

# A PAGE TALKS ABOUT

By  
reWire  
fueled by decode

# A Kickstarter Tool kit follows ....

appium server 2.x



## ❖ Prerequisite

❖ Node JS

❖ Android Environment



❖ Appium Ecosystem

❖ Server

❖ Appium Inspector

❖ Drivers

❖ Client



❖ Kick-Start

❖ Demo-Apps

❖ Automation Readiness

# A Kickstarter Tool Kit

Mobile Apps Automation



@Implement -Playlist

OPENSOURCE

## Appium Server 2.x

Learning & Adapt Made it easy !!!!



This is an attempt to simplify the ecosystem

By  
reWire

fueled by decode

## Appium Server

### CHAPTER 1

- ☐ CLI
  - Server
  - Extension
- ☐ Management
  - Driver / Plugin

## Appium Inspector

### CHAPTER 2

- ☐ Inspector Interface
  - Session
- ☐ Capabilities
  - App
  - Driver

## Selectors

### CHAPTER 3

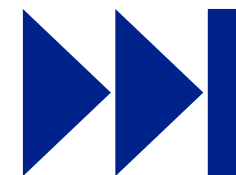
- ☐ Android Strategies
- ☐ IOS Strategies
- ☐ Define & Execute Strategies on Inspector

## Implementation

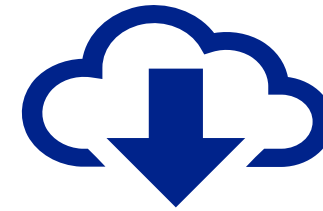
### CHAPTER 4

- ☐ Element Operations
- ☐ UiScrollable
- ☐ Mobile Gestures
- ☐ Web Context (Hybrid App)
- ☐ Platform Specific Extensions

# Session Ends



CHAPTER 1



# Appium Server

# Decode Appium Server

---

Appium Server

CLI

Server CLI

Extension CLI

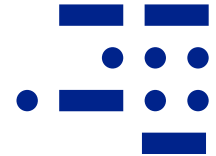
---

Management

Driver

Plugin

---



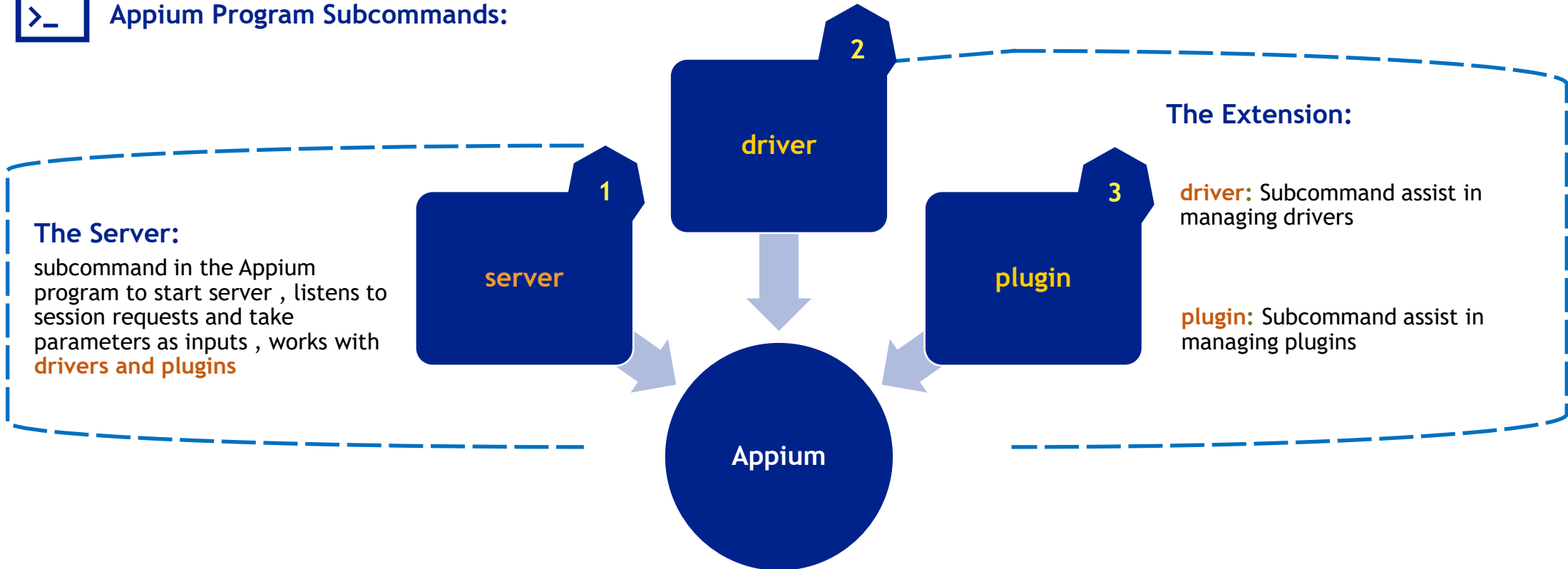


# The Server CLI - Introduction





## Appium Program Subcommands:



## Appium Arguments:

Function	Syntax	Ex:	Description
Server Start	: \> [command]	appium	Invoke appium server on localhost by default
Server Start	: \> [command] server	appium server	Inovek appium server local host by default
Server Start	: \> [command] -address "IP Address" -port "Port No"  : \> [command] -a "IP Address" -p "Port No"	appium -address 127.0.0.1 -port 4723	Invoke appium server on specified IP and Port  <b>Notes:</b> Command to know available ports netstat -ano Status: Listening - Under use Established - Available Specific port check netstat -a -n -o   find "9000"
Accept Appium server from Web browser connections	: \> [command] --allow-cors	appium --allow-cors	Allows web connections from browser for <b>ex: Web Appium Inspector client</b>  <a href="https://inspector.appiumpro.com/">https://inspector.appiumpro.com/</a>
Server and Connection time out to all requests	: \> [command] -keep-alive-timeout <Integer No>  : \> [command] -ka <Integer No>	appium -ka 500	<b>By Default, 600 sec is time out for server and connection requests</b>
Generate Log File	: \> [command] --log <Path to log file>  : \> [command] -g <Path to log file>	appium --log C:\Appiumlogs\Session1  appium -g C:\Appiumlogs\Session1	<b>Send server log output to log file</b>

1

```
C:\>appium
[Appium] Welcome to Appium v2.1.3
[Appium] Attempting to load driver xcuitest...
[debug] [Appium] Requiring driver at C:\Users\ [redacted] \.appium\node_modules\appium-xcuitest-driver
[Appium] Attempting to load driver uiautomator2...
[debug] [Appium] Requiring driver at C:\Users\ [redacted] \.appium\node_modules\appium-uiautomator2-driver
[Appium] Appium REST http interface listener started on http://0.0.0.0:4723
[Appium] You can provide the following URLs in your client code to connect to this server:
[Appium]   http://172.24.37.20:4723/
[Appium]   http://10.237.15.10:4723/
[Appium]   http://127.0.0.1:4723/ (only accessible from the same host)
[Appium] Available drivers:
[Appium]   - xcuitest@4.35.0 (automationName 'XCUIest')
[Appium]   - uiautomator2@2.29.10 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
```

2

```
C:\>appium server
[Appium] Welcome to Appium v2.1.3
[Appium] Attempting to load driver xcuitest...
[debug] [Appium] Requiring driver at C:\Users\ [redacted] \.appium\node_modules\appium-xcuitest-driver
[Appium] Attempting to load driver uiautomator2...
[debug] [Appium] Requiring driver at C:\Users\ [redacted] \.appium\node_modules\appium-uiautomator2-driver
[Appium] Appium REST http interface listener started on http://0.0.0.0:4723
[Appium] You can provide the following URLs in your client code to connect to this server:
[Appium]   http://172.24.37.20:4723/
[Appium]   http://10.237.15.10:4723/
[Appium]   http://127.0.0.1:4723/ (only accessible from the same host)
[Appium] Available drivers:
[Appium]   - xcuitest@4.35.0 (automationName 'XCUIest')
[Appium]   - uiautomator2@2.29.10 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
```

3

```
C:\>appium -a 127.0.0.1 -p 4725
[Appium] Welcome to Appium v2.1.3
[Appium] Non-default server args:
[Appium] {
[Appium]   address: '127.0.0.1',
[Appium]   port: 4725
[Appium] }
[Appium] Attempting to load driver xcuitest...
[debug] [Appium] Requiring driver at C:\Users\ [redacted] \.appium\node_modules\appium-xcuitest-driver
[Appium] Attempting to load driver uiautomator2...
[debug] [Appium] Requiring driver at C:\Users\ [redacted] \.appium\node_modules\appium-uiautomator2-driver
[Appium] Appium REST http interface listener started on http://127.0.0.1:4725
[Appium] Available drivers:
[Appium]   - xcuitest@4.35.0 (automationName 'XCUIest')
[Appium]   - uiautomator2@2.29.10 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
```

