

Native Driver

Native App UI Automation with WebDriver API

Presented at the 2011 Selenium Conference
San Francisco

Matt DeVore
Tomohiro Kaizu
Dezheng Xu
Daigo Hamura

What are we doing?

- Supply a UI automation solution for native applications
- ... through an extended WebDriver API
- Started with Android, other platforms in works:
 - iOS - Tomohiro Kaizu
 - Windows - Daigo Hamura

Why WebDriver?

1. Many operations are universal between all UIs

WebDriver UI Automation Atoms

Matching:

- clear
- click
- findElement(By.id)
- getLocation
- getSize
- isDisplayed
- isEnabled
- isSelected
- sendKeys
- setSelected

Non-matching:

- executeScript
- fire (fire a specific, synthesized event)
- getAttribute
- submit (a convenience method for submitting a form)
- toggle

Why WebDriver?

2. `WebElement.findElement` is universal - all UIs are trees

Web has DOM (elements)

Windows has HWNDs

Android has Views - can be viewed with HierarchyViewer tool

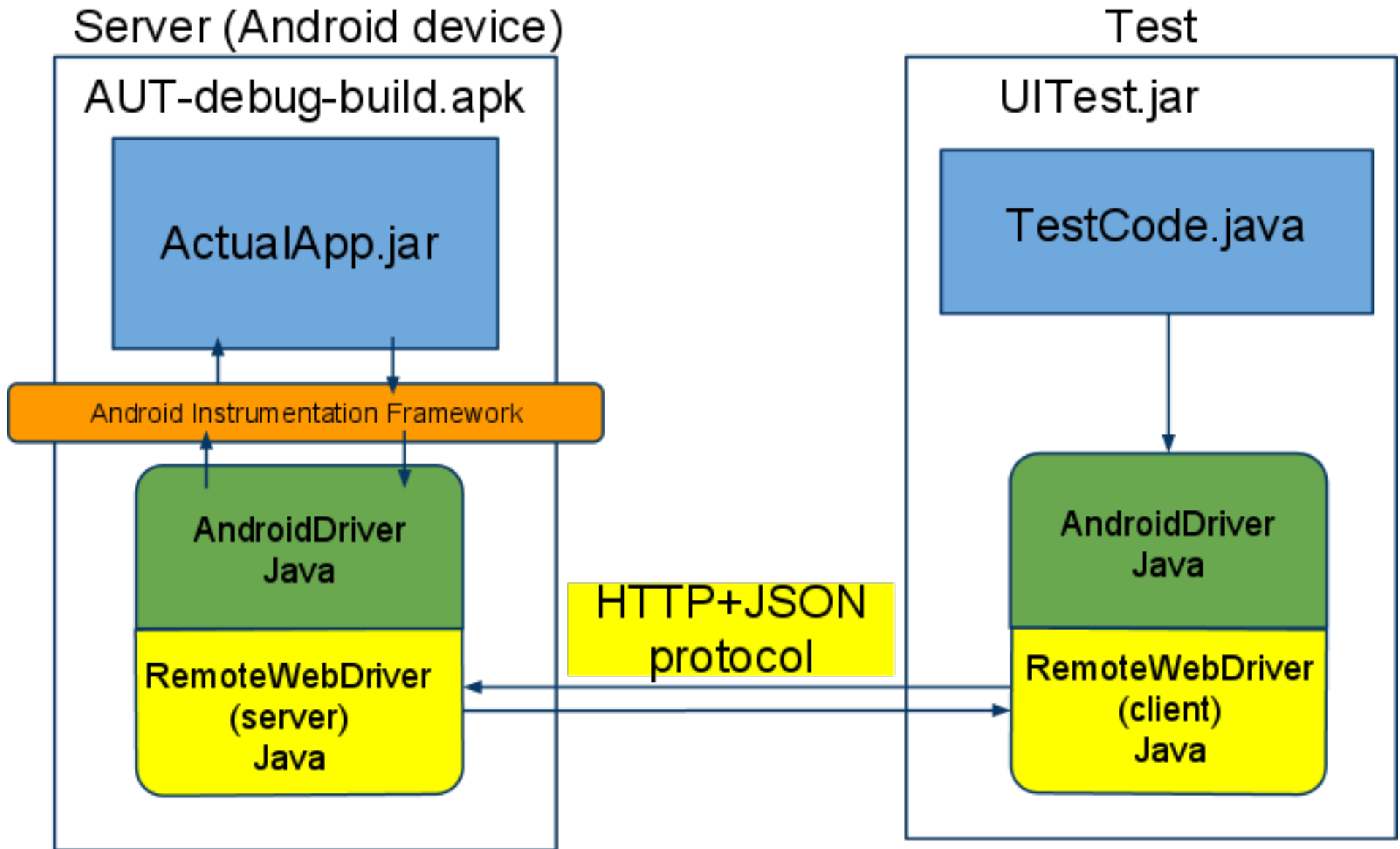
Each UI element kind has a set of attributes that can be searched on.

