.0");
    capabilities.setCapability(MobileCapabilityType.AUTOMATION_NAME,  value: "UiAutomator2");
    capabilities.setCapability(MobileCapabilityType.DEVICE_NAME,  value: "emulator-5554");
    capabilities.setCapability(MobileCapabilityType.UDID,  value: "emulator-5554");
    capabilities.setCapability(MobileCapabilityType.FULL_RESET,  value: true);
    capabilities.setCapability(MobileCapabilityType.APP,  value: "/Users/chiahungchen/Documents/AppPackage/2.92.217.196-1384.apk");
    capabilities.setCapability(MobileCapabilityType.NEW_COMMAND_TIMEOUT, 60000);
    capabilities.setCapability(AndroidMobileCapabilityType.APP_ACTIVITY,  value: "com.thecarousell.Carousell.activities.EntryActivity")
    capabilities.setCapability(AndroidMobileCapabilityType.APP_PACKAGE,  value: "com.thecarousell.Carousell");
    capabilities.setCapability(AndroidMobileCapabilityType.AVD,  value: "Pixel2_API26");
    driver = new AndroidDriver(url, capabilities);

    Log.info("setUp - End");
  }
```

Webdriver

Desired Capabilities

Locator, waitElement, findElement and clickElement

https://www.guru99.com/implicit-explicit-waits-selenium.html
https://appium.io/docs/en/commands/element/find-element/
https://appium.io/docs/en/commands/element/actions/click/

```
Log.info("Step 01 - Click Login Button");
By loginButton_1 = By.id("com.thecarousell.Carousell:id/welcome_page_login_button");
WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
wait.until(ExpectedConditions.visibilityOfElementLocated(loginButton_1));
WebElement element = driver.findElement(loginButton_1);
element.click();

Log.info("Step 02 - Click Cancel");
By cancelLink =
    By.xpath(
        "//android.widget.LinearLayout[@content-desc=\"Choose an Account\"]/android.widget.LinearLayout/android.widget.Button");
wait.until(ExpectedConditions.visibilityOfElementLocated(cancelLink));
element = driver.findElement(cancelLink);
element.click();
```

Locator
waitElement
findElement
clickElement

FindElements and SendKeys

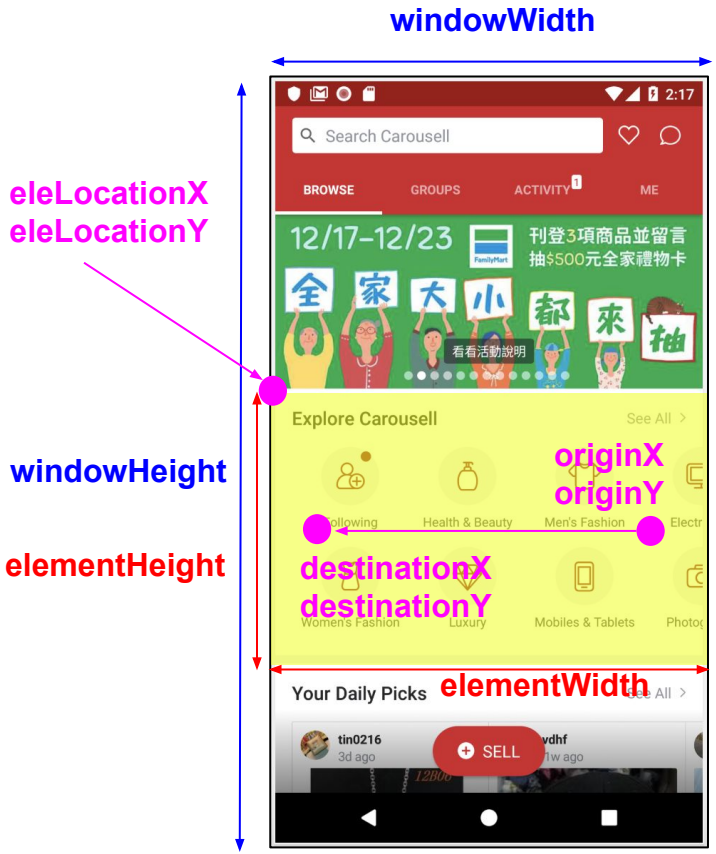
https://appium.io/docs/en/commands/element/find-elements/#find-elements
https://appium.io/docs/en/commands/element/actions/send-keys/

```
Log.info("Step 03 - Input Account");
By editTextbox = MobileBy.AndroidUIAutomator( uiautomatorText: ".className(\"android.widget.EditText\")");
wait.until(ExpectedConditions.visibilityOfElementLocated(editTextbox));
element = driver.findElements(editTextbox).get(0);
element.sendKeys( ...keysToSend: "chiahung003");

Log.info("Step 04 - Input Password");
element = driver.findElements(editTextbox).get(1);
element.sendKeys( ...keysToSend: "1111111w");

Log.info("Step 05 - Click Log in Button");
By loginButton_2 = By.id("com.thecarousell.Carousell:id/login_page_login_button");
wait.until(ExpectedConditions.visibilityOfElementLocated(loginButton_2));
element = driver.findElement(loginButton_2);
element.click();
```

FindElements

SendKeys

TouchAction https://appium.io/docs/en/writing-running-appium/touch-actions/#touchaction




```
Log.info(
    "Step 06 - Define the swipe action on the view bar of explore carousell"); // deprecated
By CategoryList = By.id("com.thecarousell.Carousell:id/recyclerView");
wait.until(ExpectedConditions.visibilityOfElementLocated(CategoryList));
element = driver.findElement(CategoryList);

int windowWidth = driver.manage().window().getSize().getWidth();
int windowHeight = driver.manage().window().getSize().getHeight();
Log.debug( message: "window width : {} , window height {}", windowWidth, windowHeight);

int eleLocationX = element.getLocation().getX();
int eleLocationY = element.getLocation().getY();
Log.debug( message: "element location x : {} , element location y : {}", eleLocationX, eleLocationY);

int destinationX = eleLocationX + (int) (element.getSize().getWidth() * 0.1F);
int destinationY = eleLocationY + (int) (element.getSize().getHeight() * 0.5F);
int originX = eleLocationX + (int) (element.getSize().getWidth() * 0.9F);
int originY = eleLocationY + (int) (element.getSize().getHeight() * 0.5F);
Log.debug( message: "Origin X : {} , Origin Y : {}", originX, originY);
Log.debug( message: "Destination X : {}, Destination Y : {}", destinationX, destinationY);

Log.info("Step 07 - Try to find Cars category on the view bar of explore carousell");
By CarsCategory =
    MobileBy.AndroidUIAutomator( uiautomatorText: ".className(\"android.widget.TextView\").text(\"Cars\")");
int count = 1;
while (count <= 8) {
  try {
    WebElement ele = driver.findElement(CarsCategory);
    if (ele != null) {
      Log.debug( message: "Count : {}, find element !", count);
      ele.click();
      break;
    }
  } catch (Exception e) {
    Log.warn( message: "Count : {} , Can not find element !", count);
    TouchAction action = new TouchAction((AndroidDriver) driver);
    action.press(PointOption.point(originX, originY));
    action.waitAction(WaitOptions.waitOptions(Duration.ofSeconds(2L)));
    action.moveTo(PointOption.point(destinationX, destinationY));
    action.release();
    action.perform();
    count = count + 1;
  }
}
```

TouchAction

Press keycode

https://appium.io/docs/en/commands/device/keys/press-keycode/