4

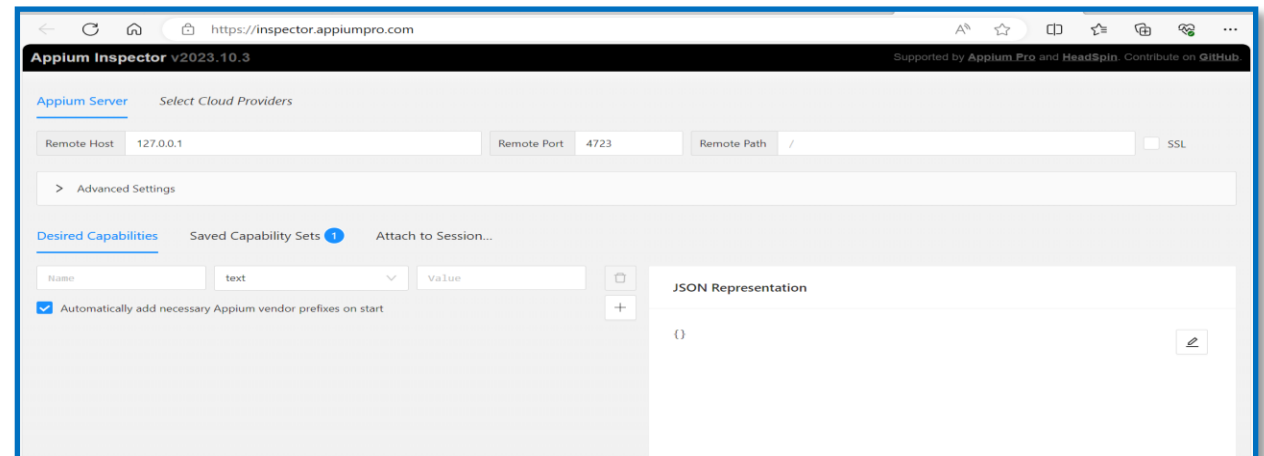
```
C:\>npm
C:\>appium --allow-cors
[Appium] Welcome to Appium v2.1.3
[Appium] Non-default server args:
[Appium] {
[Appium]   allowCors: true
[Appium] }
[Appium] Attempting to load driver xcuitest...
[debug] [Appium] Requiring driver at C:\Users\<redacted>\.appium\node_modules\appium-xcuitest-driver
[Appium] Attempting to load driver uiautomator2...
[debug] [Appium] Requiring driver at C:\Users\<redacted>\.appium\node_modules\appium-uiautomator2-driver
[Appium] You have enabled CORS requests from any host. Be careful not to visit sites which could maliciously try to start Appium sessions on your machine
[Appium] Appium REST http interface listener started on http://0.0.0.0:4723
[Appium] You can provide the following URLs in your client code to connect to this server:
[Appium]   http://172.24.37.20:4723/
[Appium]   http://10.237.15.10:4723/
[Appium]   http://127.0.0.1:4723/ (only accessible from the same host)
[Appium] Available drivers:
[Appium]   - xcuitest@4.35.0 (automationName 'XCUITest')
[Appium]   - uiautomator2@2.9.10 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
```

Start Appium Server using below flag:

**appium -allow-cors**

Launch Browser: Invoke below URL

<https://inspector.appiumpro.com>



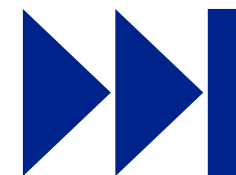
5

```
C:\>appium -ka 500
[Appium] Welcome to Appium v2.1.3
[Appium] Non-default server args:
[Appium] {
[Appium]   keepAliveTimeout: 500
[Appium] }
[Appium] Attempting to load driver xcuitest...
[debug] [Appium] Requiring driver at C:\Users\ [REDACTED] \.appium\node_modules\appium-xcuitest-driver
[Appium] Attempting to load driver uiautomator2...
[debug] [Appium] Requiring driver at C:\Users\ [REDACTED] \.appium\node_modules\appium-uiautomator2-driver
[Appium] Appium REST http interface listener started on http://0.0.0.0:4723
[Appium] You can provide the following URLs in your client code to connect to this server:
[Appium]   http://172.24.37.20:4723/
[Appium]   http://10.237.15.10:4723/
[Appium]   http://127.0.0.1:4723/ (only accessible from the same host)
[Appium] Available drivers:
[Appium]   - xcuitest@4.35.0 (automationName 'XCUITest')
[Appium]   - uiautomator2@2.29.10 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
```

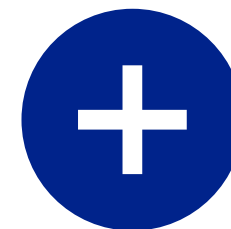
6

```
C:\>appium -g c:\kitemp\session
[Appium] Welcome to Appium v2.1.3
[Appium] Attempting to load driver xcuitest...
[debug] [Appium] Requiring driver at C:\Users\ [REDACTED] \.appium\node_modules\appium-xcuitest-driver
[Appium] Attempting to load driver uiautomator2...
[debug] [Appium] Requiring driver at C:\Users\ [REDACTED] \.appium\node_modules\appium-uiautomator2-driver
[Appium] Appium REST http interface listener started on http://0.0.0.0:4723
[Appium] You can provide the following URLs in your client code to connect to this server:
[Appium]   http://172.24.37.20:4723/
[Appium]   http://10.237.15.10:4723/
[Appium]   http://127.0.0.1:4723/ (only accessible from the same host)
[Appium] Available drivers:
[Appium]   - xcuitest@4.35.0 (automationName 'XCUITest')
[Appium]   - uiautomator2@2.29.10 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
--> OPTIONS /session/2d088911-486e-4bb4-95ed-0170ae0a0e63/timeouts
```

# Session Ends



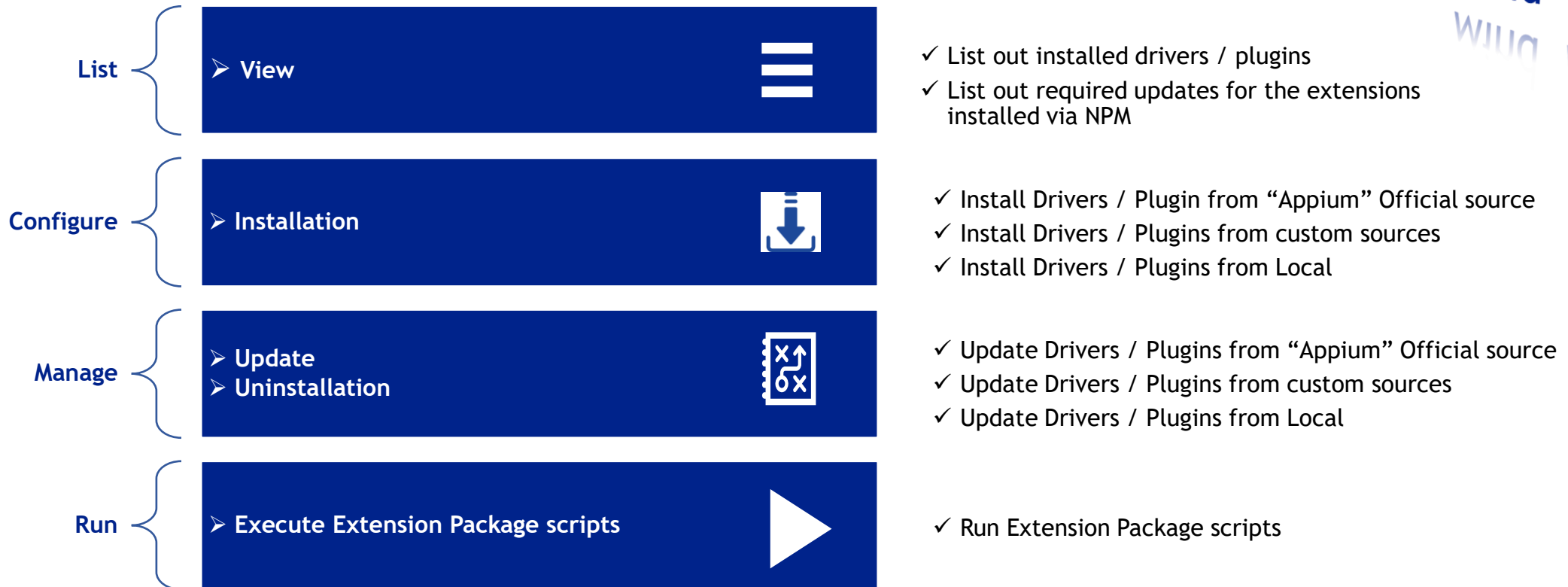
# The Extension CLI



Driver / Plugin



Mind Map  
willig wgb



----- Four Extents -----

# List @Driver / Plugin

## List:

Syntax: appium <extType> list --installed

appium driver list --installed

1

```
C:\windows\system32\cmd.exe

C:\>appium driver list --installed
✓ Listing installed drivers
- xcuitest@4.35.0 [installed (npm)]
- uiautomator2@2.29.10 [installed (npm)]
```

Syntax: appium <extType> list --updates

appium driver list --updates

2

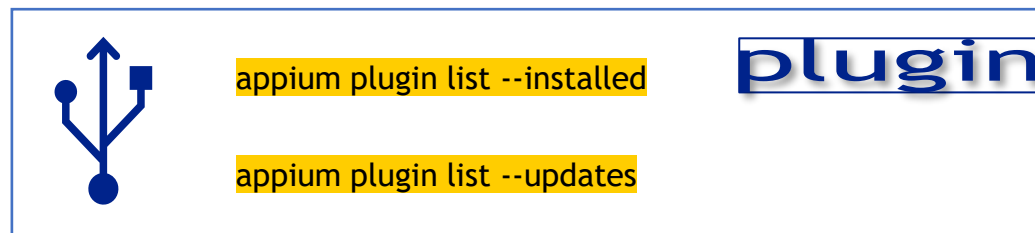
```
C:\windows\system32\cmd.exe

C:\>appium driver list --updates
✓ Listing available drivers
- xcuitest@4.35.0 [installed (npm)] [5.8.2 available (potentially unsafe)]
- uiautomator2@2.29.10 [installed (npm)] [2.34.1 available]
- mac2 [not installed]
- espresso [not installed]
- safari [not installed]
- gecko [not installed]
- chromium [not installed]
```

Syntax: appium <extType> list

appium driver list

(List out drivers from Appium source)



```
C:\windows\system32\cmd.exe

C:\>appium plugin list --installed
✓ Listing installed plugins

C:\>appium plugin list --updates
✓ Listing available plugins
- images [not installed]
- execute-driver [not installed]
- relaxed-caps [not installed]
- universal-xml [not installed]
```

## Installation:

**Syntax:** appium <extType> install <extName> --source=<source install> --package=<package name>

appium driver install xcuitest



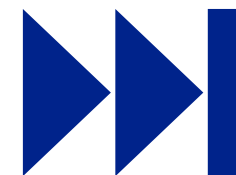
✓ By default, extension will be installed from Appium official source

