

 無料電子ブック

学習

appium

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#appium

.....	1
<b>1: appium</b> .....	<b>2</b>
.....	2
.....	2
Examples.....	3
.....	3
.....	3
Appium.....	4
Appium.....	4
Appium for Android.....	5
<b>2: Appium</b> .....	<b>8</b>
.....	8
Examples.....	8
.....	8
<b>3: Java</b> .....	<b>11</b>
.....	11
Examples.....	11
Android Play.....	11
PlayStoreAutomation.java.....	11
pom.xml.....	12
.....	14

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [appium](#)

It is an unofficial and free appium ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official appium.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# 1: appiumをいめる

Appiumは、シミュレータiOS、FirefoxOS、エミュレータAndroid、およびリアルデバイスiOS、Android、FirefoxOSでテストされた、ネイティブ、ハイブリッド、モバイルのWebアプリケーションのオープンソースのクロスプラットフォームテストツールです。

## なぜAppium

1. すべてのプラットフォームでのオートメーションAPIをするため、アプリをコンパイルするはありません。
2. すべてのプラットフォームでのオートメーションAPIをするため、アプリをコンパイルするはありません。 [Java](#)、 [Objective-C](#)、 [Node.js](#)、 [コールバック](#)、 [ジェネレータ](#)の、 [PHP](#)、 [Python](#)、 [Ruby](#)、 [C](#)、 [Clojure](#)、 [Perl](#)などの[WebDriver](#)をして、あなたのきなツールでテストをくことができます。 [Selenium WebDriver API](#)とのクライアントライブラリをしています。
3. のテストフレームワークをできます。

WebDriverプロトコルへのは、デファクトスタンダードとなったテストのために、でオープンなプロトコルにけていることをします。のスタックにをロックしないでください。

AppiumをしないAppleのUIAutomationライブラリをするは、JavaScriptをしてのみテストをすることができ、Instrumentsアプリケーションをしてテストをすることができます。に、GoogleのUiAutomatorでは、Javaでのみテストをくことができます。Appiumは、のクロスプラットフォームネイティブモバイルのをきます。

い

Appiumは々なネイティブフレームワークをし、Seleniumの[WebDriver JSONワイヤプロトコル](#)にづいたAPIをします。

AppiumはiOS 10のバージョンのAppleのUIAutomationライブラリをします。これは[Dan Cuellar](#)の[iOS Auto](#)のについています。UIAutomationライブラリにより、すべてのiOS 10およびのバージョンは、XCUITestフレームワークによってされます。

Androidのサポートでは、しいプラットフォームのUiAutomatorフレームワークといAndroidプラットフォームの[Selendroid](#)がされます。

FirefoxOSサポートは、WebDriverとがあり、Geckoベースのプラットフォームをするためにされるオートメーションドライバである[Marionette](#)をしています。

## バージョン

バージョン	
1.6.3	2016-12-12
1.6.2	2016-12-02
1.6.1	2016-11-24
1.6.0	2016-10-10
1.5.3	201667
1.5.2	2016-04-20
1.5.1	2016-03-29
1.5.0	2016-02-26
1.4.16	2015-11-20
1.4.15	2015-11-18
1.4.14	2015-11-06
1.4.13	2015-09-30
1.4.11	2015-09-16
1.4.10	2015-08-07
1.4.8	2015-07-16
1.4.7	2015-07-02
1.4.6	2015-06-19
1.4.3	2015-06-09
1.4.1	2015-05-21
1.4.0	2015-05-09
1.3.7	2015-03-25
1.3.6	2014-12-01

## Examples

インストールまたはセットアップ

するデバイスタイプのをし、Appiumをするにそれらがインストールされていることをしてください

## iOS

- Mac OS X 10.10、10.11.1
- XCode >= 6.0、7.1.1を
- Apple Developer Tools iPhoneシミュレータSDK、コマンドラインツール
- [iOSテストのためにをするのドキュメントを](#)ずんでください

## Android

- [Android SDK API = 17](#)には18/19が
- Appiumは、OS X、Linux、WindowsでAndroidをサポートしています。なるOSでテストするためのをしくするためのにってください。
  - [linux](#)
  - [OSX](#)
  -

## FirefoxOSの

- [Firefox OSシミュレータ](#)