Why WebDriver?

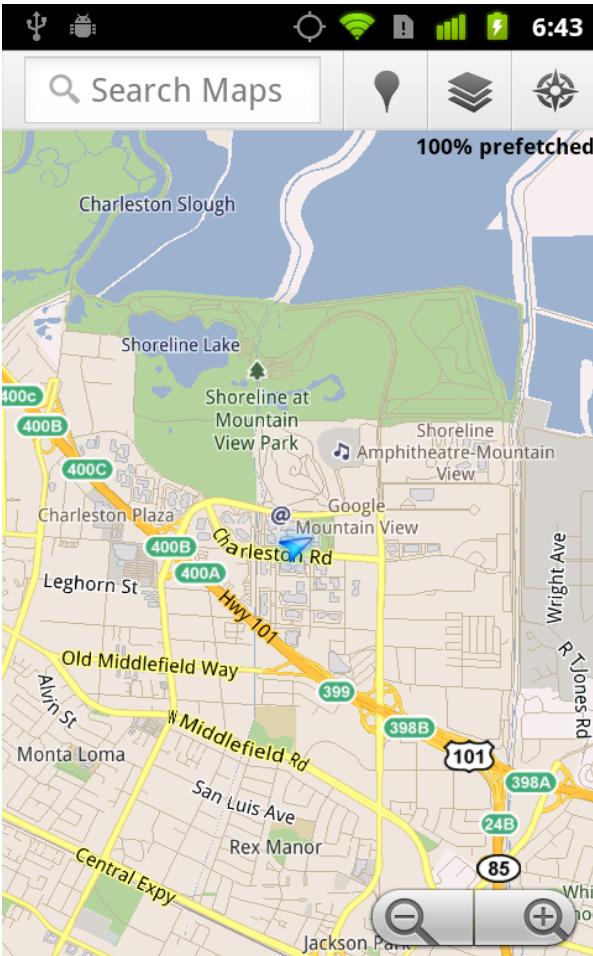
3. Extensible API

- WebDriver API extensions, such as Rotatable, RenderedWebElement, are implemented by a subset of drivers.
- Creating a new API for *every* platform is cumbersome - instead:
 - Emphasize the similarities with the core WebDriver API
 - Supply simple platform-specific extensions as needed.
- Facilitate piece-wise code re-usage