## Source Type as below:

None	No source specified
npm	npm package name with modifiers
github	Install extension via github > <b>org/repo</b>
git	Git URL > [git+ssh://host.com/repo.git]
local	Install extension from local path. This should be from Node.JS Project [Driver Package Information]

Prompt	Command
Install Latest Driver:\>	appium driver install uiautomator2
Install a Driver @version:\>	appium driver install uiautomator2@2.34.1
Install Driver beta version from NPM:\>	appium driver install -source=npm @appium/<driver name>@beta
Install locally developed drive:\>	appium driver install -source=local /path to local

# Session Ends



# Management



Driver / Plugin



## Update:

**Syntax:** appium <extType> update <extName>

`appium driver update uiautomator2`



- ✓ By default, Appium will not update drivers
- ✓ Update drivers that have been installed via NPM

## Uninstall:

**Syntax:** appium <extType> uninstall <extName>

`appium driver uninstall uiautomator2`

`appium plugin uninstall images`



## Opensource Drivers Updates Best Practices

- ✓ Understand specific version release log
- ✓ Review updated methods
- ✓ Understand impact on the developed suite
- ✓ Determine the need for change and the effort required
- ✓ Have Practice updating drivers periodically (once a 3 months)
- ✓ Before updating take the previous suite as a backup
- ✓ Install the latest version of drivers/plugins
- ✓ Implement driver/plugin updates at component level
- ✓ Conduct dry runs of the suite and review results, take appropriate action

```
C:\>appium driver update uiautomator2
✓ Checking if driver 'uiautomator2' is updatable
✓ Checking if driver 'uiautomator2' needs an update
- Updating driver 'uiautomator2' from 2.29.10 to 2.34.1- Installing 'uiautomator2' using NPM install spec 'appium-uiauto✓ Installing 'uiautomator2' using NPM install spec
appium-uiautomator2-driver@2.34.1'
✓ Updating driver 'uiautomator2' from 2.29.10 to 2.34.1
i Update report:
✓ - driver uiautomator2 updated: 2.29.10 => 2.34.1
```



```
C:\Users\ [redacted] >appium driver list --installed
√ Listing installed drivers
- xcuitest@4.35.0 [installed (npm)]
- uiautomator2@2.34.1 [installed (npm)]

C:\Users\ [redacted] >appium driver uninstall uiautomator2
√ Successfully uninstalled driver 'uiautomator2'

C:\Users\ [redacted] >appium driver list --installed
√ Listing installed drivers
- xcuitest@4.35.0 [installed (npm)]

C:\Users\ [redacted] >appium driver install uiautomator2@2.34.1
√ Installing 'uiautomator2@2.34.1' using NPM install spec 'appium-uiautomator2-driver@2.34.1'
i Driver uiautomator2@2.34.1 successfully installed
- automationName: UiAutomator2
- platformNames: ["Android"]
```

run:

**Syntax:** appium <extType> run <extName> [Script Name] [Script-args]

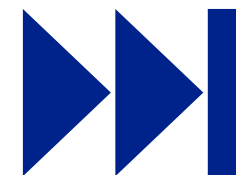
appium driver run uiautomator2 reset



- ✓ Scripts included in the Extension packages
- ✓ Scripts assist or perform intended task as required
- ✓ Extension authors will provide the required documentation along with drivers/plugins



# Session Ends



CHAPTER 2



# Appium Inspector

# Decode Appium Inspector



---

## Appium Inspector

### Inspector Interface

Inspector Flow

Inspector Key Features

---

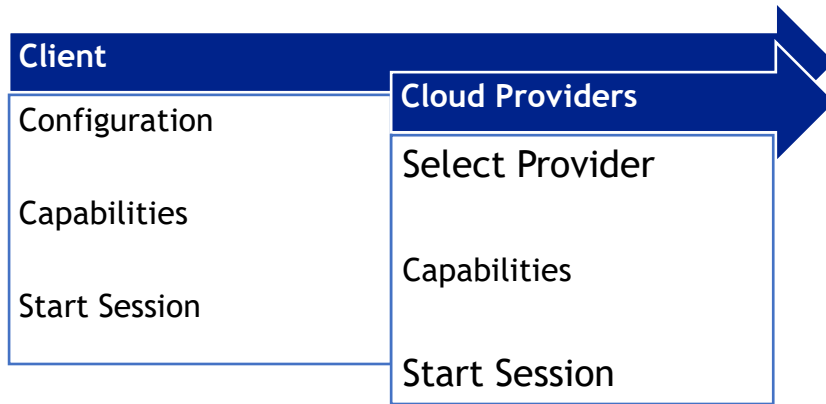
### Capabilities

As a Parameters

Format

---

# Inspector Interface

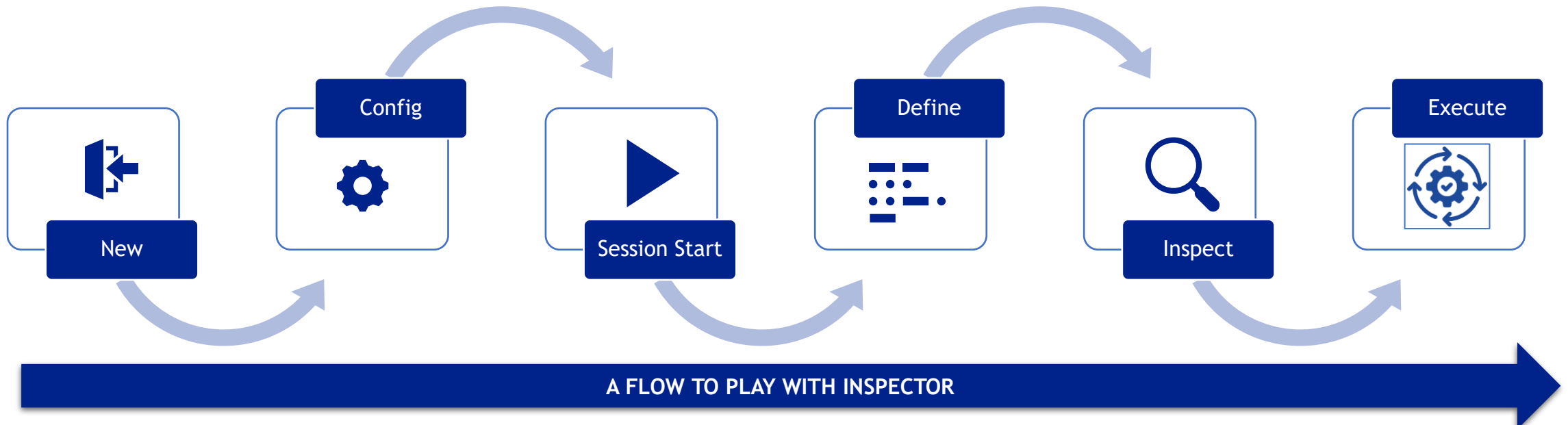


1

As a Pre-requisite Appium Server and Emulator / Device should be On

**Appium Server:** A Local Server / A Remote Server, Appium enables the execution of Mobile app operations with its capabilities

**Cloud Providers:** Enable seamless connectivity to a Cloud-based Appium server for executing mobile application operations on remote cloud devices.



## Config:

- ✓ Define Capabilities to sessions
- ✓ Save Capabilities for future sessions

## Cloud Support:

- ✓ Mobile Cloud Solutions Integration
- ✓ Define Capabilities to Cloud sessions

## Define:

- ✓ Identify Locator Strategies
- ✓ Android and IOS Platforms / Hybrid
- ✓ Test Custom Locator Strategies

## Inspect:

- ✓ Search for Elements in the app screen

## Elements Interact:

- ✓ Tap on Elements
- ✓ Values enter / clear



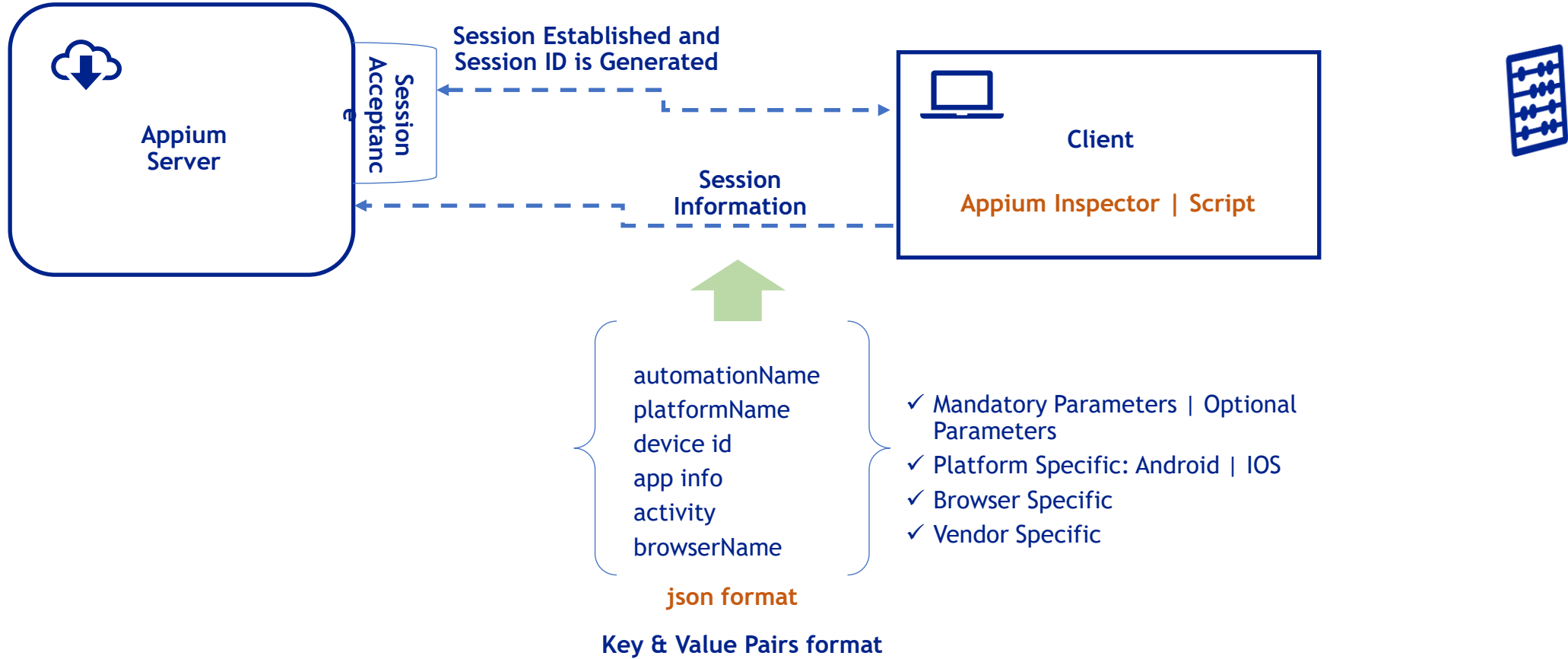
## Record:

