



Бесплатная электронная книга

УЧУСЬ

appium

Free unaffiliated eBook created from
Stack Overflow contributors.

#appium

| | |
|---------------------------------|-----------|
| | 1 |
| 1: appium | 2 |
| | 2 |
| | 3 |
| Examples..... | 4 |
| | 4 |
| | 4 |
| Appium..... | 4 |
| Appium..... | 5 |
| Appium Android | 5 |
| 2: Java | 8 |
| | 8 |
| Examples..... | 8 |
| Android Play (Real device)..... | 8 |
| PlayStoreAutomation.java..... | 8 |
| pom.xml..... | 9 |
| 3: Appium | 11 |
| | 11 |
| Examples..... | 11 |
| | 11 |
| | 14 |

Около

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

глава 1: Начало работы с appium

замечания

[Appium](#) - это инструмент для кросс-платформенной автоматизации тестов для собственных, гибридных и мобильных веб-приложений, проверенный на тренажерах (iOS, FirefoxOS), эмуляторах (Android) и реальных устройствах (iOS, Android, FirefoxOS).

Почему Апиум?

1. Вам не нужно перекомпилировать приложение или изменить его каким-либо образом из-за использования стандартных API автоматизации на всех платформах.
2. Вам не нужно перекомпилировать приложение или изменить его каким-либо образом из-за использования стандартных API автоматизации на всех платформах. Вы можете писать тесты с помощью ваших любимых инструментов для разработчиков, используя любой совместимый с [WebDriver](#) язык, такой как [Java](#) , [Objective-C](#) , JavaScript с Node.js (в [обещаниях](#), [обратном вызове](#) или [генераторах](#)), PHP, [Python](#) , [Ruby](#) , [C #](#) , Clojure или Perl с API Selenium WebDriver и языковыми клиентскими библиотеками.
3. Вы можете использовать любую платформу тестирования.

Инвестирование в протокол WebDriver означает, что вы делаете ставку на единый, бесплатный и открытый протокол для тестирования, который стал стандартом де-факто. Не запирайте себя в запатентованном стеке.

Если вы используете библиотеку UIAutomation от Apple без Appium, вы можете писать тесты только с помощью JavaScript, и вы можете запускать тесты только через приложение «Инструменты». Аналогично, с помощью UiAutomator от Google вы можете писать тесты только на Java. Appium открывает возможность реальной кросс-платформенной мобильной автоматизации.

Как это устроено

Appium управляет различными собственными системами автоматизации и предоставляет API на основе [протокола Selenium WebDriver JSON](#) .

Appium управляет библиотекой UIAutomation от Apple для версий до iOS 10, которая основана на работе [Дэна Куэллара](#) в iOS Auto. С устаревшей библиотекой UIAutomation все iOS 10 и будущая версия управляются инфраструктурой XCUITest.

Поддержка Android использует инфраструктуру UiAutomator для новых платформ и [Selendroid](#) для старых платформ Android.

Поддержка FirefoxOS использует [Marionette](#) , драйвер автоматизации, совместимый с

WebDriver, и используется для автоматизации платформ на базе Gecko.

Версии

| Версия | Дата выхода |
|--------|-------------|
| 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 | 2016-06-07 |
| 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

Установка или настройка

Предварительно требования

Проверьте требования к каждому типу устройства, которое вы хотите автоматизировать, и убедитесь, что они установлены, прежде чем пытаться использовать Appium!

Требования к iOS

- Mac OS X 10.10 или выше, рекомендуется 10.11.1
- XCode >= 6.0, 7.1.1 рекомендуется
- Apple Developer Tools (iPhone simulator SDK, инструменты командной строки)
- [Убедитесь, что вы прочитали документацию по настройке себя для тестирования iOS!](#)

Требования к Android

- [Android SDK API](#) >= 17 (дополнительные функции требуют 18/19)
- Appium поддерживает Android на OS X, Linux и Windows. Обязательно следуйте инструкциям по настройке среды для тестирования на разных ОС:
 - [линукс](#)
 - [OSX](#)
 - [окна](#)

Требования FirefoxOS

- [Симулятор Firefox OS](#)

Установка Appium

Глобальная установка с использованием Node.js

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

Локальная установка с ведущей ветки Github

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

Использование приложения для Mac или Windows

- [Загрузите приложение Appium](#)
- Запустить его!

Написание тестов для Appium

Форматированную версию [документов](#) Appium можно найти [здесь](#), с возможностью выбора языка примера кода из верхнего правого угла.

Запуск Appium для платформы Android и создание тестового теста

Настройка среды:

- Загрузите sdk для Android уровня API 17 или более
- Node.js (<https://nodejs.org/>)
- Программное обеспечение Appium (<http://appium.io/>)
- Селен-банки (<http://www.seleniumhq.org/download/>)
- Appium jar (<https://search.maven.org/#search%7Cga%7C1%7Cg%3Aio.appium%20a%3Ajava-client>)
- .арк-файла приложения, которое необходимо протестировать

Предпосылки:

- убедитесь, что Eclipse загружен с сайта www.eclipse.org/downloads/
- java установлен (как jdk, так и jre)
- Установлен Android-sdk
- Убедитесь, что установлена переменная среды (путь) для Java, Android SDK, платформы и платформы.