Android NativeDriver architecture

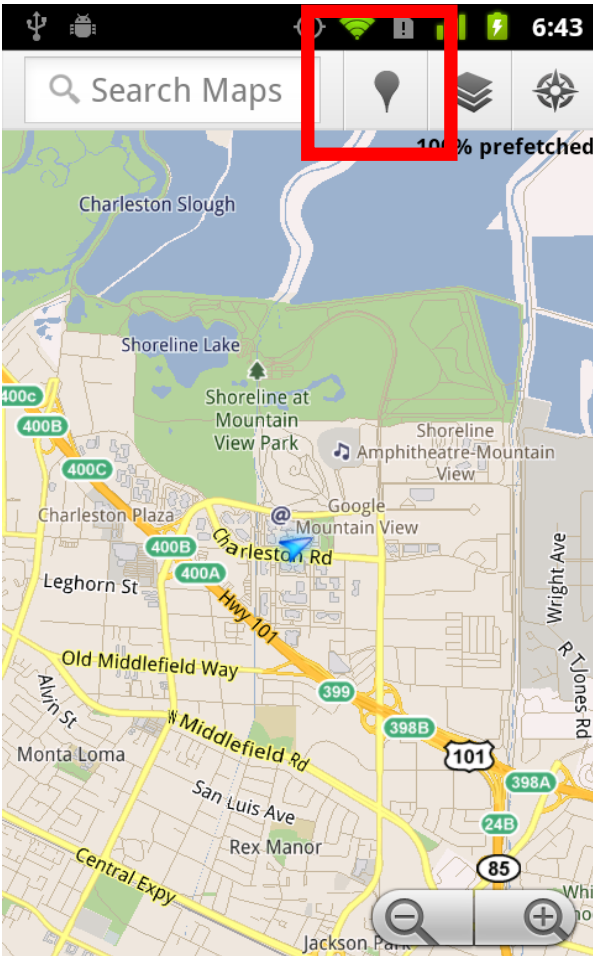


Demo on Google Map for Android



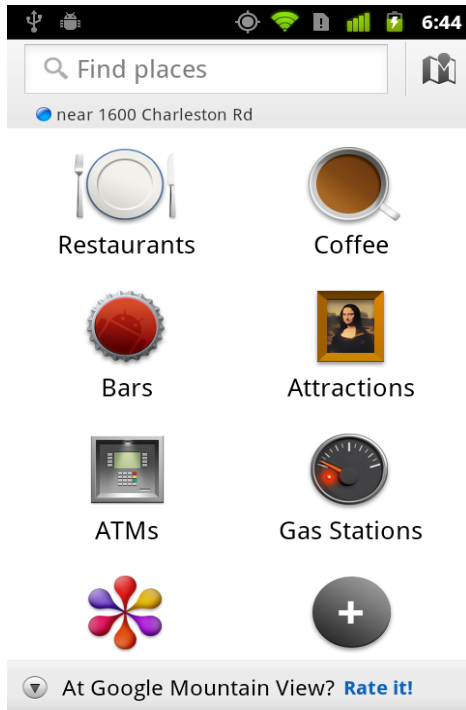
```
driver.startActivity(  
    "com.google.android.maps.MapsActivity");  
WebElement btn  
    = driver.findElement(By.id("btn_header_places"));  
btn.click();  
...  
driver.navigate().back();  
...  
driver.rotate(ScreenOrientation.LANDSCAPE);  
...  
driver.rotate(ScreenOrientation.PORTRAIT);  
...  
WebElement searchBox  
    = driver.findElement(By.id("search_box"));  
assertEquals("Search Maps", searchBox.getText());  
searchBox.sendKeys("shibuya station, tokyo, japan\n");  
...
```

Demo on Google Map for Android

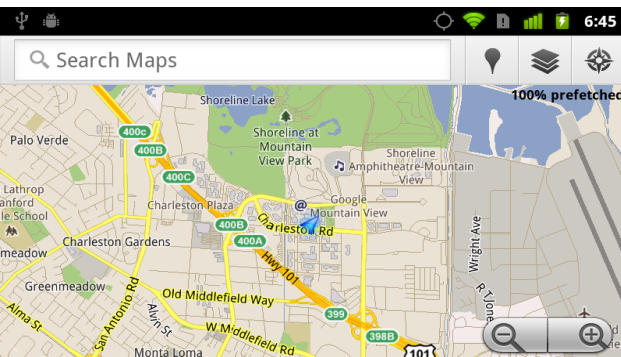


```
driver.startActivity(  
    "com.google.android.maps.MapActivity");  
WebElement btn  
    = driver.findElement(By.id("btn_header_places"));  
btn.click();  
...  
driver.navigate().back();  
...  
driver.rotate(ScreenOrientation.LANDSCAPE);  
...  
driver.rotate(ScreenOrientation.PORTRAIT);  
...  
WebElement searchBox  
    = driver.findElement(By.id("search_box"));  
assertEquals("Search Maps", searchBox.getText());  
searchBox.sendKeys("shibuya station, tokyo, japan\n");  
...
```