- ✓ Generate code samples as per chosen language
- ✓ Generate element level action code samples
- ✓ Generate script with defined capabilities and driver instance creation and driver / element actions

# Capabilities



## Capabilities / Options





## Android:

### @Appium Inspector

```
{
  "appium:platformName": "Android",
  "appium:automationName": "uiautomator2",
  "appium:appPackage": "com.android.settings",
  "appium:appActivity": ".Settings",
  "appium:deviceName": "emulator-5554",
  "appium:udid": "emulator-5554",
  "appium:noReset": false,
  "appium:fullReset": false,
  "appium:printPageSourceOnFindFailure": true
}
```

“appium:app”: “path to apk file”

### @Code

```
UiAutomator2Options options = new UiAutomator2Options();
options.setAutomationName("UiAutomator2");
options.setUdid("emulator-5554");
options.setAppPackage("io.appium.android.apis");
options.setAppActivity("io.appium.android.apis.ApiDemos");
options.setPlatformName("Android");
options.setPlatformVersion("14");
options.setNoReset(false);
options.setFullReset(false);
options.setNewCommandTimeout(Duration.ofSeconds(500));
options.setPrintPageSourceOnFindFailure(true);
```

## IOS:

“appium:bundleId”: <IOS App bundle ID>

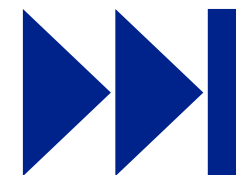
### Browser:

browserName

browserVersion

platformName

# Session Ends



CHAPTER 3

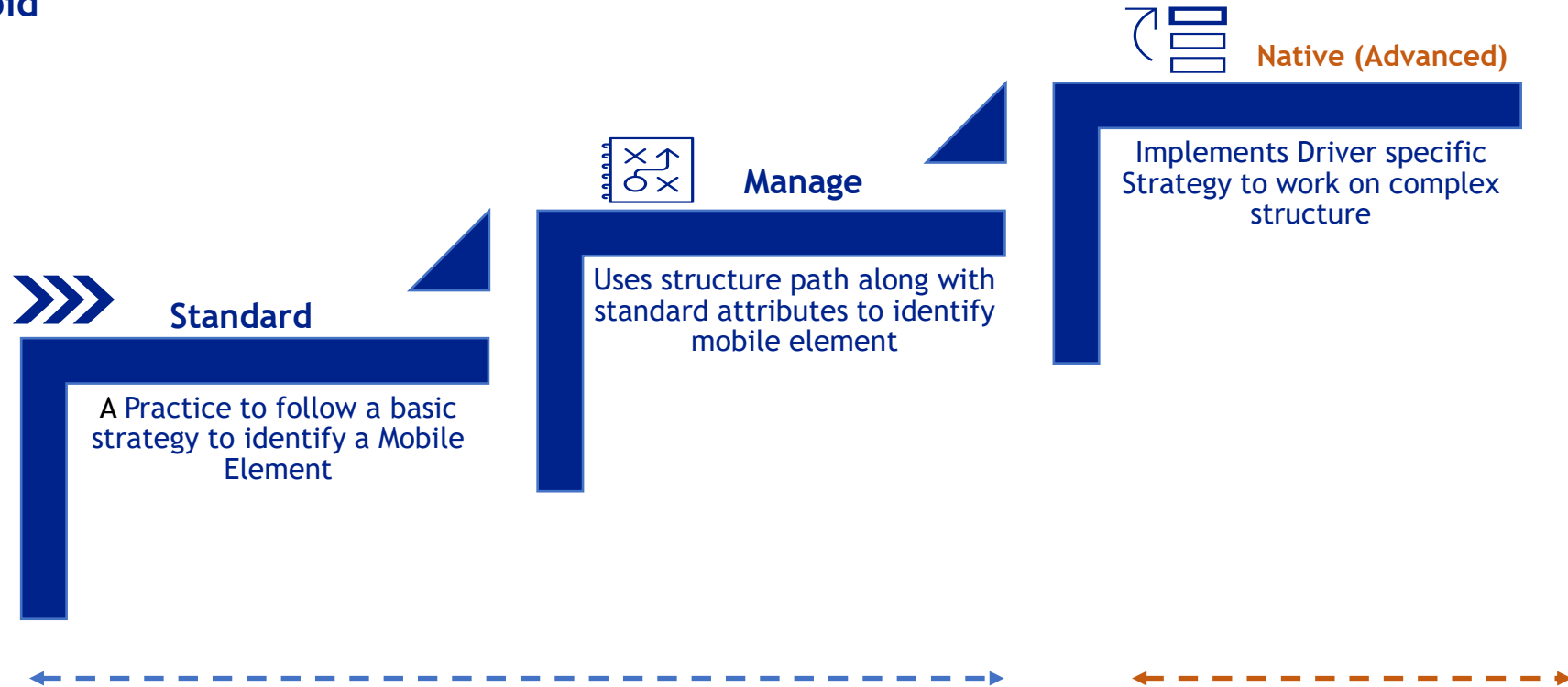


# Selector Strategies

# Decode Selectors



## Android



Standard				
Platform	Selector Strategy	Attribute	Applicability	Description
Android	ID	Resource-id		A unique-id assigned to an element
Android	Accessibility ID	Content-desc		Value of Accessibility is same as the value of the attribute "content-desc"
Android	Class Name	Class Name	Will be used along with other locators	Class Name is called as full name of the UIAutomator2 class Ex: android.widget.TextView
Manage				
Android	Xpath		Applicable only when standard strategy is not suitable	Locates Element based on XML structure of App
Native (Advanced)				
Android	UiAutomator			

## AppiumBy.accessibilityId

```
WebElement element = driver.findElement(AppiumBy.accessibilityId("App"));
```

## AppiumBy.Id

```
WebElement element = driver.findElement(AppiumBy.id("io.appium.android.apis:id/start_alarm"));  
System.out.println("Element Start Alarm Text By ID:" + element.getText());  
element = driver.findElement(AppiumBy.id("io.appium.android.apis:id/stop_alarm"));  
System.out.println("Element Stop Alarm Text By ID:" + element.getText());
```

## AppiumBy.className

```
List<WebElement> elementList = driver.findElements(AppiumBy.className("android.widget.TextView"));  
System.out.println("Element List Size:" + elementList.size());  
  
for(WebElement element: elementList) {  
    System.out.println("Element Text: " + element.getText());  
}
```

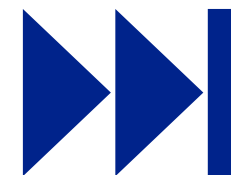
```
WebElement element = driver.findElement(AppiumBy.xpath("//android.widget.TextView[@content-desc='App\\']"));
```

AppiumBy.xpath

```
WebElement element = driver.findElement(AppiumBy.xpath("//android.widget.TextView[@content-desc=\"App\"]));
```



# Session Ends





# Android -UiAutomator

Advanced Locator Strategy @Native

Android:

Strategy: **UiAutomator** - **ClassName**

Facilitate Android Advanced operations on Mobile Elements

```
className(String Class_Name)
```

```
List<WebElement> elementList = driver.findElements(AppiumBy.androidUIAutomator("new  
UiSelector().className(\"android.widget.TextView\")"));
```

```
System.out.println("Element List Size:" + elementList.size());
```

**className**



Inspect based on Class  
Name of mobile  
elements

**Appium Inspector**

Search for element

Locator Strategy:

Id	XPath	Name	Class Name
	Accessibility ID	UIAutomator	

Selector:

new UiSelector().className("android.widget.TextView")

Search

Android:

Strategy: **UiAutomator - resourceId**

Facilitate Android Advanced operations on Mobile Elements

## ResourceId

resourceId(String Id)

```
List<WebElement> elementList = driver.findElements(AppiumBy.androidUIAutomator("new  
UiSelector().resourceId(\"android:id/text1\")"));
```

```
System.out.println("Element List Size:" + elementList.size());
```

Appium Inspector

Search for element

Locator Strategy:

Id	XPath	Name	Class Name
	Accessibility ID	UIAutomator	

Selector:

new UiSelector().resourceId("android:id/text1")

Search



Inspect based on  
Resource ID of mobile  
elements

Android:

## packageName

Strategy: **UiAutomator** - **packageName**

Facilitate Android Advanced operations on Mobile Elements

`packageName(String Pname))`

```
List<WebElement> elementList = driver.findElements(AppiumBy.androidUIAutomator("new  
UiSelector().packageName(\"io.appium.android.apis\")");
```

```
System.out.println("Element List Size:" + elementList.size());
```



Search for element

Locator Strategy:

Id	XPath	Name	Class Name
	Accessibility ID	UIAutomator	

Selector:

new UiSelector().packageName("io.appium.android.apis")

Search

Android:

Strategy: **UIAutomator -description**

Facilitate Android Advanced operations on Mobile Elements

content-desc

description(String Content\_Desc))

```
WebElement element = driver.findElement(AppiumBy.androidUIAutomator("new UiSelector().description(\"App\")"));
```

```
System.out.println("Text of Mobile Element:" + element.getText());
```