```
Log.info("Step 10 - Search 'BMW' and press Enter");
By inputSearchTextbox = By.id("com.thecarousell.Carousell:id/input_search_bar");
wait.until(ExpectedConditions.visibilityOfElementLocated(inputSearchTextbox));
element = driver.findElement(inputSearchTextbox);
element.sendKeys( ...keysToSend: "BMW");
KeyEvent keyEvent = new KeyEvent(AndroidKey.ENTER);
((AndroidDriver) driver).pressKey(keyEvent);
```

PressKey

Get Attribute

https://appium.io/docs/en/commands/element/attributes/attribute/#get-element-attribute

```
Log.info("Step 14 - Check if title contain 'BMW'");
By productDetailTitle = By.id("com.thecarousell.Carousell:id/tvInfo");
wait.until(ExpectedConditions.visibilityOfElementLocated(productDetailTitle));
element = driver.findElement(productDetailTitle);
String title = element.getAttribute( name: "text").toLowerCase();
Assert.assertTrue(title.contains("bmw"));
```

GetAttribute

First Test Automation Script

- Practice

EXAMPLE 3
Page Object

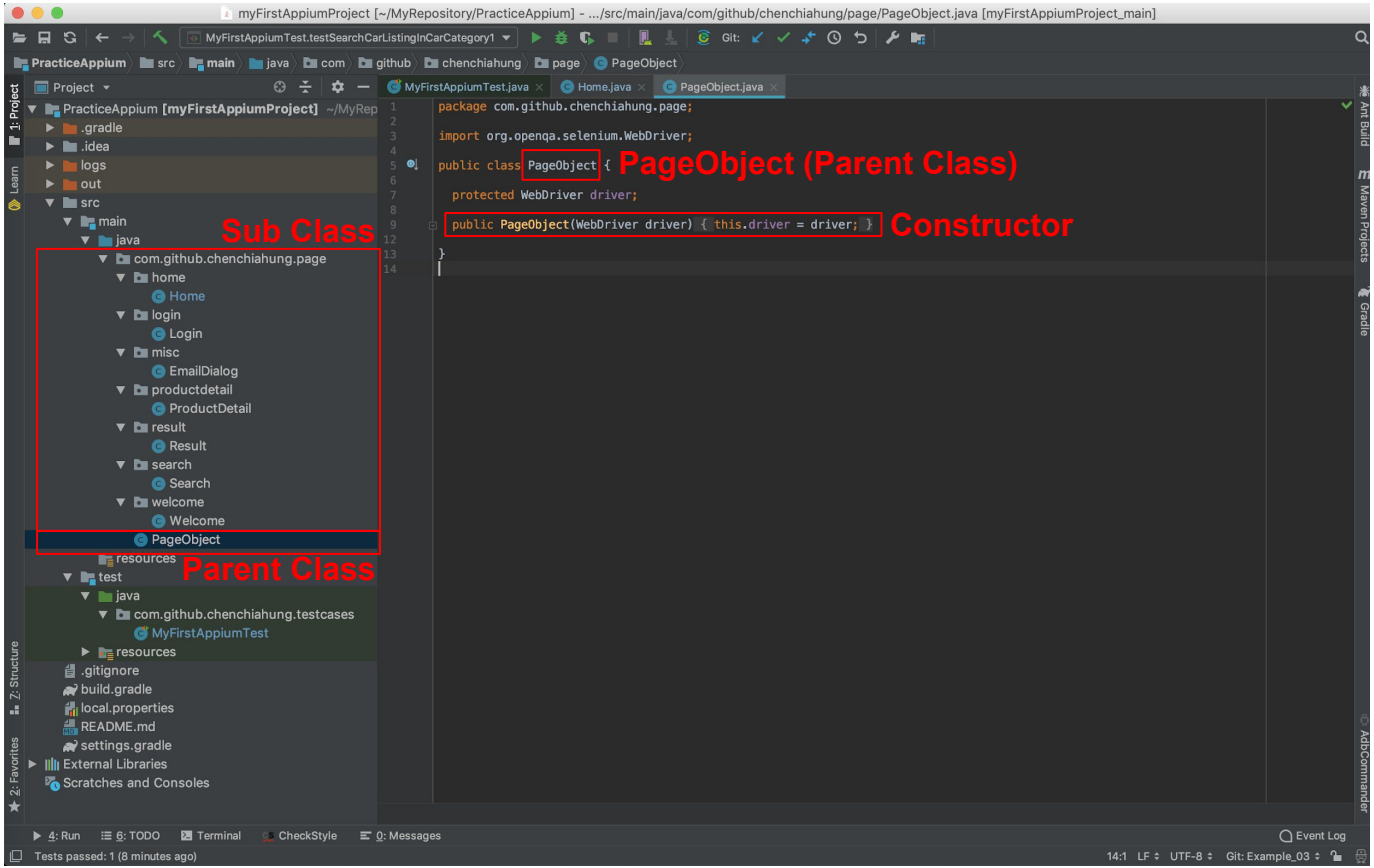
Page Object

- https://martinfowler.com/bliki/PageObject.html
- https://github.com/SeleniumHQ/selenium/wiki/PageObjects
- https://stackoverflow.com/questions/47140697/should-we-write-separate-page-object-for-pop-up-with-a-selection-drop-down

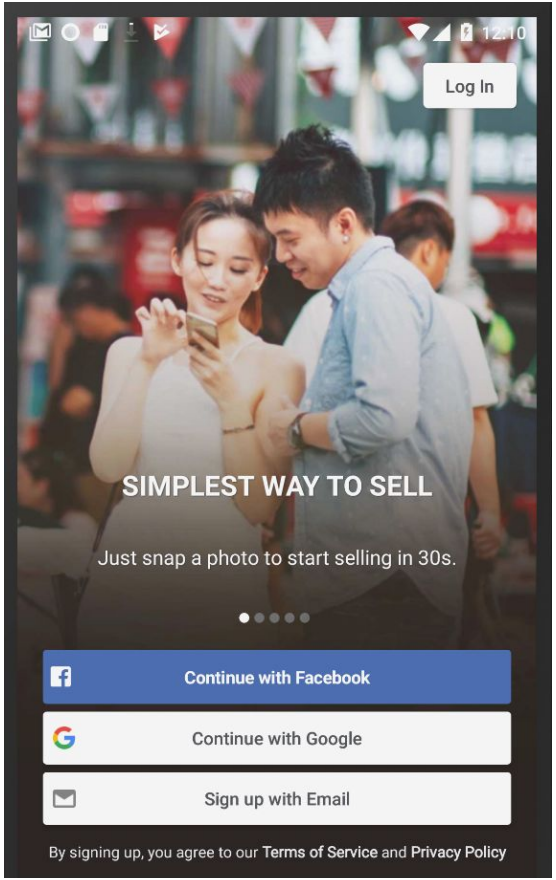
Summary

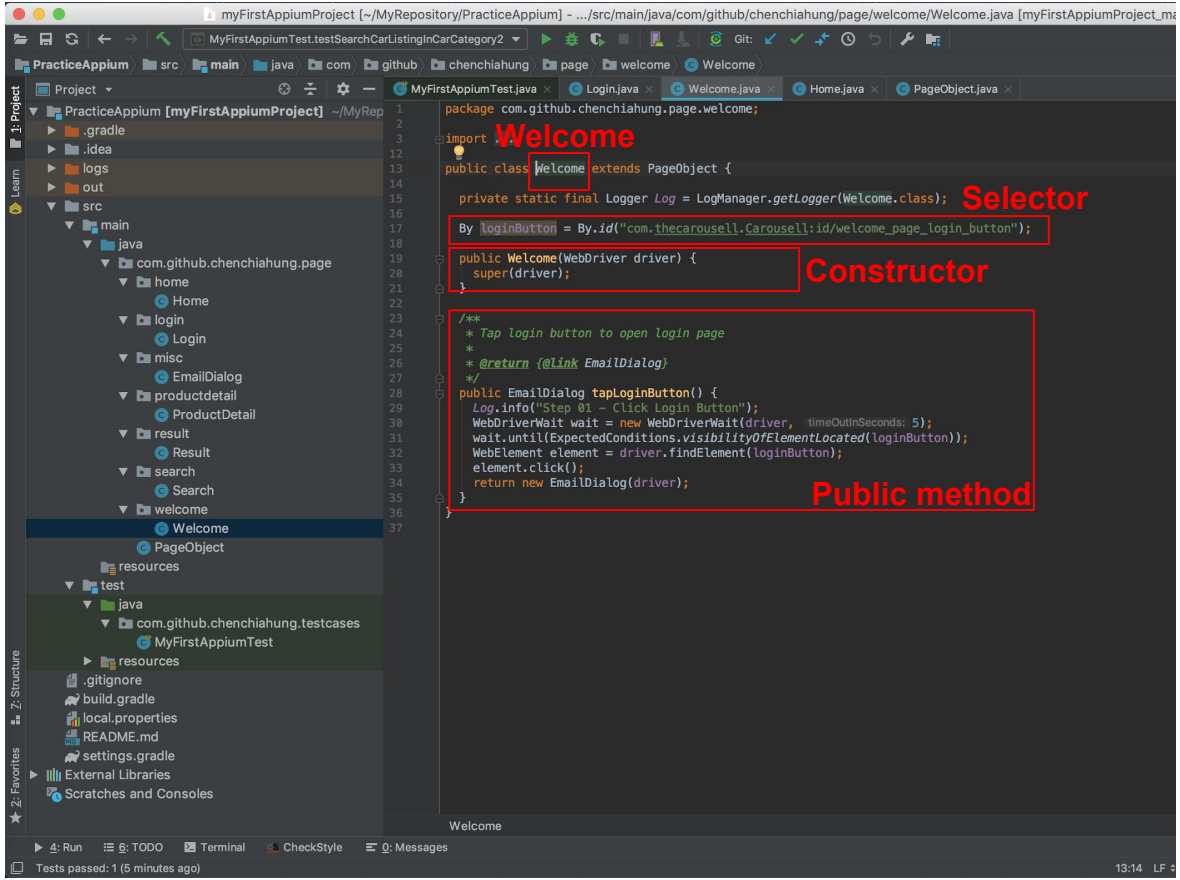
- The public methods represent the services that the page offers
- Try not to expose the internals of the page
- Generally don't make assertions
- Methods return other PageObjects
- Need not represent an entire page
- Different results for the same action are modelled as different methods

Page Object

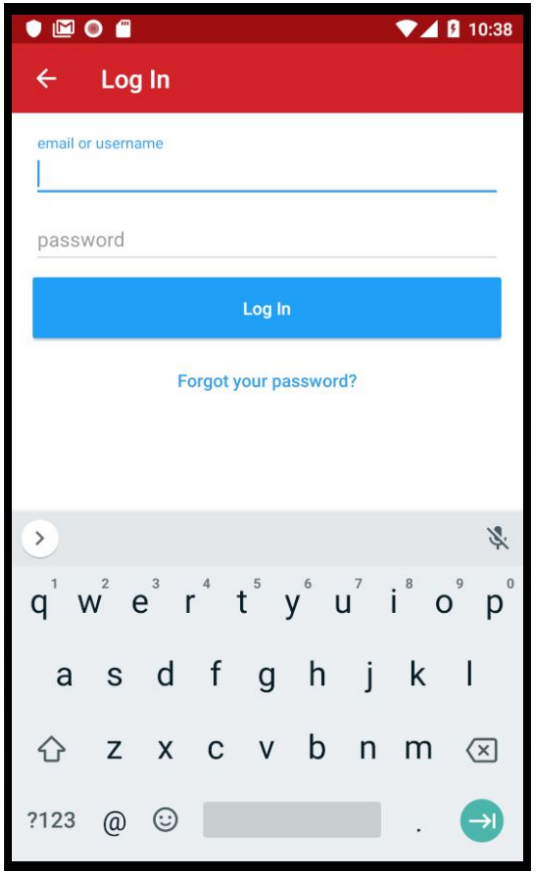

myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/main/java/com/github/chenchiahung/page/PageObject.java [myFirstAppiumProject_main]
PracticeAppium
src
main
java
com
github
chenchiahung
page
PageObject
MyFirstAppiumTest.java
Home.java
PageObject.java
package com.github.chenchiahung.page;
import org.openqa.selenium.WebDriver;
public class PageObject {
protected WebDriver driver;
public PageObject(WebDriver driver) { this.driver = driver; }
}
PageObject (Parent Class)
Constructor
Sub Class
Parent Class
PracticeAppium [myFirstAppiumProject]
.gradle
.idea
logs
out
src
main
java
com.github.chenchiahung.page
home
Home
login
