nbviewer

temp / Appium_Tips.ipynb

Qxf2 examples

10 lectures on Mobile Testing

1. [Chess on Emulator](#)
2. [ATP on Phone](#)
3. [Identify Mobile Elements](#)
4. [Mobile Gestures](#)
5. [Test on Multiple Mobile Devices - Big Basket](#)
6. [Scroll Result Table](#)
7. [Virtual Keyboard](#)
8. [What is NAF](#)
9. [AppWait](#)
10. [Android Unit Testing: Android Testing Framework, Robolectric](#)

Mobile Automation

- [Chess on SauceLabs](#)
- [Parametrize](#)
- [Selendroid Python](#)

Automation Framework

[Qxf2 GUI Automation Framework](#)

Selenium

- [Getting Started with XPath Selenium](#)
- [XPath Tutorial](#)
- [Firebug](#)
- [Selenium on SauceLabs - part1](#)
- [Selenium on SauceLabs - part2](#)
- [Selenium on SauceLabs - part3](#)

[An example Appium + Python script with different locator strategies](#)

[MondoDB and Python](#)

[OptParse](#)

POM

[POM model](#)

[Batman & POM](#)

[Practice Apps](#)

[Jenkins](#)

[Jenkins & Python](#)

[Python Unit Test](#)

[All Unit Testing Articles](#)

How Appium Works

- http://toolsqa.com/mobile-automation/appium/appium-a-cross-platform-mobile-automation-tool/

Appium with Eclipse on Windows

Appium Tutorial: Set up Appium with Eclipse on Windows

- http://toolsqa.com/mobile-automation/appium/appium-tutorial/

Setup UI Automator with Eclipse

- https://looksok.wordpress.com/2014/02/08/uiautomator-in-eclipse/

Locate Element in UI Automator Viewer

- http://toolsqa.com/mobile-automation/appium/appium-a-cross-platform-mobile-automation-tool/

- http://toolsqa.com/mobile-automation/appium/locate-element-using-uiautomatorviewer/

Running UI Automator

- https://stackoverflow.com/questions/25201743/error-in-using-uiautomatorviewer-for-testing-android-app-in-appium/28526871#28526871/8

- https://github.com/appium/appium-uiautomator2-server

Appium Inspector

- https://medium.com/@eliasnogueira/inspect-an-app-with-the-new-appium-desktop-8ce4dc9aa95c

- While the inspector works well for iOS, there are some problem areas with it in Android on Appium at the moment. To that end, the Appium team encourages the use of uiautomatorviewer (which is an inspector tool provided by Google that provides similar functionality to the Appium inspector tool). For more info on how to set that up, read this.

  https://saucelabs.com/resources/articles/appium-bootcamp-chapter-1

Desired Capabilities

Device Name

- Connect your android device with PC and get device name by running adb devices command in command prompt.

Software App Package Name

- Find your app package name from setting ->app details section.

Activity Name

- Find your app package name from setting ->app details section .

Find Android OS Version

- In your android device, Open settings -> About phone -> Android version.

Reference

```
desired_caps = {}
desired_caps['platformName'] = 'Android'
desired_caps['platformVersion'] = '5.1'
desired_caps['deviceName'] = 'Android GoogleAPI Emulator'
desired_caps['appiumVersion'] = '1.6.5'
desired_caps['deviceOrientation'] = 'portrait'
desired_caps['browserName'] = ''
desired_caps['app'] = 'C:\\Users\\bilsto1\\python\\apps\\uk.co.aifact
ory.chessfree.apk'
desired_caps['appPackage'] = 'uk.co.aifactory.chessfree'
desired_caps['appActivity'] = 'uk.co.aifactory.chessfree.ChessFreeAct
ivity'
```

- [package-name/activity-name: Android defines activities for specific actions supported by the application. The activity name can be supplied relative to the application package-name by prefixing a '.' to the activity name, or full activity name with its own package-name](#)

Path to Application

```
desired_caps['app'] = os.path.abspath('../../../apps/TestApp5.apk')
```

Connect to Appium Server

```
driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
```

Special capabilities

[AUTOMATION_NAME uiautomator2](#)

[adb shell getprop - to Get Device Properties](#)

[newCommandTimeout desired capability 600](#) - Appium Log shows *[Appium] Closing session, cause was 'New Command Timeout of 60 seconds expired. Try customizing the timeout using the 'newCommandTimeout' desired capability'*

[newCommandTimeout capability integer](#)

```
MobileCapabilityType.NEW_COMMAND_TIMEOUT, "3600"
MobileCapabilityType.LAUNCH_TIMEOUT, "1200000"
MobileCapabilityType.TAKES_SCREENSHOT, "true"
```

Appium Log shows *[BaseDriver] The following capabilities were provided, but are not recognized by appium: appiumVersion, deviceOrientation.*

Appium locator by class name

Appium locator by xpath

- //android.widget.ImageView[contains(@resource-id,'action_bar') and contains(@content-desc,'Navigation panel')]

https://discuss.appium.io/t/issue-using-xpath-with-appium-for-native-android-apps/5664/8

Appium XPath Macaca Inspector

http://toolsqa.com/mobile-automation/appium/extract-xpath-from-app-using-macaca-inspector/

Appium Python

Appium Library Installation for Python

```
pip install Appium-Python-Client
from appium import webdriver
```

Appium - is application installed

- driver.is_app_installed(package_name)

```
driver.is_app_installed('com.google.android.youtube')
```

package-name: a unique identification string for the application, used by the OS to identify the application. For example, the package name for the maps application developed by Google for Android may be "com.google.android.apps.maps"

Appium - wait for application or activity to load

https://stackoverflow.com/questions/28799420/how-to-wait-to-activity-using-appium-on-begin-and-during-test-itself

https://discuss.appium.io/t/do-we-have-wait-for-activity-to-be-loaded/11925/5

Appium Multiple Activities vs Switching Views vs Fragments

- Retrieving the current running activity, returns the name of the current activity running on the device

```
activity = driver.current_activity
package = driver.current_package
```

android app design: Activity, Fragment, View

sample code

Appium Launch and Close App

```
driver.close_app()
driver.launch_app()
```