Locates Mobile Elements  
based on **content-desc**  
attribute value

Appium Inspector

Search for element

Locator Strategy:

Id	XPath	Name	Class Name
	Accessibility ID	UIAutomator	

Selector:

new UiSelector().description("App")

Search

Android:

Strategy: **UIAutomator -text**

Facilitate Android Advanced operations on Mobile Elements

text

text (String Name)

```
WebElement element = driver.findElement(AppiumBy.androidUIAutomator("new UiSelector().text(\"App\")"));
```

```
System.out.println("Text of Mobile Element:" + element.getText());
```

Appium Inspector



Inspect Mobile Elements  
based on **text** that  
appeared on the current  
app screen

Search for element

Locator Strategy:

Id	XPath	Name	Class Name
	Accessibility ID	UIAutomator	

Selector:

new UiSelector().resourceId("android:id/text1")

Search



## Additional UiSelector methods:

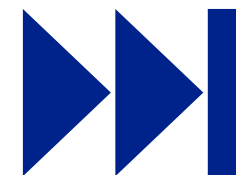
- ✓ checkable(boolean val)
- ✓ checked(boolean val)
- ✓ checked (boolean val)
- ✓ childSelector(UiSelector selector)
- ✓ classNameMatches(String regex)
- ✓ Clickable(boolean val)
- ✓ enabled(boolean val)
- ✓ focusable(boolean val)
- ✓ to String()
- ✓ cloneSelector()
- ✓ focused(boolean val)
- ✓ fromParent(UiSelector selector)
- ✓ index(int index)
- ✓ instance(int instance)
- ✓ longclickable(boolean val)
- ✓ scrollable(boolean val)
- ✓ selected(boolean val)
- ✓ packageNameMatches(String regex)
- ✓ resourceIdMatches(String regex)
- ✓ descriptionContains(String desc)
- ✓ descriptionmatches(String regex)
- ✓ descriptionStartsWith(String desc)
- ✓ textContains(String text)
- ✓ textMatches(String regex)
- ✓ textStartsWith(String text)



All methods returns UiSelector / element



# Session Ends

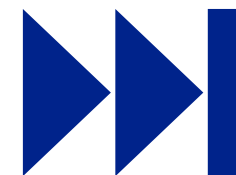


# IOS - Selector Strategies



Standard				
Platform	Selector Strategy	Attribute	Applicability	Description
IOS	ID	Name		
IOS	Accessibility ID	Name		The default Accessibility ID set to the name of the UI Element
IOS	Class Name	Class Name	Will be used along with other locators	Full Name name of XCUI Element, starts with XCUIElementType
Manage				
IOS	Xpath		Applicable only when standard strategy is not suitable	Locates Element based on XML structure of App
Native (Advanced)				
IOS	BylosClassChain BylosNSPredicate			

# Session Ends



CHAPTER 4

# Implementation



## Elements Interact:

- ✓ Element Operations

## UiScrollable:

- ✓ No Selector Specification
- ✓ With Selector Specification
- ✓ Text Specification

## Mobile Gestures:

- ✓ Drag & Drop
- ✓ Swipe (Left / Right)
- ✓ Scroll (Up / Down)

## Web Context:

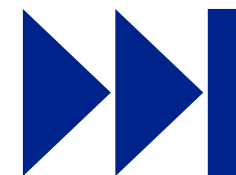
- ✓ Hybrid App

## Platform Specific Extensions:

- ✓ Device Information
- ✓ App State
- ✓ Keyboard
- ✓ Press Keys
- ✓ App Management



# Session Ends

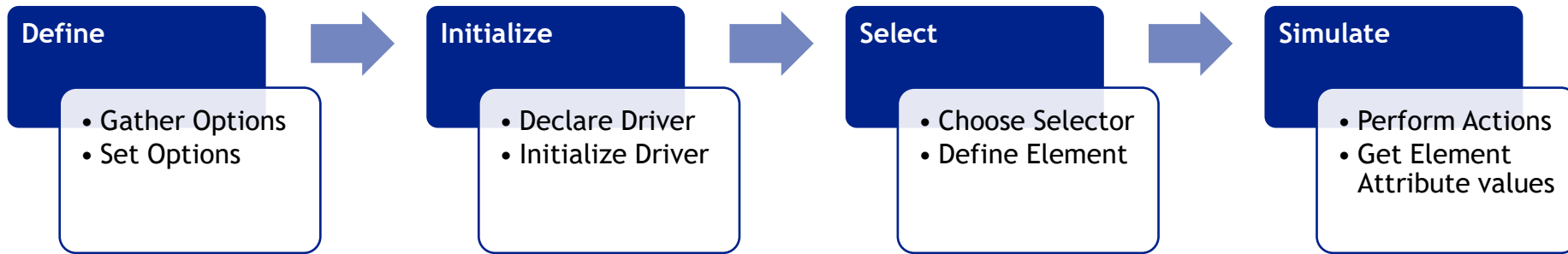


## Element Operations





## Script Flow:

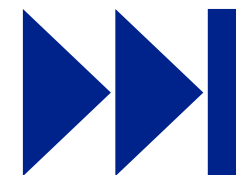


Mind Map  
will wgb

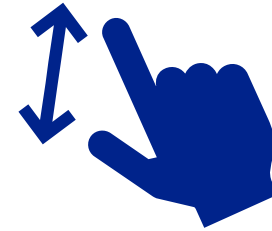


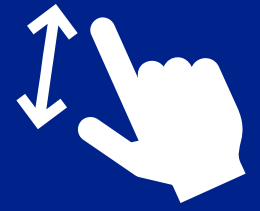
- ✓ Click Operation
- ✓ Grouping Elements and Click on desired element
- ✓ Enter values in the Edit boxes
- ✓ Get Attribute values
- ✓ Perform Mobile Gestures like Scroll, Swipe, and Drag & Drop
- ✓ Press Key Board Keys
- ✓ Simulate various control operations like Radio button, Checkboxes etc

# Session Ends



**UiScrollable**

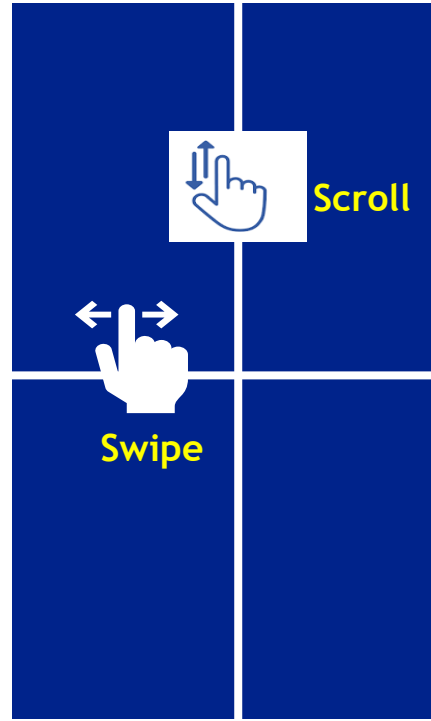
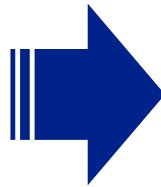




Scroll / Swipe Operations:

## UiScrollable

Mind Map  
WUIG WGB



No  
Specification

- `scrollToEnd`
- `scrollToBeginning`

With  
Specification

- `UiSelector`
- `Text`

### scrollToEnd:

```
driver.findElement(new AppiumBy.ByAndroidUIAutomator("new UiScrollable(new UiSelector().scrollable(true).instance(0)).scrollToEnd(3)"));
```

### scrollToBeginning:

```
driver.findElement(new AppiumBy.ByAndroidUIAutomator("new UiScrollable(new UiSelector().scrollable(true).instance(0)).scrollToBeginning(3)"));
```

### Text Specification

`scrollTextIntoView(String text):`

Scroll to Last Element of the screen from top to bottom

```
driver.findElement(new AppiumBy.ByAndroidUIAutomator("new UiScrollable(new UiSelector().scrollable(true).instance(0)).scrollTextIntoView(\"WebView3\")"));
```



Scroll to First Element of the screen from bottom to top

```
driver.findElement(new AppiumBy.ByAndroidUIAutomator("new UiScrollable(new UiSelector().scrollable(true).instance(0)).scrollTextIntoView(\"Animation\")"));
```



## UiSelector Specification

`scrollIntoView(UiSelector selector):`

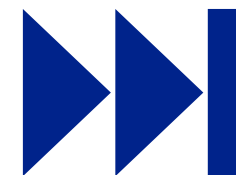
Scroll to Last Element of the screen from top to bottom

```
driver.findElement(new AppiumBy.ByAndroidUIAutomator("new UiScrollable(new  
UiSelector().scrollable(true).instance(0)).scrollIntoView(textMatches(\"WebView3\").instance(0));"));
```

Scroll to First Element of the screen from bottom to top

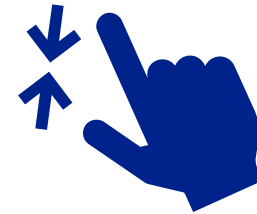
```
driver.findElement(new AppiumBy.ByAndroidUIAutomator("new UiScrollable(new  
UiSelector().scrollable(true).instance(0)).scrollIntoView(textMatches(\"Animation\").instance(0));"));
```