Login
misc
EmailDialog
productdetail
ProductDetail
result
Result
search
Search
welcome
Welcome
PageObject
resources
test
java
com.github.chenchiahung.testcases
MyFirstAppiumTest
resources
.gitignore
build.gradle
local.properties
README.md
settings.gradle
External Libraries
Scratches and Consoles
Tests passed: 1 (8 minutes ago)

Page Object


12:10
Log In
SIMPLEST WAY TO SELL
Just snap a photo to start selling in 30s.
Continue with Facebook
Continue with Google
Sign up with Email
By signing up, you agree to our Terms of Service and Privacy Policy


myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/main/java/com/github/chenchiahung/page/welcome/Welcome.java [myFirstAppiumProject_ma
MyFirstAppiumTest.testSearchCarListingInCarCategory2
Git:
PracticeAppium
src
main
java
com
github
chenchiahung
page
welcome
Welcome
Project
MyFirstAppiumTest.java
Login.java
Welcome.java
Home.java
PageObject.java
PracticeAppium [myFirstAppiumProject] ~/MyRep
.gradle
.idea
logs
out
src
main
java
com.github.chenchiahung.page
home
Home
login
Login
misc
EmailDialog
productdetail
ProductDetail
result
Result
search
Search
welcome
Welcome
PageObject
resources
test
java
com.github.chenchiahung.testcases
MyFirstAppiumTest
resources
.gitignore
build.gradle
local.properties
README.md
settings.gradle
External Libraries
Scratches and Consoles
package com.github.chenchiahung.page.welcome;
import
Welcome
public class Welcome extends PageObject {
private static final Logger Log = LogManager.getLogger(Welcome.class);
Selector
By loginButton = By.id("com.thecarousell.Carousell:id/welcome_page_login_button");
public Welcome(WebDriver driver) {
super(driver);
}
Constructor
/**
* Tap login button to open login page
*
* @return {@link EmailDialog}
*/
public EmailDialog tapLoginButton() {
Log.info("Step 01 - Click Login Button");
WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 5);
wait.until(ExpectedConditions.visibilityOfElementLocated(loginButton));
WebElement element = driver.findElement(loginButton);
element.click();
return new EmailDialog(driver);
}
}
Public method
Welcome
4: Run
6: TODO
Terminal
CheckStyle
0: Messages
Tests passed: 1 (5 minutes ago)
13:14
LF

Page Object



myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/main/java/com/github/chenchiahung/page/login/Login.java [myFirstAppiumProject_main]