---

# Appiumのインストール

## Node.jsをしたグローバルインストール

```
$ npm install -g appium
$ appium
```

## Githubのマスターブランチからのローカルインストール

```
$ git clone git@github.com:appium/appium.git
$ cd appium
$ npm install
$ node .
```

## MacまたはWindowsのAppの

- [Appiumアプリケーションをダウンロードする](#)
- それをします

## Appiumのテストをく

Appium [ドキュメント](#)のされたバージョンは、のコードサンプルをするとともに[ここ](#)にあります。

## Appium for Android プラットフォームのとサンプルテストの

- APIレベル17のAndroidSDKをダウンロード
- Node.js <https://nodejs.org/>
- Appiumソフトウェア <http://appium.io/>
- Selenium jars <http://www.seleniumhq.org/download/>
- Appium jar  
<https://search.maven.org/#search%7Cga%7C1%7Cg%3Aio.appium%20a%3Ajava-client>
- テストするがあるアプリケーションの.apkファイル
  
- Eclipseが[www.eclipse.org/downloads/](http://www.eclipse.org/downloads/)からダウンロードされていることをしてください
- javaがインストールされていますjdkとjreの
- android sdkがインストールされています
- Java、Android SDK、プラットフォーム、プラットフォームツールのPathがされていることをしてください。

Windowsのパスをする「マイコンピュータ」をクリックします。「プロパティ」パネルの「アドバンスドシステム」のシステム->タイプパス->「パス」ダブルクリックシステムにJAVA jdkのパスをしてから;をし、 android sdk;アンドロイドプラットフォームのパス;パスアンドロイドプラットフォームのツール->OKをクリックします。

- Eclipseプラグインがインストールされていることをする

AndroidEclipseプラグインのインストールuit Eclipseをし、ヘルプ/ソフトウェアのインストールをします。●の[]をクリックします。されるリポジトリのダイアログで、の「ADT Plugin」とロケーションのURLをします。 <https://dl-ssl.google.com/android/eclipse/>「OK」をクリックしますにがあるプラグインをしているは、「https」ではなく「URL」で「http」をしてくださいセキュリティのからhttpsがされます。

- ANDROID\_HOMEがされていることをしてください。

ANDROID\_HOMEをするのはとおりですuitトップパネルのPreferences-> Preferences->パネルのAndroidをダブルクリックします。Androidで、SDKのをコピーします。「マイコンピュータ」をクリックします。「プロパティ」パネルの「アドバンスド・システム」のユーザー->->ので、ANDROID\_HOME、パスを - EclipseからコピーしたSDKのを -> OKをクリックに、システム->->、ANDROID\_HOME、パス -> EclipseからコピーされたSDKのを -> OKをクリック

- Android Virtual Device Managerをできることをしてください。>のデバイスがあればそれをクリックするか、カスタマイズされたでしいデバイスをします。>>ウィンドウのパネルで[スタート]をクリックします。ちげ

Appiumの

- node.js " <http://nodejs.org/> "をインストールします。
- のからコマンドラインからAppiumをしますAppiumフォルダ→node\_modules  
→appium→bin→shift +クリック→くコマンドプロンプト→タイプノードappium→enter

infoAppium v1.3.4へようこそREV c8c79a85fbd6870cd6fc3d66d038a115ebe22efeinfoAppium  
REST httpインターフェイスリスナ0.0.0.0:4723でinfoConsole LogLeveldebug infoAppium REST  
httpインターフェイスリスナ0.0.0.0で 4723infoコンソールLogLevelデバッグ

EclipseでAppiumをするためのプログラムをく package appium.com;

```
import java.net.MalformedURLException; import java.net.URL;
```

```
import org.openqa.selenium.remote.CapabilityType; import  
org.openqa.selenium.remote.DesiredCapabilities; import  
org.openqa.selenium.remote.RemoteWebDriver;
```

パブリッククラスAppiumLaunch {public static void mainString args []MalformedURLExceptionを  
スローする{RemoteWebDriverドライバ。 DesiredCapabilities capabilities =しい  
DesiredCapabilities;

```
capabilities.setCapability("platformName", "Android");  
capabilities.setCapability("deviceName", "");  
  
capabilities.setCapability("version", "4.4.2");  
capabilities.setCapability("device ID", "");  
capabilities.setCapability("app-package", "");  
capabilities.setCapability(CapabilityType.BROWSER_NAME, "");  
  
capabilities.setCapability("app-activity", "");  
capabilities.setCapability("takesScreenshot", true);  
  
capabilities.setCapability("app", "C:/Users/.....apk");  
  
driver=new RemoteWebDriver( new URL("http://127.0.0.1:4723/wd/hub"), capabilities);  
System.out.println("app is launched on the device");  
  
}
```