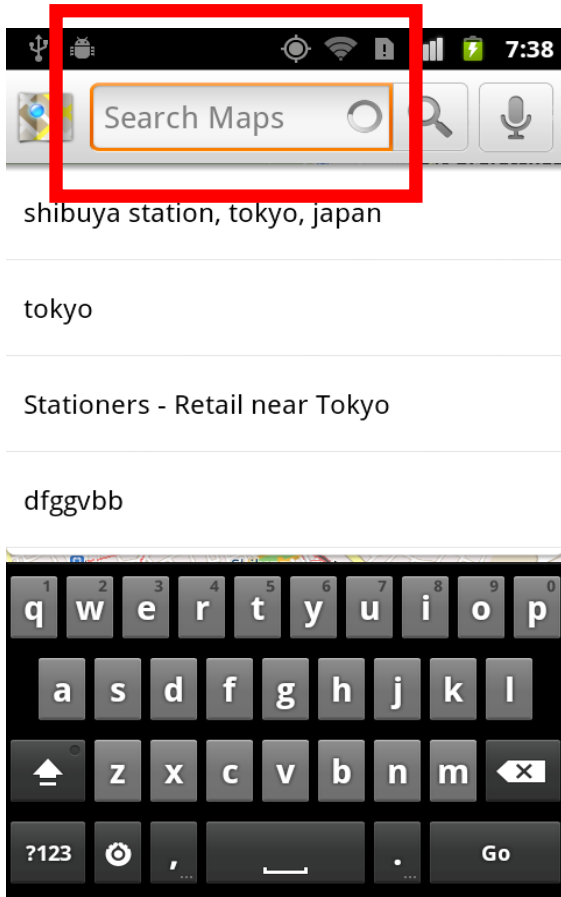

Demo on Google Map for Android



```
driver.startActivity(  
    "com.google.android.maps.MapsActivity");  
WebElement btn  
    = driver.findElement(By.id("btn_header_places"));  
btn.click();  
...  
driver.navigate().back();  
...  
driver.rotate(ScreenOrientation.LANDSCAPE);  
...  
driver.rotate(ScreenOrientation.PORTRAIT);  
...  
WebElement searchBox  
    = driver.findElement(By.id("search_box"));  
assertEquals("Search Maps", searchBox.getText());  
searchBox.sendKeys("shibuya station, tokyo, japan\n");  
...
```