```
 *
 * @param account
 */
private Login enterAccount(String account) {
  WebDriverWait wait = new WebDriverWait(driver,  timeOutInSeconds: 5);
  wait.until(ExpectedConditions.visibilityOfElementLocated(editTextbox));
  WebElement element = driver.findElements(editTextbox).get(0);
  element.sendKeys(account);
  return this;
}

/**...*/
private Login enterPassword(String password) {
  WebDriverWait wait = new WebDriverWait(driver,  timeOutInSeconds: 5);
  wait.until(ExpectedConditions.visibilityOfElementLocated(editTextbox));
  WebElement element = driver.findElements(editTextbox).get(1);
  element.sendKeys(password);
  return this;
}

/**...*/
public Home tapLoginButtonExpectingSuccess() {
  Log.info("Step 05 - Click Log in Button");
  WebDriverWait wait = new WebDriverWait(driver,  timeOutInSeconds: 5);
  wait.until(ExpectedConditions.visibilityOfElementLocated(loginButton));
  WebElement element = driver.findElement(loginButton);
  element.click();
  return new Home(driver);
}

/**...*/
public Login tapLoginButtonExpectingFailure() {
  WebDriverWait wait = new WebDriverWait(driver,  timeOutInSeconds: 5);
  wait.until(ExpectedConditions.visibilityOfElementLocated(loginButton));
  WebElement element = driver.findElement(loginButton);
  element.click();
  return new Login(driver);
}