}

- システムのapkファイルのパスがしいことをしてください
- システムのapkファイルへのパスがプログラムでしいことをしてください。しいパッケージ  
とapkファイルをコンパイルすることでつかるアクティビティをしてください。 apkファイ  
ルをコンパイルするには、 [http //www.decompileandroid.com/](http://www.decompileandroid.com/)にアクセスしてください。

approid for Androidをする

1. まず、コマンドプロンプトでappiumサーバーをするか、 appium.exeファイルをします。
2. デバイスがされ、 adbadbデバイスにされているかどうかをする

3. Eclipseでプログラムをします。プログラムがされ、デバイスにインストールされた.apkファイルがアプリをします。

オンラインでappiumをいめるをむ <https://riptutorial.com/ja/appium/topic/5122/appiumをいめる>

## 2: Appiumでのテスト

き

セレニウムGRIDコンセプトをしたappiumでの。ステップバイステップのプロセスをつけてください。

### Examples

ステップバイステップのプロセス

GRIDをしたAppiumによるテストがいていたをします。 AppiumでSelenium Gridをする

1. Seleniumグリッドのセットアップローカルファイルシステムのselenium standalone server jarをダウンロードします。をき、jarファイルをしたディレクトリにし、のコマンドをします。

```
java -jar selenium-server-standalone-2.53.3.jar -role hub
Open http://localhost:4444/grid/console and you should be able to see GRID console in your browser.
```

2. Appium Nodesのここでは、jsonファイルをするがあります。 2つのデバイスでし、2つのなるjsonファイルをするをします。ここに1つのjsonファイルがあります。{"capabilities" {"applicationName "" ONEPLUS A3003 "、" browserName "" ONEPLUS A3003 "、" platformName "" ANDROID "、" maxInstances "1}}、"configuration" {"cleanUpCycle"2000、"timeout"30000、"proxy" "org.openqa.grid.selenium.proxy.DefaultRemoteProxy"、"host" "127.0.0.1"、"port" 4723、"maxSession"1、"register"true、"registerCycle"5000、"hubPort"4444、"hubHost" "あなたのIPアドレス"}}のファイルをjasonFile1.jsonとしてします。ここでapplicationNameは ->あなたのモバイル -> ->について ->モデルここでhubHostはあなたのIPアドレスになりますここであなたはデフォルトのcmdのとしてくがあることにしてください

```
appium --nodeconfig C/richa/jasonfile1.json -p 4723 -bp 4724 -U xxxx
```

ijsonファイルのパスをするがあります。 iiポートは4723です。 iiiブートストラップポートは4724です。 iv-Uたとえば、xxxx

->あなたのモバイル -> ->ステータス ->シリアルのようにデバイスIDをつけることができます "adbデバイス"をし、このデバイスIDをするともできます。

に、1つのデバイスでSelenium Gridをします。

もう、2のjsonファイルをする、 appiumがします。 2のjsonファイルはのとおりです。

```
{ "applicationName": "Lenovo K50a40", "browserName": "Lenovo K50a40", "platformName": "ANDROID", "maxInstances": 1, "configuration": { "cleanUpCycle": 2000, "timeout": 30000, "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy", "host": "127.0.0.1", "port": 4730, "maxSession": 1, "register": true, "registerCycle": 5000, "hubPort": 4444, "hubHost": "あなたのIPアドレス" } }のファイルをjasonFile2.jsonとします
```

Lenovo Mobileで2のノードをします。 `appium --nodeconfig C:/richa/jasonFile2.json -p 4730 -bp 4731 -U xxxx`

Selenium Gridはのようになります

3テストをするTestNGパラレルメソッドをします。

->デバイスのはあなたがにしたudidになることにしてください。コマンドプロンプトでadbデバイスをすることでできます。

4.