Demo on Google Map for Android

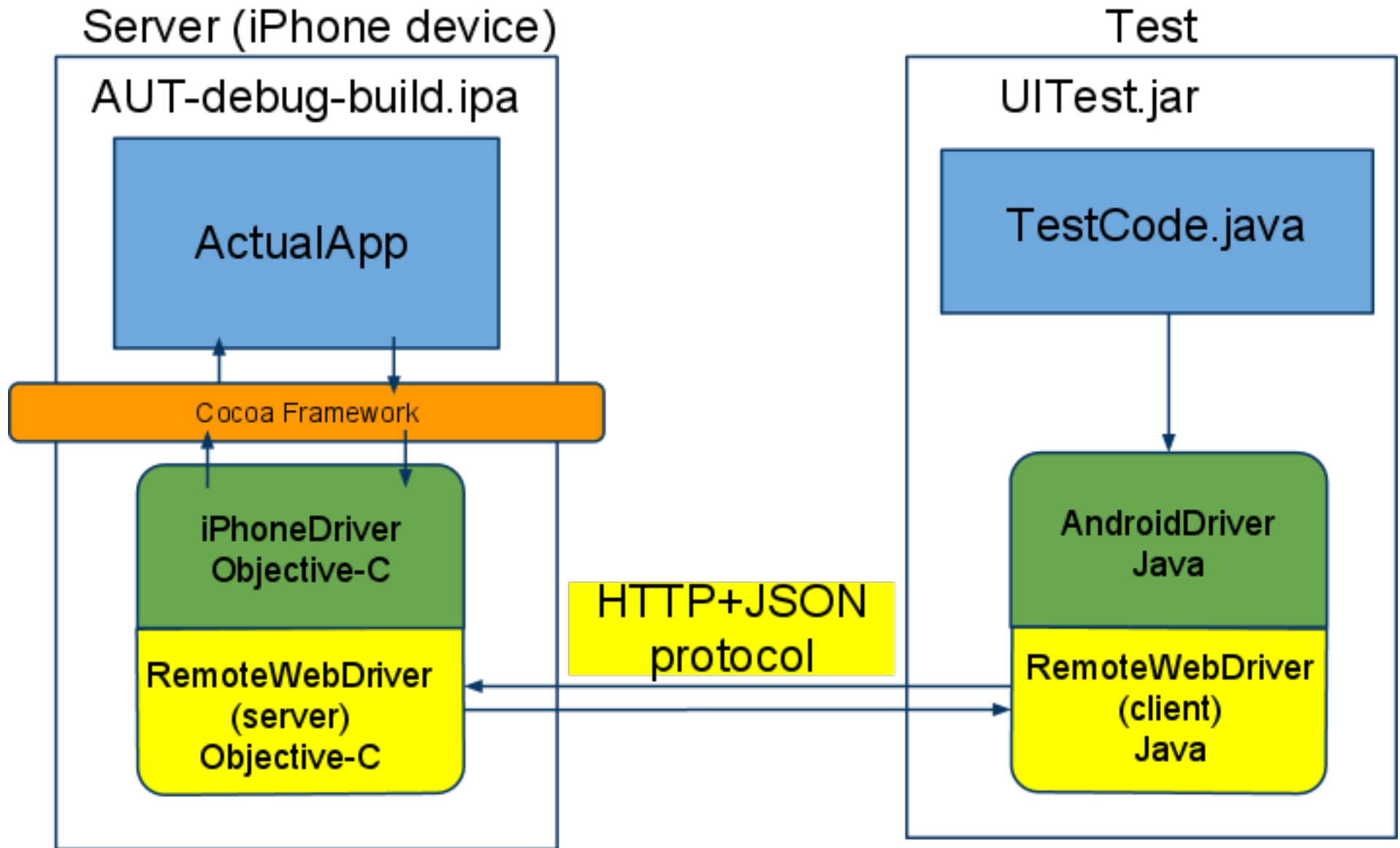


```
driver.startActivity(  
    "com.google.android.maps.MapActivity");  
WebElement btn  
    = driver.findElement(By.id("btn_header_places"));  
btn.click();  
  
...  
driver.navigate().back();  
  
...  
driver.rotate(ScreenOrientation.LANDSCAPE);  
  
...  
driver.rotate(ScreenOrientation.PORTRAIT);  
  
...  
WebElement searchBox  
    = driver.findElement(By.id("search_box"));  
assertEquals("Search Maps", searchBox.getText());  
searchBox.sendKeys("shibuya station, tokyo, japan\n");  
  
...
```

iPhone NativeDriver

- iPhone NativeDriver project has just started
- iOS 4 has a standard UI Automation tool, but...
 - There's no command line interface to start test automatically
 - Testers must study new automation API
- Using iPhone NativeDriver, you can drive iPhone apps in the same way as Android apps and Web apps!

iPhone NativeDriver architecture



Next steps

- Exploring ways to drive black-box components (GLSurfaceView, user custom views)
- Test web content embedded in native applications (WebView) with classic WebDriver semantics
 - Relies on ongoing work to integrate WebDriver into Android natively
 - Make native and web "conceptual switch" seamless.