# Session Ends



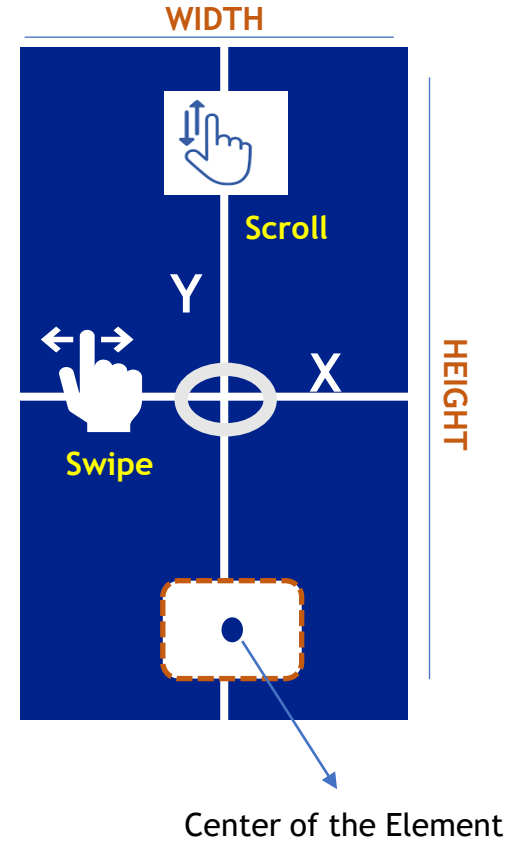
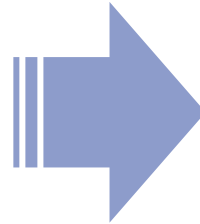


## Mobile Gestures





- ✓ Swipe
- ✓ Scroll
- ✓ Drag
- ✓ LongClick
- ✓ Pinch
- ✓ Fling



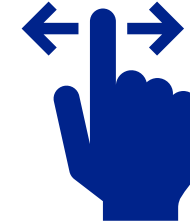
Mind Map

**Note:** This is also included part of Platform Specific Extensions

```
WebElement elementCard = driver.findElement(AppiumBy.xpath("(//android.view.ViewGroup[@content-desc=\"card\"])[1]"));
```

*//Left Swipe*

```
((JavascriptExecutor) driver).executeScript("mobile: swipeGesture", ImmutableMap.of(  
    "elementId", ((RemoteWebElement) elementCard).getId(),  
    "direction", "left",  
    "percent", 0.5,  
    "speed", 500  
));
```



```
elementCard = driver.findElement(AppiumBy.xpath("(//android.view.ViewGroup[@content-desc=\"card\"])[2]"));
```

*//right Swipe*

```
((JavascriptExecutor) driver).executeScript("mobile: swipeGesture", ImmutableMap.of(  
    "elementId", ((RemoteWebElement) elementCard).getId(),  
    "direction", "right",  
    "percent", 0.5,  
    "speed", 500  
));
```

```
WebElement elementCard = driver.findElement(AppiumBy.xpath("//android.view.ViewGroup[@content-desc=\"card\"])[1]"));
```

*//Scroll Up*

```
((JavascriptExecutor) driver).executeScript("mobile: swipeGesture", ImmutableMap.of(  
    "elementId", ((RemoteWebElement) elementCard).getId(),  
    "direction", "up",  
    "percent", 1.0,  
    "speed", 500  
));
```



*//Scroll Down*

```
WebElement elementLogo = driver.findElement(AppiumBy.accessibilityId("WebdriverIO logo"));
```

*//Scroll Down*

```
((JavascriptExecutor) driver).executeScript("mobile: swipeGesture", ImmutableMap.of(  
    "elementId", ((RemoteWebElement) elementLogo).getId(),  
    "direction", "down",  
    "percent", 1.0,  
    "speed", 500  
));
```

## Drag to Specific Location:

```
WebElement sourceElement = driver.findElement(AppiumBy.xpath("//android.view.ViewGroup[@content-desc=\"drag-c1\"]"));

((JavascriptExecutor) driver).executeScript("mobile: dragGesture", ImmutableMap.of(
    "elementId", ((RemoteWebElement) sourceElement).getId(),
    "endX", 100,
    "endY", 100
));
```

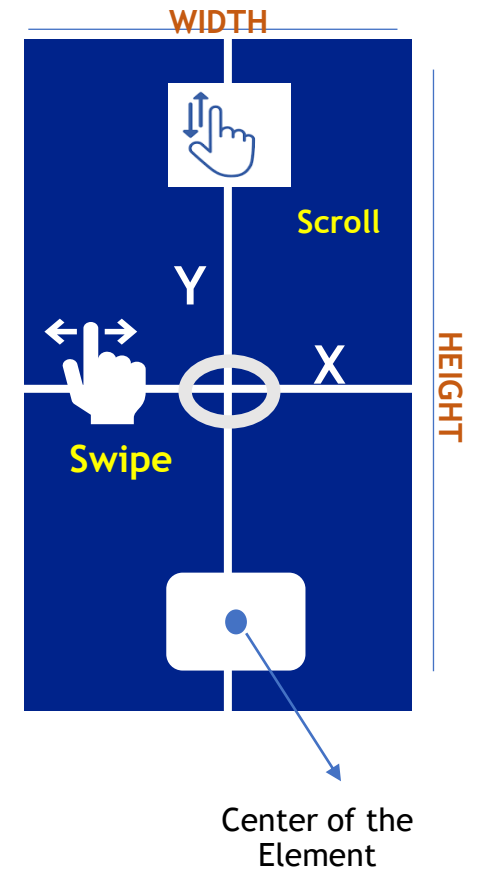
## Drag to Specific Element:

```
WebElement sourceElement = driver.findElement(AppiumBy.xpath("//android.view.ViewGroup[@content-desc=\"drag-c1\"]"));

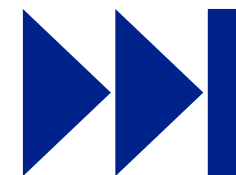
WebElement targetElement = driver.findElement(AppiumBy.xpath("//android.view.ViewGroup[@content-desc=\"drop-c1\"]"));
```

```
//Get X,Y Coordinates of Target Element
Point targetLocation = targetElement.getLocation();
//Get Width and Height of Target Element
Dimension targetSize = targetElement.getSize();
//Get Center of the Target Element
Point targetElementCenter = new Point(targetLocation.x + targetSize.getWidth()/2, targetLocation.y + targetSize.getHeight()/2);

((JavascriptExecutor) driver).executeScript("mobile: dragGesture", ImmutableMap.of(
    "elementId", ((RemoteWebElement) sourceElement).getId(),
    "endX", targetElementCenter.x,
    "endY", targetElementCenter.y
));
```