SearchHotelTestCase.Javaをのようになります。 `package com.trivago.TestCases;`

```
import java.net.MalformedURLException; import java.net.URL; import java.util.concurrent.TimeUnit;
```

```
import org.openqa.selenium.remote.DesiredCapabilities; import org.openqa.selenium.remote.RemoteWebDriver; import org.testng.annotations.BeforeMethod; import org.testng.annotations.Parameters; import org.testng.annotations.Test;
```

```
import com.trivago.pages.LocaleSelectionPage; import com.trivago.pages.SearchLocation; import com.trivago.pages.SplashScreenPage;
```

```
import io.appium.java_client.MobileElement; import io.appium.java_client.android.AndroidDriver;
```

```
public class SearchHotelTestCase {プライベートAndroidDriverドライバ。
```

```
@Parameters({"deviceName_", "platformVersion_", "applicationName_"})@BeforeMethod public void beforeMethodString deviceName_, String platformVersion_, String applicationName_ MalformedURLException、 InterruptedException {
```

```
DesiredCapabilities capabilities = しいDesiredCapabilities; capabilities.setCapability "deviceName"、 deviceName_; capabilities.setCapability "platformVersion"、 platformVersion_; capabilities.setCapability "platformName"、 "Android"; capabilities.setCapability "applicationName"、 applicationName_; capabilities.setCapability "app"、 "/Users/richa.b.shrivastava/Downloads/com.trivago_2017-04-28.apk"; capabilities.setCapability "appPackage"、 "com.trivago"; capabilities.setCapability "appActivity"、 "com.trivago.activities.SplashActivity";
```

```
URL url = しいURL "http //0.0.0.0:4723/wd/hub/ "; System.out.println "webdriverの"; ドライバ= しいAndroidDriverURL、 ; System.out.println "webdriverの"; 。にする10、 TimeUnit.SECONDS;
```

```
Thread.sleep(4000); }
```

```
@Test public void SearchHotel(){//ページクラスのオブジェクトを作るLocaleSelectionPage  
localeSelectionPage = new LocaleSelectionPageDriver(); SplashScreenPage splashScreenPage =  
新しいSplashScreenPage ドライバ; SearchLocation searchLocation = 新しいSearchLocation ドライバ
```

```
。
```

```
//ページクラス localeSelectionPage.selectLocale のメソッドを呼び出します。  
splashScreenPage.clickSplashSearchText; searchLocation.inputSearchText "Paris";  
searchLocation.selectSuggestions "エッフェル、パリ";
```

```
}
```

```
}
```

オンラインでAppiumでのテストをむ <https://riptutorial.com/ja/appium/topic/10016/appiumでのテスト>

## 3: Java クライアント

JavaクライアントAPI

Javaクライアントソースコード

### Examples

Android Play ストアのリアルデバイス

ファイル

- pom.xml
- src / test / java / PlayStoreAutomation.java

コマンド

```
mvn test -Dtest = PlayStoreAutomation
```

### PlayStoreAutomation.java

```
import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.android.AndroidKeyCode;
import io.appium.java_client.MobileElement;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.By;
import java.util.concurrent.TimeUnit;
import java.net.URL;

public class PlayStoreAutomation {
    public static AndroidDriver<MobileElement> driver;

    @BeforeClass
    public static void setUp() throws Exception {
        DesiredCapabilities capabilities = new DesiredCapabilities();
        capabilities.setCapability("platformName", "Android");
        capabilities.setCapability("deviceName", "Android Device");
        capabilities.setCapability("appPackage", "com.android.vending");
        capabilities.setCapability("appActivity",
            "com.google.android.finsky.activities.MainActivity");

        driver = new AndroidDriver<MobileElement>(new URL("http://localhost:4723/wd/hub"),
            capabilities);
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    }

    @AfterClass
    public static void tearDown() {
        driver.quit();
    }
}
```

```

}

@Test
public void testPlayStore() throws Exception {
    driver.findElement(By.id("com.android.vending:id/text_container")).sendKeys("Google");
    driver.pressKeyCode(AndroidKeyCode.ENTER);

    // First item in the search result by Xpath

driver.findElement(By.xpath("//android.support.v7.widget.RecyclerView[1]/android.widget.LinearLayout[1]

    // Confirm element found
    driver.findElement(By.xpath("//android.widget.TextView[@text='Google']"));
}
}

```

## pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.testner.appium</groupId>
    <artifactId>tester-tests</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>io.appium</groupId>
            <artifactId>java-client</artifactId>
            <version>4.0.0</version>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>2.10</version>
                <configuration>
                    <reportsDirectory>${project.build.directory}/reports</reportsDirectory>
                </configuration>
            </plugin>
        </plugins>
    </build>

```

```
</plugins>  
</build>  
</project>
```

オンラインでJavaクライアントをむ <https://riptutorial.com/ja/appium/topic/6195/javaクライアント>

## クレジット

S. No		Contributors
1	appiumをいめる	<a href="#">BenJi</a> , <a href="#">Community</a> , <a href="#">Domestus</a> , <a href="#">mrtuovinen</a> , <a href="#">Priya</a> , <a href="#">Richa Shrivastava</a>
2	Appiumでのテスト	<a href="#">Richa Shrivastava</a>
3	Javaクライアント	<a href="#">Domestus</a>