/**...*/
public Home loginAsExpectingSuccess(String account, String password) {
  Log.info("Step 03 - Input Account");
  this.enterAccount(account);
  Log.info("Step 04 - Input Password");
  this.enterPassword(password);
  return this.tapLoginButtonExpectingSuccess();
}
```

Private method

Private method

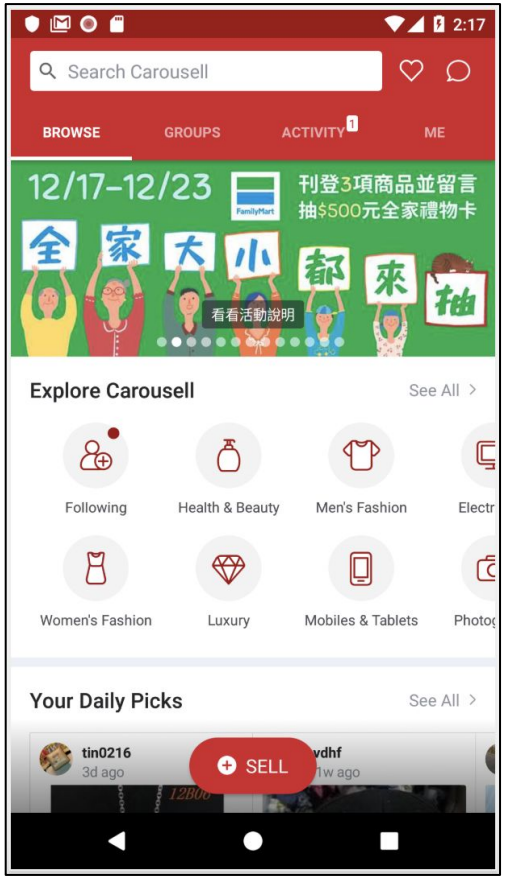
Public method

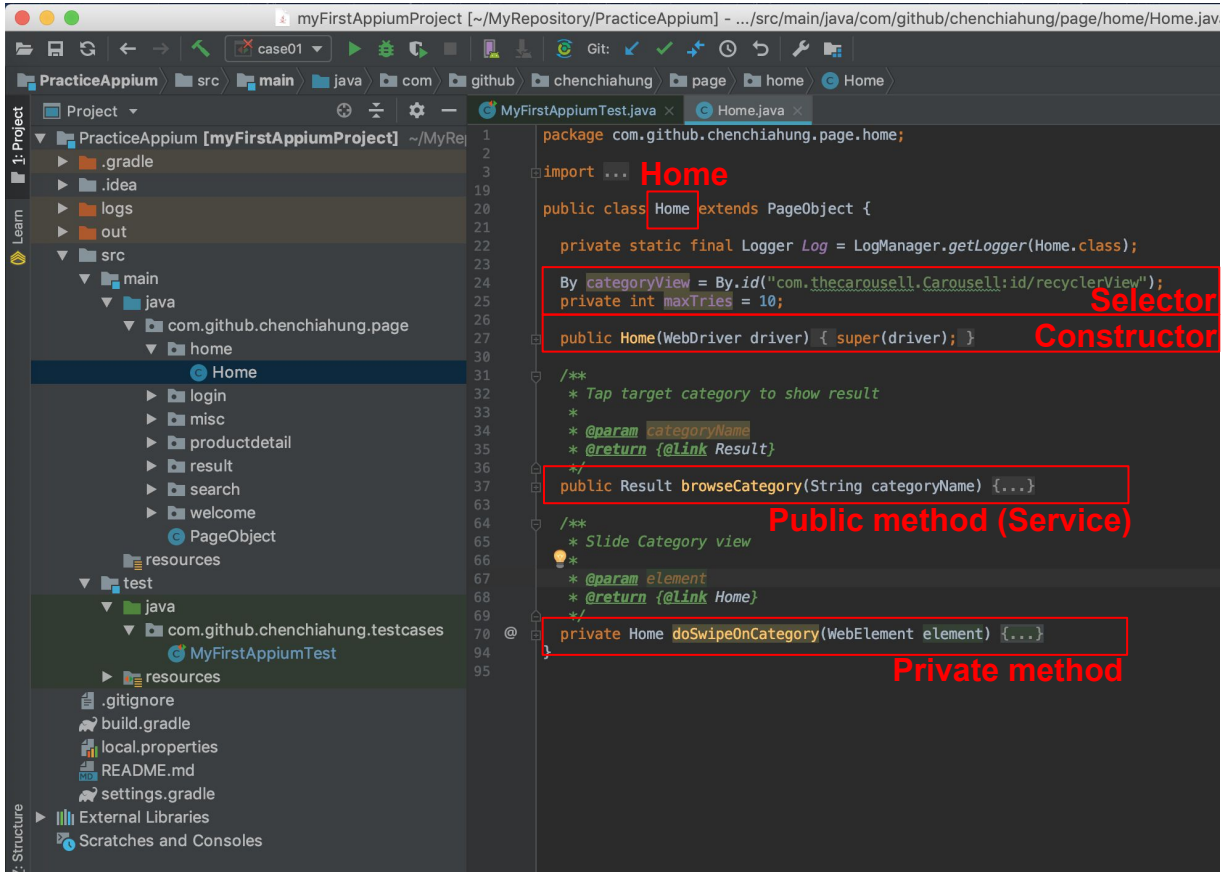
Public method

Public method

Tests passed: 1 (9 minutes ago)

Page Object


2:17
Search Carousell
BROWSE
GROUPS
ACTIVITY
ME
12/17-12/23
刊登3項商品並留言
抽$500元全家禮物卡
全 家 大 小 都 來 抽
看看活動說明
Explore Carousell
See All
Following
Health & Beauty
Men's Fashion
Women's Fashion
Luxury
Mobiles & Tablets
Your Daily Picks
See All
tin0216
3d ago
SELL

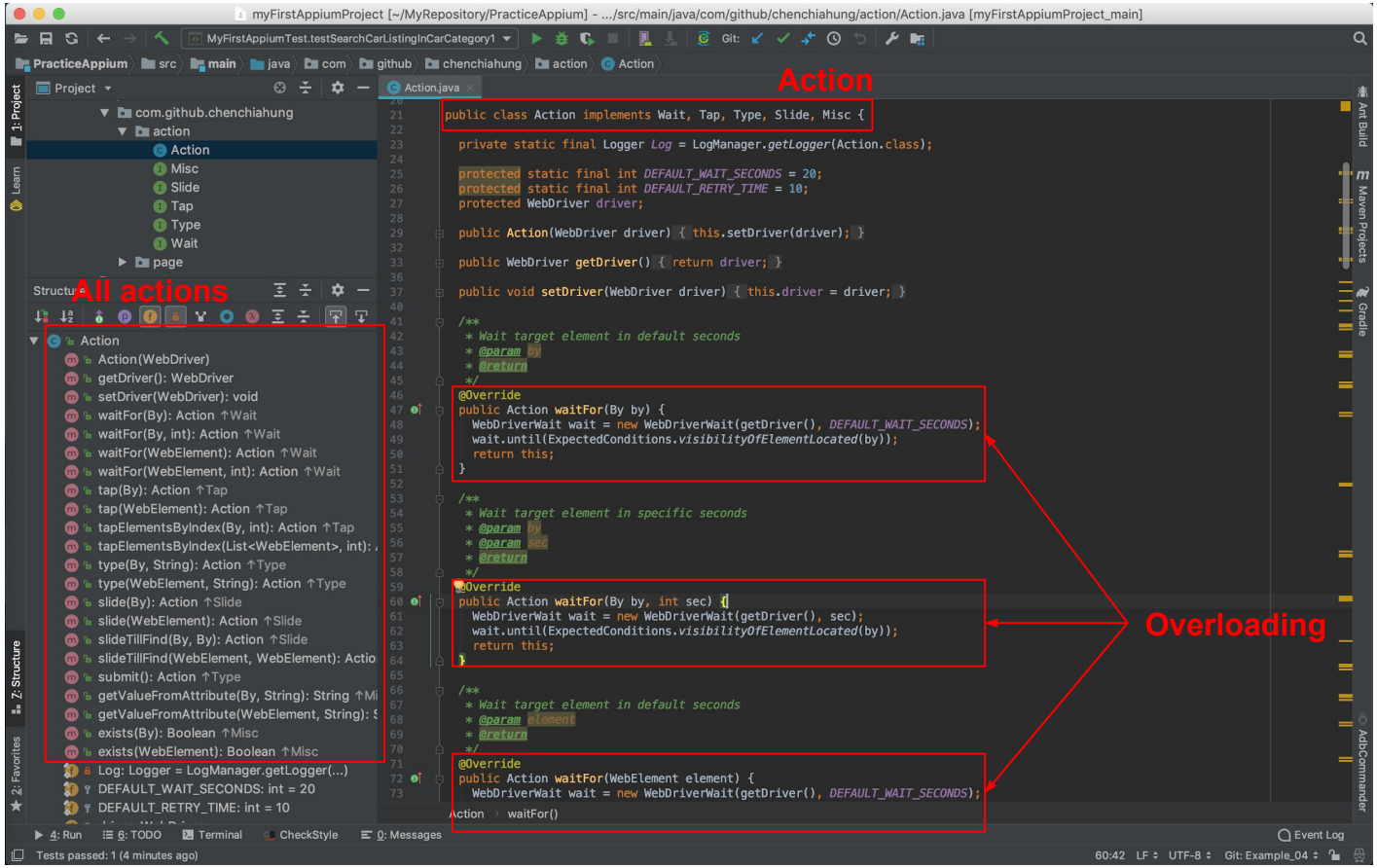

myFirstAppiumProject [~/MyRepository/PracticeAppium] - .../src/main/java/com/github/chenchiahung/page/home/Home.java
case01
Git:
PracticeAppium
src
main