# Session Ends

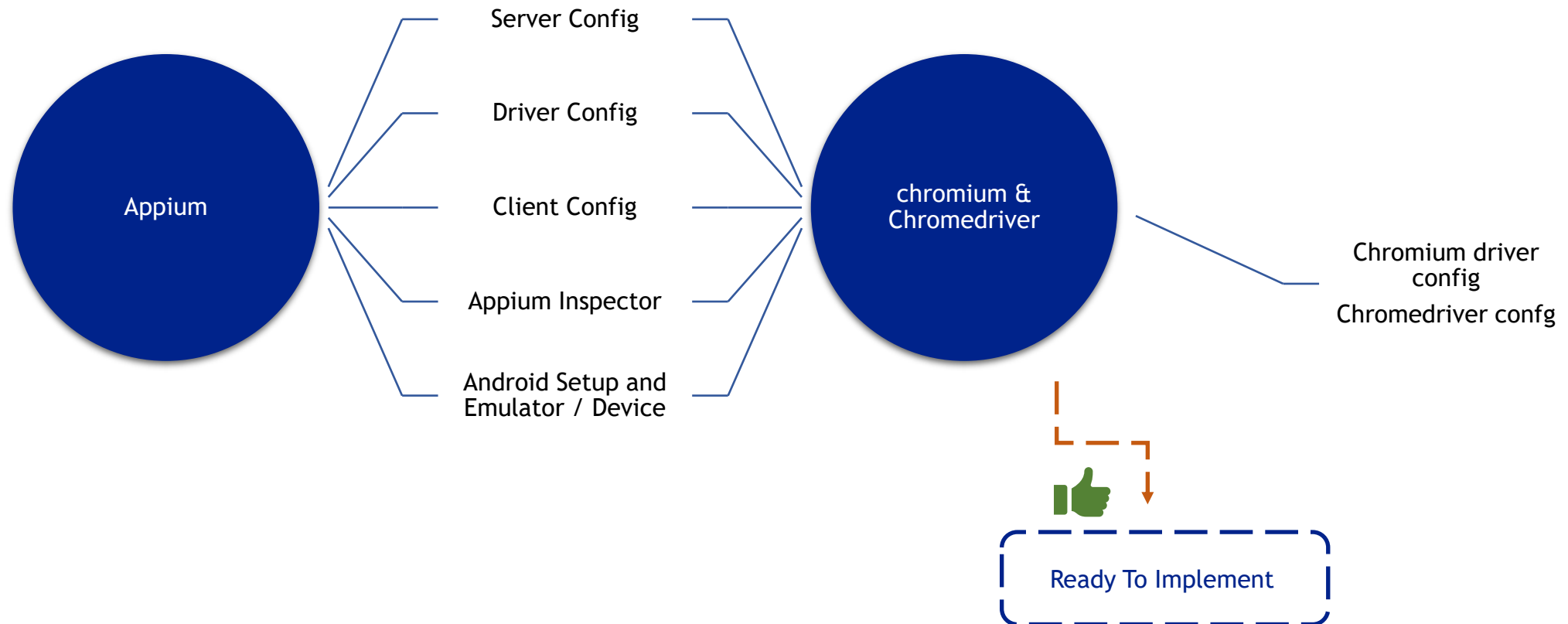


**Web Context**

**Hybrid App**



## Automation Readiness



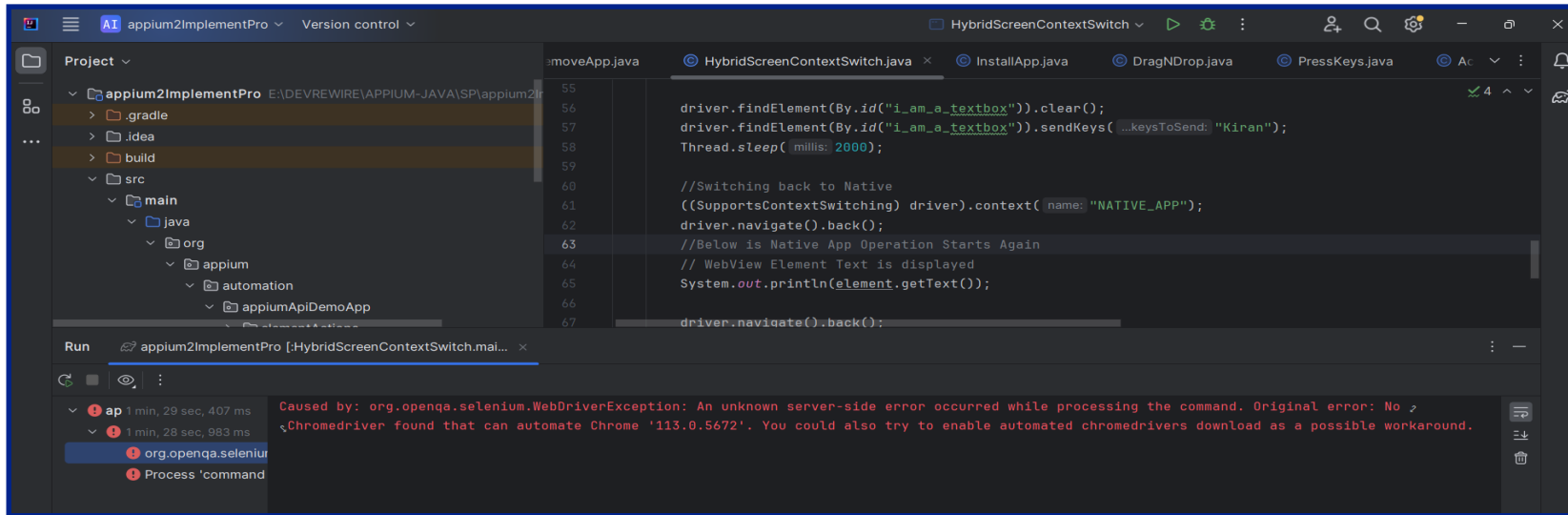


chromium driver:



Command to Install  
appium driver install chromium

```
C:\Windows\system32\cmd.e: X + v
C:\Users\kiran>appium driver install chromium
WARN Appium Appium encountered 2 warnings while validating drivers found in manifest C:\Users\kiran\.appium\node_modules
.cache\appium\extensions.yaml
WARN Appium Driver "xcuitest" has 1 potential problem:
WARN Appium - Driver "xcuitest" (package 'appium-xcuitest-driver') may be incompatible with the current version of App
ium (v2.3.0) due to its peer dependency on older Appium v^2.4.1. Please upgrade 'appium-xcuitest-driver' to v5.14.2 or n
ewer.
✓ Installing 'chromium' using NPM install spec 'appium-chromium-driver'
i Driver chromium@1.3.7 successfully installed
- automationName: Chromium
- platformNames: ["Windows", "mac", "macOS", "Linux"]
C:\Users\kiran>
```



An error will be popped up in case Chromedriver is not available in the current system under execution

Command to install Chromedriver directly by Appium.  
One Time command as below , start appium server as below

**appium server --allow-insecure chromedriver\_autodownload**

```
npm view appium-chromedriver
[ChromedriverGoogLeapisStorageClient] The entry '72.0.3626.7/chromedriver_mac64.zip' does not contain any notes. Skipping it
[ChromedriverGoogLeapisStorageClient] The entry '72.0.3626.7/chromedriver_win32.zip' does not contain any notes. Skipping it
[ChromedriverGoogLeapisStorageClient] The entry '87.0.4280.87/chromedriver_linux64.zip' does not contain any notes. Skipping it
[ChromedriverGoogLeapisStorageClient] The entry '87.0.4280.87/chromedriver_mac64.zip' does not contain any notes. Skipping it
[ChromedriverGoogLeapisStorageClient] The total count of entries in the mapping: 546
[ChromedriverChromelabsStorageClient] The total count of entries in the mapping: 2235
[Chromedriver@aedd] Got chromedrivers mapping from the storage: {
[Chromedriver@aedd]   "100.0.4896.20/chromedriver_linux64.zip": {
[Chromedriver@aedd]     "url": "https://chromedriver.storage.googleapis.com/100.0.4896.20/chromedriver_linux64.zip",
[Chromedriver@aedd]     "etag": "d7604df0340573318e05672c2d7efae",
[Chromedriver@aedd]     "version": "100.0.4896.20",
[Chromedriver@aedd]     "minBrowserVersion": "100",
[Chromedriver@aedd]     "os": {
[Chromedriver@aedd]       "name": "linux",
[Chromedriver@aedd]       "arch": "64",
[Chromedriver@aedd]       "cpu": "intel"
[Chromedriver@aedd]     }
[Chromedriver@aedd]   },
[Chromedriver@aedd]   "100.0.4896.20/chromedriver_mac64.zip": {
[Chromedriver@aedd]     "url": "https://chromedriver.storage.googleapis.com/100.0.4896.20/chromedriver_mac64.zip",
[Chromedriver@aedd]     "etag": "5..."
[ChromedriverStorageClient] Selecting chromedrivers whose minimum supported browser version matches to 113
[ChromedriverStorageClient] Got 8 items
[ChromedriverStorageClient] Will select candidate drivers versioned as '113.0.5672.24,113.0.5672.63'
[ChromedriverStorageClient] Selecting chromedrivers whose platform matches to win:intel64
[ChromedriverStorageClient] Got 2 items
[ChromedriverStorageClient] Excluding older patches if present
[ChromedriverStorageClient] Versions mapping: {
[ChromedriverStorageClient]   "113": [
[ChromedriverStorageClient]     "113.0.5672.63",
[ChromedriverStorageClient]     "113.0.5672.24"
[ChromedriverStorageClient]   ]
[ChromedriverStorageClient] }
[ChromedriverStorageClient] Got 1 driver to sync: [
[ChromedriverStorageClient]   "113.0.5672.63/chromedriver_win32.zip"
[ChromedriverStorageClient] ]
[ChromedriverStorageClient] Retrieving 'https://chromedriver.storage.googleapis.com/113.0.5672.63/chromedriver_win32.zip' to 'C:\Users\kiran\AppData\Local\Temp\2024027-13768-151291p.3nfs\0.zip'
[Support] Traversed 1 directory and 1 file in 6ms
[ChromedriverStorageClient] Moving the extracted 'chromedriver.exe' to 'C:\Users\kiran\.appium\node_modules\appium-uiautomator2-driver\node_modules\appium-chromedriver\chromedriver\win\chromedriver_win32_v113.0.5672.63.exe'
[ChromedriverStorageClient] Permissions of the file 'C:\Users\kiran\.appium\node_modules\appium-uiautomator2-driver\node_modules\appium-chromedriver\chromedriver\win\chromedriver_win32_v113.0.5672.63.exe' have been changed to 755
[ChromedriverStorageClient] Successfully synchronized 1 chromedriver
[Chromedriver@aedd] No mapping path provided
```

Chromium and Chromedriver will allow to automate Web on the Native App Container

Web Content embedded in Native App Container

- ✓ Occurrences as below
  - ❖ Starting screens of App
  - ❖ Middle screens of App
  - ❖ Last event of App

Automation Context

WEBVIEW

NATIVE\_APP

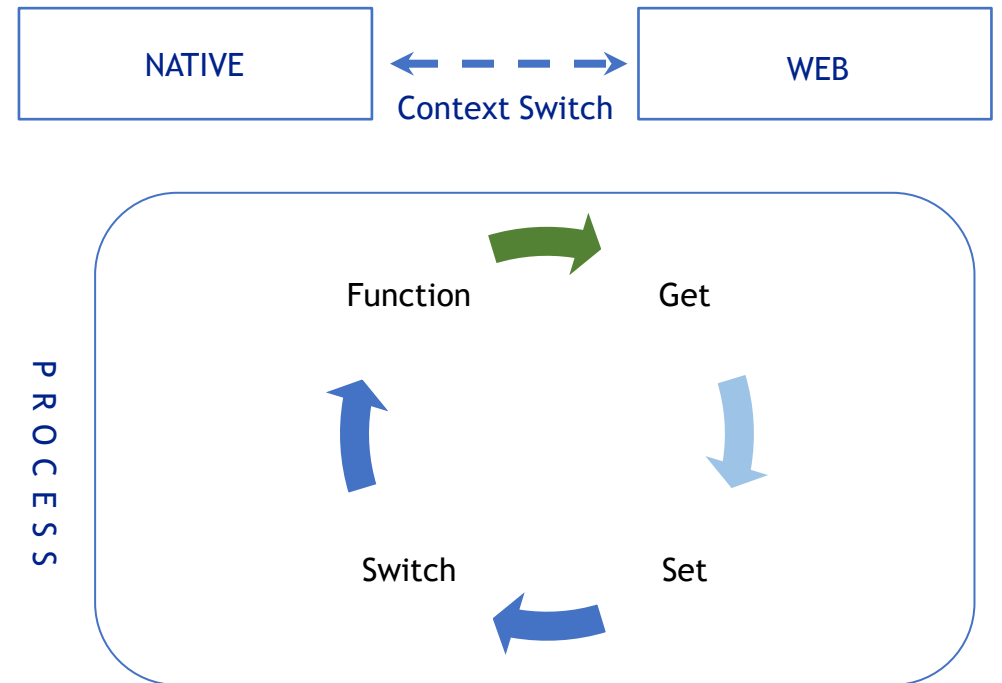


Context

WEBVIEW\_<Package Name>  
NATIVE\_APP

```
NATIVE_APP
WEBVIEW_io.appium.android.apis
```

Middle / Last Event of Web View Handling



Mind Map  
Will Map  
Web

```
//Iterate Context names and display
```

```
Set<String> contextNames = ((SupportsContextSwitching) driver).getContextHandles();
```

```
for (String contextName : contextNames) {  
    System.out.println(contextName);  
}
```

```
((SupportsContextSwitching) driver).context("WEBVIEW_io.appium.android.apis");
```

```
//Web controls operation will start
```