Шаги по установке Путь на ОС Windows: Щелкните правой кнопкой мыши «Мой компьютер». «Свойства» На левой панели «Предварительные системные настройки» Выбрать переменные среды Системные переменные-> Тип Путь-> «Путь» двойной щелчок Введите путь к JAVA jdk в вашей системе, а затем (;) затем путь к вашему android sdk (;) путь к вашему пути платформы Android (;) к вашим инструментам платформы Android -> Нажмите «ОК».

- Убедитесь, что подключен плагин Eclipse

Шаги по установке Eclipse Plug-in для Android: Запустите Eclipse, затем выберите «Справка»> «Установить новое программное обеспечение». Нажмите «Добавить» в верхнем правом углу. В появившемся диалоговом окне «Добавить репозиторий» введите « ADT Plugin» для имени и следующего URL-адреса для местоположения: <https://dl-ssl.google.com/android/eclipse/> Нажмите «ОК» (если у вас возникли проблемы с приобретением плагин, попробуйте использовать «http» в URL-адресе местоположения

вместо «https» (https предпочтительнее из соображений безопасности).

- Убедитесь, что установлена переменная ANDROID_HOME.

Шаги для установки переменной ANDROID_HOME: Перейдите в Eclipse-> Window на верхней панели-> Настройки-> Дважды щелкните Android на левой панели В настройках Android скопируйте местоположение SDK Щелкните правой кнопкой мыши «Мой компьютер». «Свойства» На левой панели «Предварительные настройки системы» Выберите «Переменные среды» В верхней части «Пользовательские переменные» -> «Выбрать новое» - «Имя переменной», «Введите ANDROID_HOME», «Переменная путь» -> «Введите скопированное местоположение SDK из Eclipse-> Нажмите « OK » Затем системные переменные-> Выберите новое-> Имя переменной, введите ANDROID_HOME, Переменная Path-> Введите скопированное местоположение SDK из Eclipse-> Нажмите OK
Выход

- Убедитесь, что Android Virtual Device Manager можно запустить. Eclipse-> Окно на верхней панели-> Android Virtual Device Manager-> Нажмите на существующее виртуальное устройство, если оно существует / Создайте новый с настраиваемыми конфигурациями .-> Нажмите «Пуск» на правой панели окна .-> запуск

Запуск Appium:

- Установите node.js (" <http://nodejs.org/> ").
- Запустите Appium из командной строки из нижерасположенного места: Goto Appium folder node_modules appiumbinshift + right clickopen command prompttype node appium enter

Следующее должно отображаться: информация: Добро пожаловать в Appium v1.3.4 (REV c8c79a85fbd6870cd6fc3d66d038a115ebe22efe) info: Appium REST http-интерфейс прослушивателя запущен на 0.0.0.0:4723 информация: Console LogLevel: debug info: Appium REST http-интерфейс прослушивателя запущен на 0.0.0.0: 4723info: Console LogLevel: debug

Напишите программу для запуска Appium в Eclipse: 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;
```

```
public class AppiumLaunch {public static void main (String args [])  
выдает  
MalformedURLException {RemoteWebDriver driver; Возможности DesiredCapabilities = new  
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");

}
```

```
}
```

- Убедитесь, что путь к файлу арк в системе верен
- Убедитесь, что путь к файлу арк в вашей системе верен в программе. Используйте правильный пакет и активность, которые можно найти, декомпилировав файл арк. Для декомпиляции файла арк перейдите по [адресу http://www.decompileandroid.com](http://www.decompileandroid.com) .

Шаги для запуска appium для Android:

1. Сначала запустите appium server в командной строке или запустите файл appium.exe.
2. Проверьте, подключено ли устройство и отображается ли оно в adb: устройства adb
3. Выполните программу на Eclipse. Программа будет выполнена, и файл .apk, который был установлен на устройстве, запустит приложение.

Прочитайте Начало работы с appium онлайн: <https://riptutorial.com/ru/appium/topic/5122/начало-работы-с-appium>

глава 2: Клиент Java

замечания

[API Java Client](#)

[Исходный код Java-клиента](#)

Examples

Автоматизация магазина Android Play (Real device)

Структура файла:

- pom.xml
- SRC / тест / Java / PlayStoreAutomation.java

Запуск команды:

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

```

```
        <configuration>
            <reportsDirectory>${project.build.directory}/reports</reportsDirectory>
        </configuration>
    </plugin>
</plugins>
</build>
</project>
```

Прочитайте Клиент Java онлайн: <https://riptutorial.com/ru/appium/topic/6195/клиент-java>

глава 3: Параллельное тестирование в Appium

Вступление

Параллельное выполнение в appium с использованием концепции сериового GRID. Пожалуйста, найдите шаг за шагом.