java
com
github
chenchiahung
page
home
Home
Project
MyFirstAppiumTest.java
Home.java
PracticeAppium [myFirstAppiumProject]
.gradle
.idea
logs
out
src
main
java
com.github.chenchiahung.page
home
Home
login
misc
productdetail
result
search
welcome
PageObject
resources
test
java
com.github.chenchiahung.testcases
MyFirstAppiumTest
resources
.gitignore
build.gradle
local.properties
README.md
settings.gradle
External Libraries
Scratches and Consoles
package com.github.chenchiahung.page.home;
import ...
Home
public class Home extends PageObject {
private static final Logger Log = LogManager.getLogger(Home.class);
By categoryView = By.id("com.thecarousell.Carousell:id/recyclerView");
private int maxTries = 10;
Selector
public Home(WebDriver driver) { super(driver); }
Constructor
/**
* Tap target category to show result
*
* @param categoryName
* @return {@link Result}
*/
public Result browseCategory(String categoryName) {...}
Public method (Service)
/**
* Slide Category view
*
* @param element
* @return {@link Home}
*/
private Home doSwipeOnCategory(WebElement element) {...}
Private method
}

Page Object

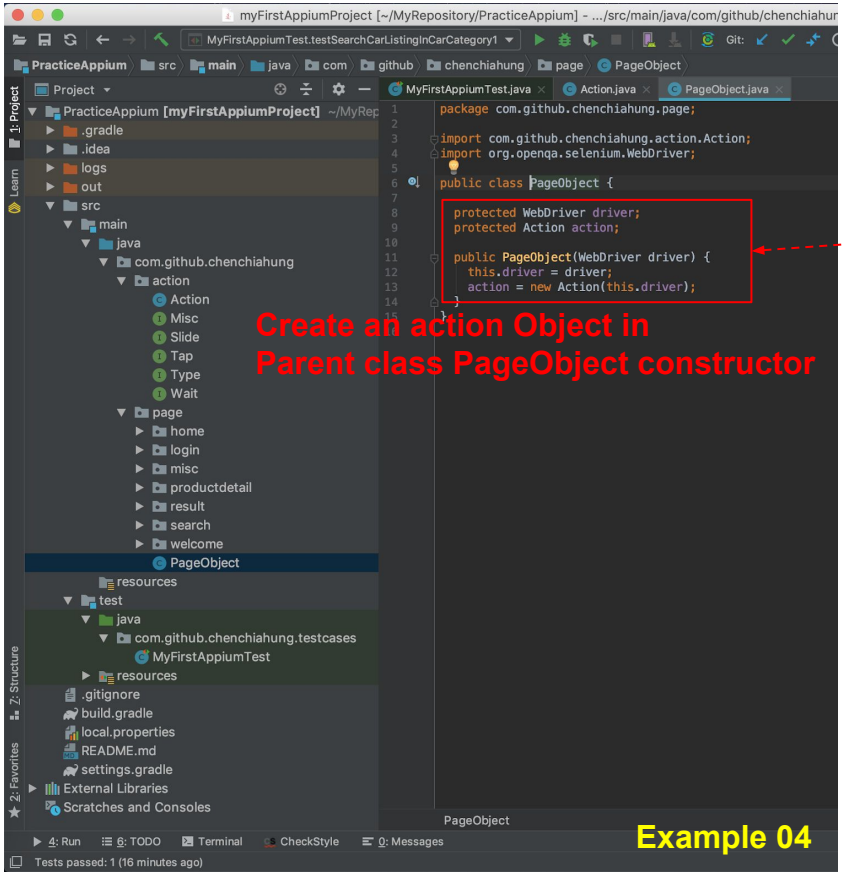
- **Practice**

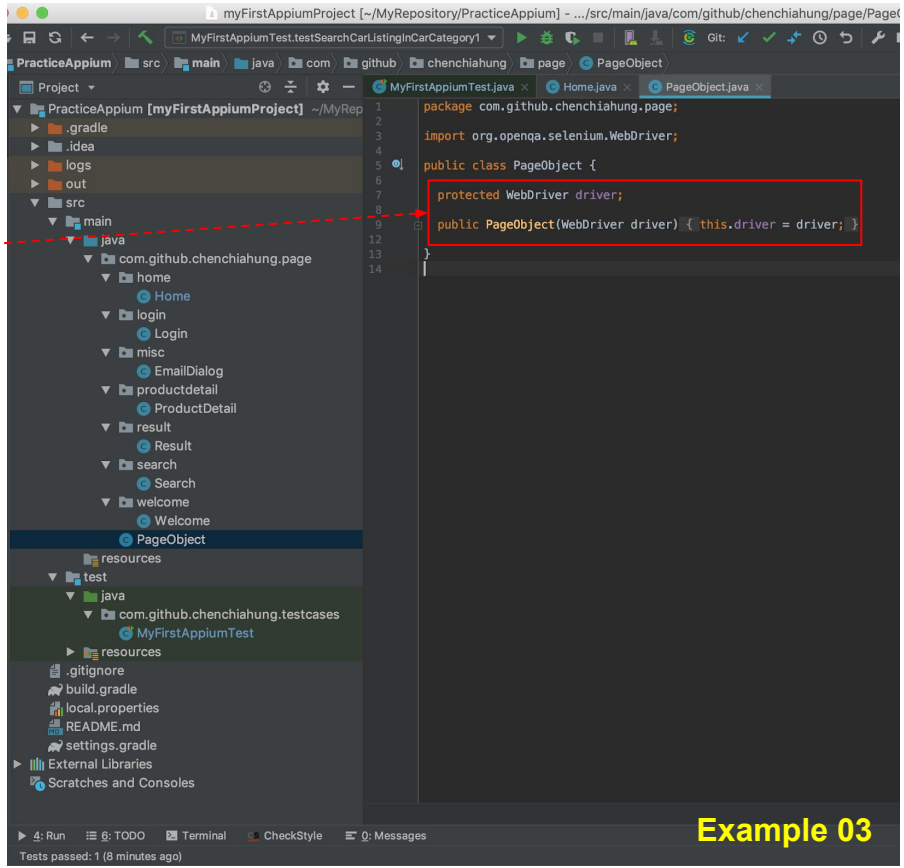
EXAMPLE 4
Reduce Duplicated Code

Reduce Duplicated Code

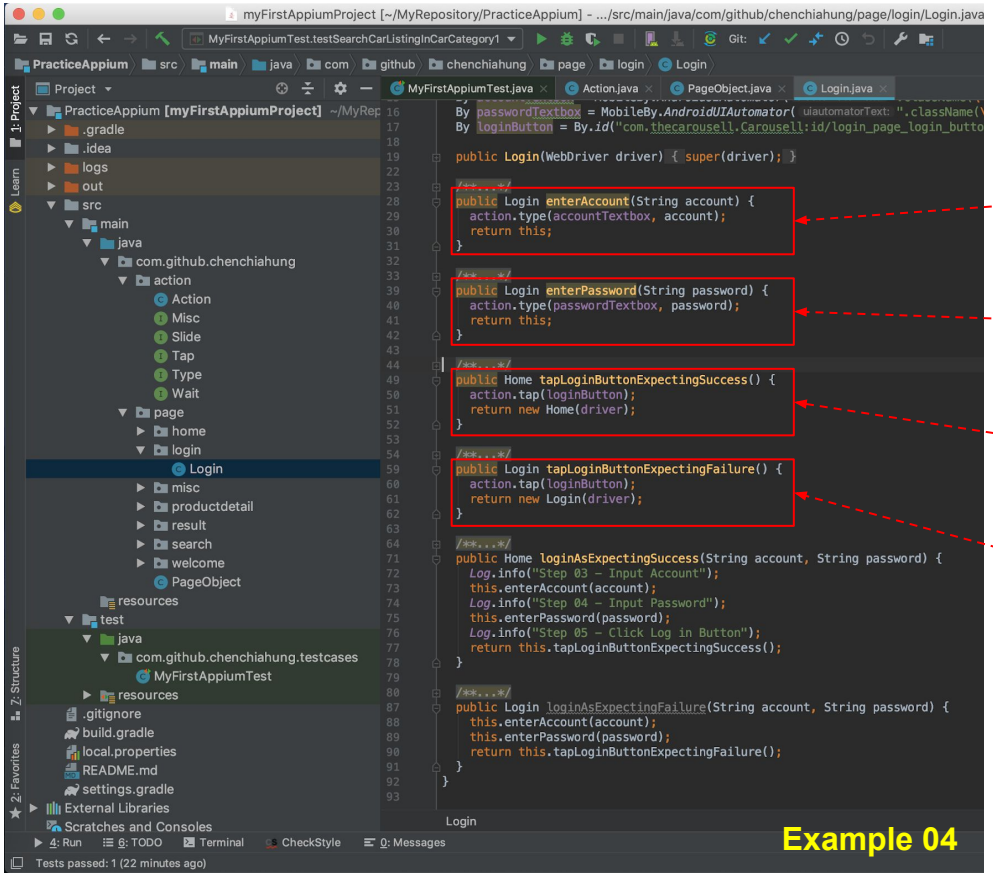


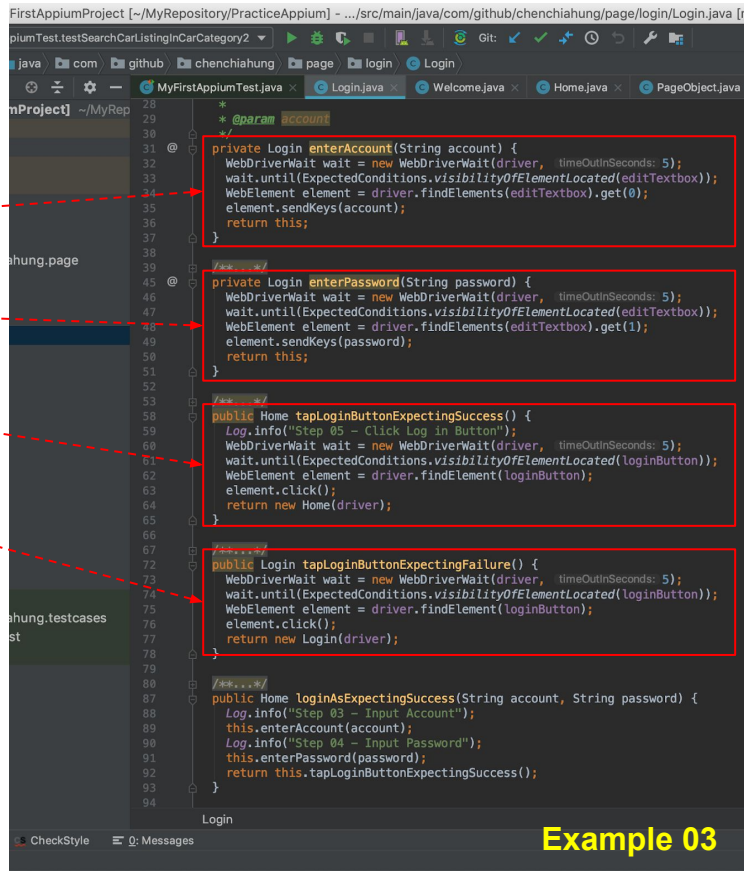
Reduce Duplicated Code





Reduce Duplicated Code



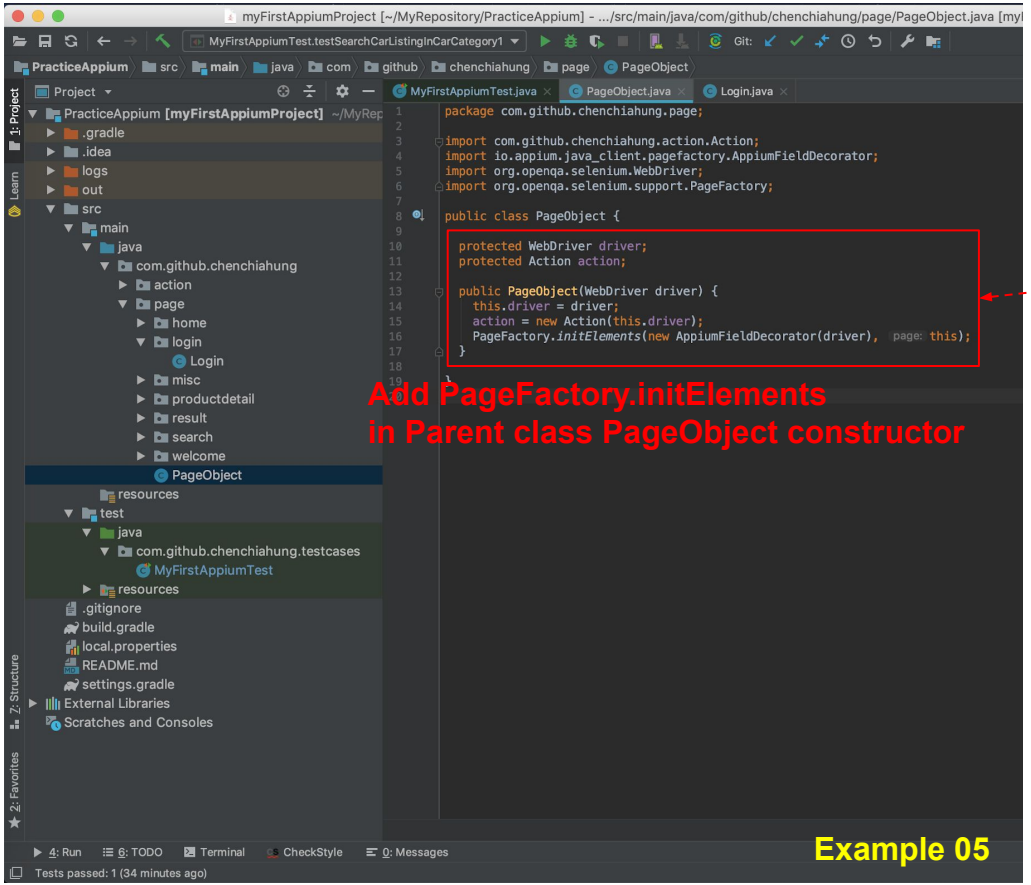


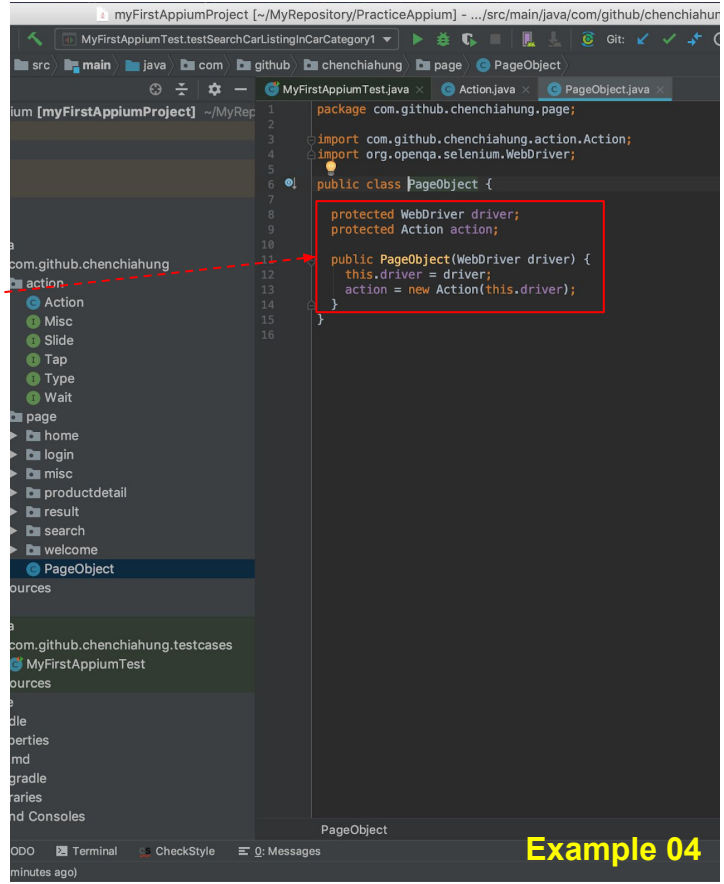
Reduce Duplicated Code

- **Practice**

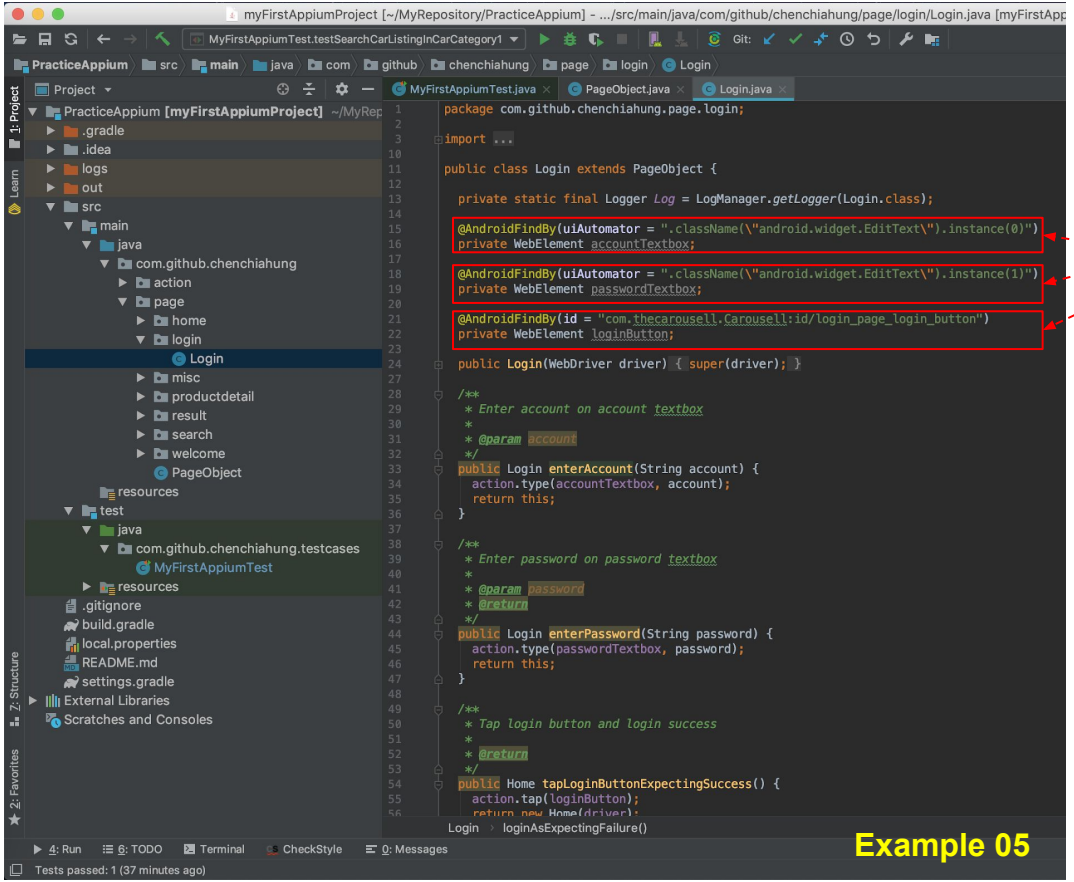
EXAMPLE 5
Page Factory

Page Factory

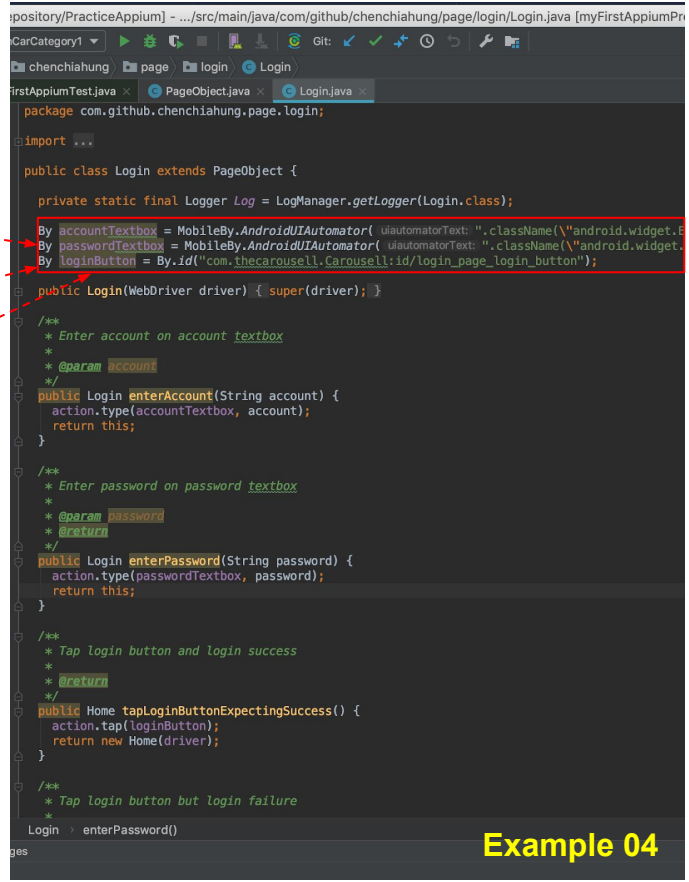




Page Factory



Example 05



Example 04

Page Factory

- **Practice**
  - **Page Factory annotation**
    https://appium.github.io/java-client/io/appium/java_client/pagefactory/package-frame.html

```
public class Login extends PageObject {

    String uiAutomator() default ""
    String accessibility() default ""
    String id() default ""
    String className() default ""
    String tagName() default ""
    String xpath() default ""
    int priority() default 0

    @AndroidFindBy(id = "com.thecarousell.Carousell:id/login_page_login_button")
    @AndroidFindBy()
    private WebElement loginButton;

    public Login(WebDriver driver) {
        super(driver);
    }
```

Command + P

APPIUM
Pros and Cons

Appium Pros and Cons

Pros	Cons
• Cross platform (iOS, Android ...) • Commonly used • Cloud solution https://aws.amazon.com/tw/device-farm/ https://www.browserstack.com/ https://saucelabs.com/ https://bitbar.com/	• Long execution time (Install / execute / debug) • Selector / Locator • Cloud limitation and Issues • Complex framework

Final Scenario

Thanks for attending this event !

https://goo.gl/forms/bGtwLQitfFDLNxs63