# Sravanthi Madhapuram 23/6/2016

# Understanding Desired Capabilities

Before we start writing further test code to automate and do other stuff, we need to understand why we are using Desired Capabilities.

Desired Capabilities are present in the below library and hence it needs to be imported.

org.openqa.selenium.remote.DesiredCapabilities

Desired Capabilities is a way of telling the Appium Server which kind of session we are interested in. So it's a hash of Key and Value to let us have more control on the server during automation. So some of the key info which we will be using are :

automationName Which automation engine to use Appium (default) or Selendroid

platformName Which mobile OS platform to use iOS, Android, or FirefoxOS

Similarly, we can specify *platformVersion* which will be the version of OS running on the device, could be "8.3" or "5.1".

Below is one such table which contains the various Desired Capabilities which would be set at the server level and customized to either Android or iOS.

| **Capability** | **Description** | **Values** |
| --- | --- | --- |
| automationName | Which automation engine to use | Appium (default) or Selendroid |
| platformName | Which mobile OS platform to use | iOS, Android, or FirefoxOS |
| platformVersion | Mobile OS version | e.g., 7.1, 4.4 |
| deviceName | The kind of mobile device or emulator to use | iPhone Simulator, iPad Simulator,  iPhone Retina 4-inch,  Android Emulator, Galaxy S4, etc… |
| App | The absolute local path or remote http URL to an .ipa or .apk file, or a .zip containing one of these. Appium will attempt to install this app binary on the appropriate device first. Note that this capability is not required for Android if you specify appPackage and appActivity capabilities (see below). Incompatible with browserName. | /abs/path/to/my.apk orhttp://myapp.com/app.ipa |
| browserName | Name of mobile web browser to automate. Should be an empty string if automating an app instead. | ‘Safari’ for iOS and ‘Chrome’,  ‘Chromium’, or ‘Browser’ for Android |
| newCommandTimeout | How long (in seconds) Appium will wait for a new command from the client before assuming the client quit and ending the session | e.g. 60 |
| autoLaunch | Whether to have Appium install and launch the app automatically. Default true | true, false |
| language | (Sim/Emu-only) Language to set for the iOS Simulator | e.g. fr |
| Locale | (Sim/Emu-only) Locale to set for the iOS Simulator | e.g. fr\_CA |
| Udid | Unique device identifier of the connected physical device | e.g. 1ae203187fc012g |
| orientation | (Sim/Emu-only) start in a certain orientation | LANDSCAPE or PORTRAIT |
|  |  |  |
| autoWebview | Move directly into Webview context. Default false | true, false |

**Desired Capabilities for Android**

Since **Appium** caters to both Android and iOS, there are different set of desired capabilities for Android and iOS.

This section will list all the desired capabilities associated with Android. Majority of them are optional but you can choose to use them as it suits your needs.

| **Capability** | **Description** | **Values** |
| --- | --- | --- |
| appActivity | Activity name for the Android activity you want to launch from your package | MainActivity, .Settings |
| appPackage | Java package of the Android app you want to run | com.example.android.myApp,  com.android.settings |
| appWaitActivity | Activity name for the Android activity you want to wait for | SplashActivity |
| appWaitPackage | Java package of the Android app you want to wait for | com.example.android.myApp, com.android.settings |
| deviceReadyTimeout | Timeout in seconds while waiting for device to become ready | 5 |
| androidCoverage | Fully qualified instrumentation class. Passed to -w in adb shell am instrument -e coverage true -w | com.my.Pkg/com.my.  Pkg.instrumentation.  MyInstrumentation |
| enablePerformanceLogging | (Chrome and webview only) Enable Chromedriver’s performance logging (default false) | true, false |
| androidDeviceReadyTimeout | Timeout in seconds used to wait for a device to become ready after booting | e.g., 30 |
| androidDeviceSocket | Devtools socket name. Needed only when tested app is a Chromium embedding browser. The socket is open by the browser and Chromedriver connects to it as a devtools client. | e.g., chrome\_devtools\_remote |
| avd | Name of avd to launch | e.g., api19 |
| avdLaunchTimeout | How long to wait in milliseconds for an avd to launch and connect to ADB (default 120000) | 300000 |
| avdReadyTimeout | How long to wait in milliseconds for an avd to finish its boot animations (default 120000) | 300000 |
| avdArgs | Additional emulator arguments used when launching an avd | e.g., -netfast |
| useKeystore | Use a custom keystore to sign apks, default false | true or false |
| keystorePath | Path to custom keystore, default ~/.android/debug.keystore | e.g., /path/to.keystore |
| keystorePassword | Password for custom keystore | e.g., foo |
| keyAlias | Alias for key | e.g., androiddebugkey |
| keyPassword | Password for key | e.g., foo |
| chromedriverExecutable | The absolute local path to webdriver executable (if Chromium embedder provides its own webdriver, it should be used instead of original chromedriver bundled with Appium) | /abs/path/to/webdriver |
| specialChromedriverSessionArgs | Custom arguments passed directly to chromedriver in chromeOptions capability. Passed as object which properties depend on a specific webdriver. | e.g., {'androidDeviceSocket': 'opera\_beta\_devtools\_remote',} |
| autoWebviewTimeout | Amount of time to wait for Webview context to become active, in ms. Defaults to 2000 | e.g. 4 |
| intentAction | Intent action which will be used to start activity (default android.intent.action.MAIN) | e.g.android.intent.action.MAIN, android.intent.action.VIEW |
| intentCategory | Intent category which will be used to start activity (default android.intent.category.LAUNCHER) | e.g. android.intent.category.LAUNCHER, android.intent.category.APP\_CONTACTS |
| intentFlags | Flags that will be used to start activity (default 0x10200000) | e.g. 0x10200000 |
| optionalIntentArguments | Additional intent arguments that will be used to start activity. See Intent arguments | e.g. --esn , --ez , etc. |
| unicodeKeyboard | Enable Unicode input, default false | true or false |
| resetKeyboard | Reset keyboard to its original state, after running Unicode tests with unicodeKeyboard capability. Ignored if used alone. Default false | true or false |

If you notice above some of the capabilities like *appWaitActivity* , *avd*, *androidDeviceReadyTimeout* are very handy and would be recommended to make use of in your automation suite.