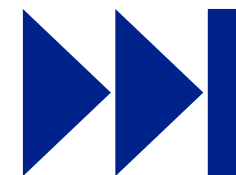


Switch between context <Web & Native>

```
((SupportsContextSwitching) driver).context("NATIVE_APP");
```

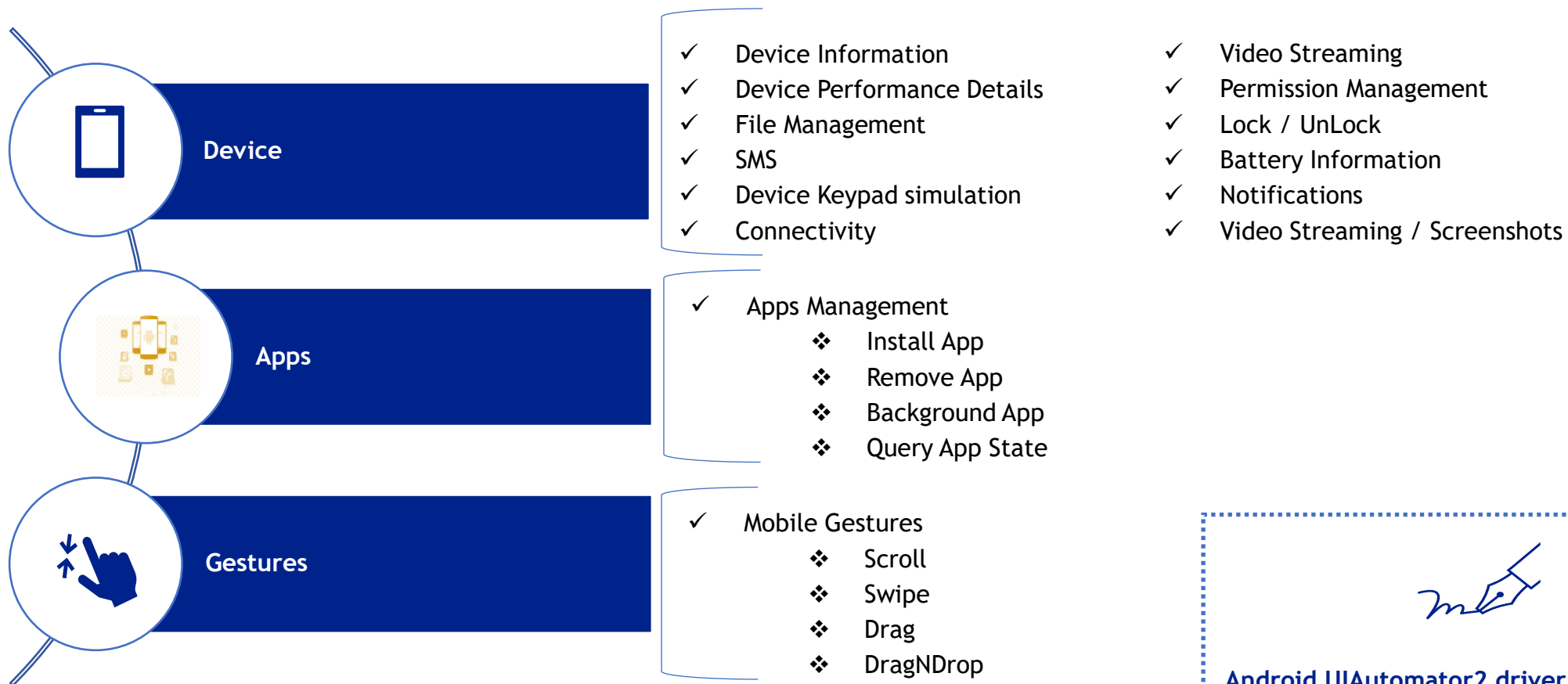
```
//Native App controls operations will resume
```

# Session Ends



# Platform Specific Extensions





Android UIAutomator2 driver provides  
Native methods to work on Device  
Operations / Apps Management





Category	Method	Description
Device	deviceInfo	A method to get Device Information such as Model , API Version etc
	getPerformanceData	Allows to get CPU Info, Battery Info , Memory Info and Network Info

## //Get Device Information

```
var deviceInfo = driver.executeScript("mobile: deviceInfo");
```

## //Get Battery Information

```
var data = driver.executeScript("mobile: getPerformanceData", ImmutableMap.of(
    "packageName", "io.appium.android.apis",
    "dataType", "batteryinfo"
));
System.out.println("battery info: " + data.toString());
```

## //Get CPU Information

```
data = driver.executeScript("mobile: getPerformanceData", ImmutableMap.of(
    "packageName", "io.appium.android.apis",
    "dataType", "cpuinfo"
));
System.out.println("cpu info: " + data.toString());
```

## //Get Memory Information

```
data = driver.executeScript("mobile: getPerformanceData", ImmutableMap.of(
    "packageName", "io.appium.android.apis",
    "dataType", "memoryinfo"
));
System.out.println("memory info: " + data.toString());
```

## //Get Network Information

```
data = driver.executeScript("mobile: getPerformanceData", ImmutableMap.of(
    "packageName", "io.appium.android.apis",
    "dataType", "networkinfo"
));
System.out.println("network info: " + data.toString());
```



Category	Method	Description
Device Keypad	pressKey	A method to simulate keypad buttons , it could be Device OS or TV OS keypad
	type	A method to simulate keypad entry on the Edit box controls

```
//Code to launch Calender Application
driver.executeScript("mobile: pressKey", ImmutableMap.of(
    "keycode", 208
));
```

```
//code to launch Contacts App
driver.executeScript("mobile: pressKey", ImmutableMap.of(
    "keycode", 207
));
```

```
//code press Android Home Key
driver.executeScript("mobile: pressKey", ImmutableMap.of(
    "keycode", 3
));
```

```
//code to enter values using Device Keypad
driver.executeScript("mobile: type", ImmutableMap.of(
    "text", "test1"
));
```

```
//Validate IsKeyboard Shown
//retrun true if visible
Boolean isKBShown = (Boolean) driver.executeScript("mobile: isKeyboardShown");
System.out.println("Keyboard Shown: " + isKBShown);
```

```
//Hide Keyboard is successfully hidden , value true is returned
//false , if it was already invisible
//throws an exception if keyboard unable to hide
Boolean isKBHidden = (Boolean) driver.executeScript("mobile: hideKeyboard");
System.out.println("Keyboard Hide Status: " + isKBHidden);
```

**URL: Keys Information** - <https://developer.android.com/reference/android/view/KeyEvent>



Category	Method	Description
Apps Management	installApp	Install specific App
	isAppInstalled	Allows to get status of app installation
	removeApp	Remove specified App
	backgroundApp	Move App to background state
	queryAppState	A method to query app state
	getCurrentActivity	A method to get current activity of the screen
	getCurrentPackage	Allows to retrieve current app package name

```
//Install WebdriverIO App
driver.executeScript("mobile: installApp", ImmutableMap.of(
  "appPath", "C:\\DEMOAPPS\\Android-NativeDemoApp-0.4.0.apk"
));
```

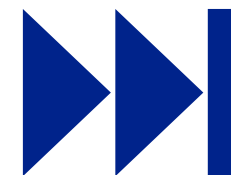
```
//Remove WebdriverIO App
Boolean isRemoveApp = (Boolean) driver.executeScript("mobile: removeApp", ImmutableMap.of(
  "appId", "com.wdiidemoapp"
));
```

## App State

```
// The app is not installed: 0
// The app is installed and is not running: 1
// The app is running in background: 3
// The app is running in foreground: 4
```



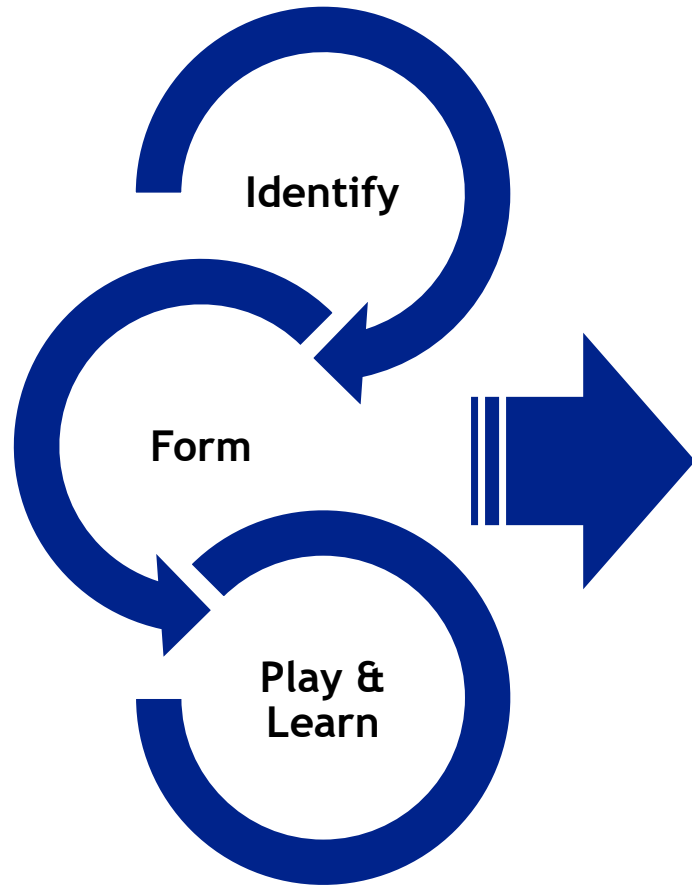
# Session Ends



# Appium Server 2.x

@Implement - Playlist - Conclusion





Resources	References
Demo Apps	<a href="#">Refer Link</a>
Demo Project	<a href="#">Refer Link</a>
Appium Server 2.x - @Implement - Guide	<a href="#">Refer Link</a>
Appium Server 2.x - @Implement - Playlist	<a href="#">Refer Link</a>



[YouTube Channel: @reWireByAutomation](#)

[Medium Blogging: @reWireByAutomation](#)



*THANK YOU*



reWire  
fueled by decode