Examples

Шаг за шагом

Параллельное тестирование с помощью Appium с использованием GRID: я опишу способ, который работал для меня. Создание селеновой сетки с помощью Appium

1. Настройка селеновой сетки. Загрузите автономный серверный сервер selenium в локальной файловой системе. Откройте терминал и перейдите в каталог, где вы разместили файл jar, и выполните следующую команду:

```
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. Настройка Appium Nodes Здесь вам нужно создать json-файлы. Предположим, вы хотите запустить на двух устройствах, затем создать два разных файла json. Вот один json-файл, у меня есть: {"возможности": [{"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»: «ваш IP-адрес»}} сохраните указанный выше файл как jsonFile1.json Здесь applicationName будет -> Your Mobile-> settings-> about phone-> Номер модели Здесь hubHost будет вашим IP-адресом. Обратите внимание, что вам нужно перейти в CMD по умолчанию, а затем выполнить команду ниже

```
appium --nodeconfig C:/richa/jsonfile1.json -p 4723 -bp 4724 -U xxxx
```

- i) Обратите внимание, что вам нужно предоставить абсолютную path json-файла, расположенного в ii) порт как 4723 iii) Порт Bootstrap как 4724 iv) -U, например, я указал как xxxx

вы можете найти идентификатор устройства как -> Your Mobile-> settings-> status-> Serial number. Вы также можете использовать «adb device» и проверить этот идентификатор

устройства.

Затем он создаст селеновую сетку с одним устройством.

Теперь снова запустите второй json-файл, и вы получите appium. Вот второй файл json:

```
{«возможности»: [{«имя_приложения»: «Lenovo K50a40», «имя_буфера»: «Lenovo K50a40», «platformName»: «ANDROID», «maxInstances»: 1}], «configuration»: {«cleanUpCycle»: 2000, «таймаут»: 30000, «прокси»: «org.openqa.grid.selenium.proxy.DefaultRemoteProxy», «host»: «127.0.0.1», «port»: 4730, «maxSession»: 1, «register ": true," registerCycle ": 5000," hubPort ": 4444," hubHost ":" ваш IP-адрес "}}
```

 сохраните вышеуказанный файл как `jsonFile2.json`

Запустите второй узел с помощью мобильного телефона Lenovo. `appium --nodeconfig C:/richa/jsonFile2.json -p 4730 -bp 4731 -U xxxx`

Селеновая сетка будет выглядеть так:

3) Создайте методы параллельного выполнения TestNG для запуска теста.

-> Обратите внимание, что значение имени устройства будет удид, который вы предоставили ранее. Вы можете получить его, запустив adb-устройства в командной строке.

4.

Теперь создайте `SearchHotelTestCase.java`, как показано ниже: `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;
```

открытый класс `SearchHotelTestCase` {частный драйвер `AndroidDriver`;

```
@Parameters ({"deviceName _", "platformVersion _", "applicationName_"}) @BeforeMethod public void beforeMethod (String deviceName_, String platformVersion_, String applicationName_) throws MalformedURLException, InterruptedException {
```

```
Возможности DesiredCapabilities = new DesiredCapabilities (); capability.setCapability ("deviceName", deviceName_); capability.setCapability ("platformVersion", platformVersion_); capability.setCapability ("platformName", "Android"); capability.setCapability ("applicationName",
```

```
applicationName_); capability.setCapability ("app",
"/Users/richa.b.shrivastava/Downloads/com.trivago_2017-04-28.apk"); capability.setCapability
("appPackage", "com.trivago"); capability.setCapability ("appActivity",
"com.trivago.activities.SplashActivity");
```

```
URL url = новый URL (" http://0.0.0.0:4723/wd/hub/ "); System.out.println («до веб-диска»); driver
= новый AndroidDriver (url, возможности); System.out.println («после webdriver»);
driver.manage (). timeouts (). implicitlyWait (10, TimeUnit.SECONDS); Thread.sleep (4000); }
```

```
@Test public void SearchHotel () { // Создание объектов класса страницы LocaleSelectionPage
localeSelectionPage = новый LocaleSelectionPage (драйвер); SplashScreenPage
splashScreenPage = новый SplashScreenPage (драйвер); SearchLocation searchLocation =
new SearchLocation (драйвер);
```

```
// Вызов методов класса страницы localeSelectionPage.selectLocale ();
splashScreenPage.clickSplashSearchText (); searchLocation.inputSearchText ( "Париж");
searchLocation.selectSuggestions ("Эйфелева башня, Париж");
```

```
}
```

```
}
```

Прочитайте [Параллельное тестирование в Appium онлайн](https://riptutorial.com/ru/appium/topic/10016/параллельное-тестирование-в-appium):

<https://riptutorial.com/ru/appium/topic/10016/параллельное-тестирование-в-appium>

кредиты

| S. No | Главы | Contributors |
|-------|------------------------------------|---|
| 1 | Начало работы с appium | BenJi , Community , Domestus , mrtuovinen , Priya , Richa Shrivastava |
| 2 | Клиент Java | Domestus |
| 3 | Параллельное тестирование в Appium | Richa Shrivastava |