## Download nodejs

* Download nodejs msi from official website
* Install it
* Validate the installation by checking the environment path(which will be auto set)
* Validate the installation by giving the command *node -v*

## Install Appium via npm

* Provide the command *npm i -g appium*
* Validate the installation by providing the command *appium –version*, you should get the version if appium is installed successfully.
* Give the command *where appium* to get the path of appium in the system

## Start appium from cmd

* Give the command *appium* to start appium server
* Press control + Z to stop the server

## Installing appium-doctor

Appium doctor gives information on what installations are available and what not.

Use the command *npm install -g appium-doctor* to install appium-doctor

Use the command *appium-doctor help* , to get what options we can use

## Install android components

Check what are installed related to android using the command *appium-doctor --android.*

Install Java (JDK)  
Set JAVA\_HOME and PATH for Java

### To find list of drivers installed in the machine

Provide the command *appium driver list ,* to know what drivers are installed and what not.

You can install the driver using the command,

* Android: appium driver install uiautomator2
* IOS: appium driver install xcuitest
* Check: appium driver list
* To check for Updates: appium driver list –updates
* To install updates: appium driver list update uiautomator2/xcuitest

### Setup Android SDK and SDK tools

Download android studio for official site

Create a project / set the SDK manager

* Set the environment paths like below
* Android home : Environmental variables --- > create ANDROID\_HOME with value as …..\Sdk
* Append PATH: %ANDROID\_HOME%\build-tools and %ANDROID\_HOME%\platform-tools

Now check whether the android paths are set well and what else is pending using the command *appium-doctor –android*

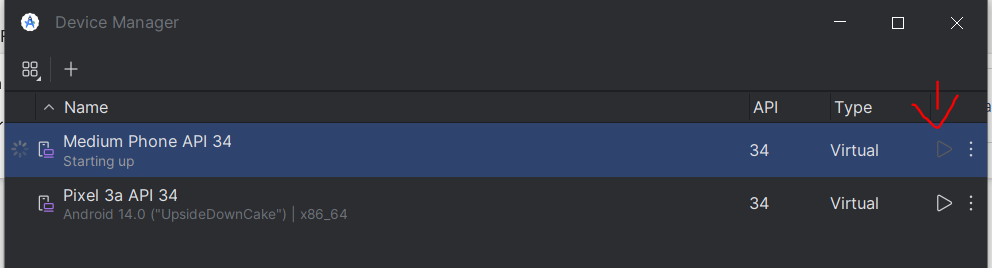
### Setting up Android virtual devices

Start android studio

Open the virtual device manager

Select the mobile, android version, configuration and create the device

Start the device, by clicking on the start ok on the device info



### Setting up and using appium inspector

You can install desktop app or use the website url

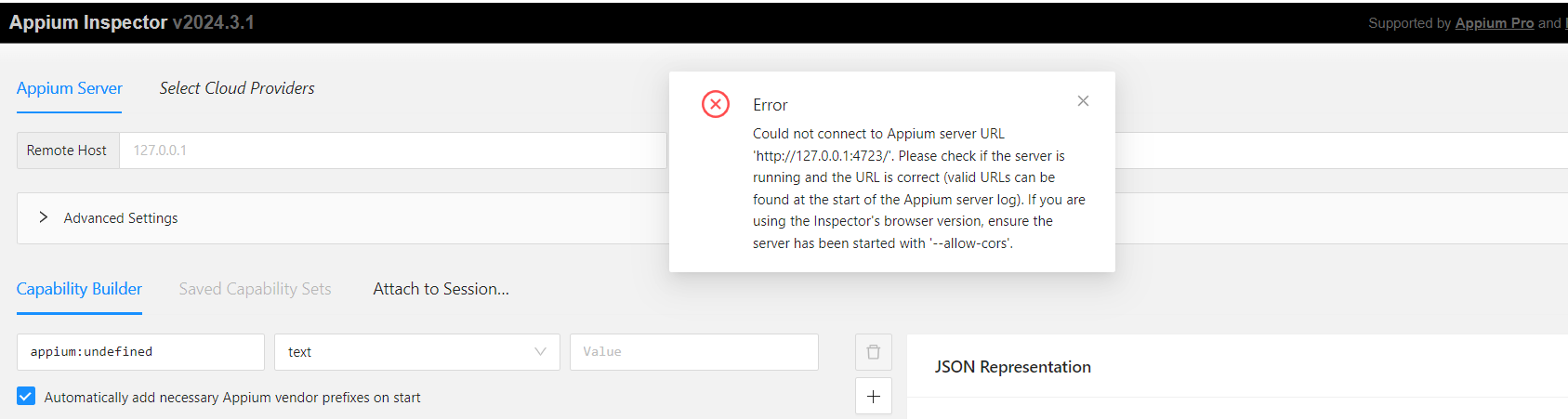
#### Desktop App

Download appium inspector from github – official site, choose the correct exe file and download

#### Website url

You can get it from git , looks like <https://inspector.appiumpro.com/>

* Start appium from cmd
* You will get the host name and remote port
* Provide them in the appium inspector
* If you get the below error then stop the appium and give the below command



*Control + c*

*appium --allow-cors*

cors stands for cross origin resource sharing

Now, provide desired capabilities in the appium inspector

Provide the below JSON in JSON Representation

{

"appium:automationName": "UiAutomator2",

"appium:platformName": "Android",

"appium:platformVersion": "11",

"appium:deviceName": "4b316ae9",

"appium:app": "C:\\....\\..\\...\\APIDemos.apk"

}

deviceName - can get using command *adb devices*

platformName - can get using command *appium driver list*

platformVersion - can get using command *adb shell getprop ro.build.version.release*

If you have multiple devices connected first get the device id using adb devices then run command

*adb -s DEVICE\_ID shell getprop ro.build.version.release*

app - provide the location of APK file (for testing download it from git -

*adb shell getprop ro.build.version.sdk* - This command will provide the API level (e.g., “30” for Android 11, “29” for Android 10, etc.)

## Connect to virtual device

* Make sure appium is running, emulator is on and appium inspector is connected
* Provide the below info and click on start session in appium server

JSON in JSON Representation

{

"appium:automationName": "UiAutomator2",

"appium:platformName": "Android",

"appium:platformVersion": "11",

"appium:deviceName": "4b316ae9",

"appium:app": "C:\\....\\..\\...\\APIDemos.apk"

}

## Browserstack Devices with Appium Inspector

Use Appium Inspector desktop version

Step 1 - On Appium Inspector top menu click Select Cloud Providers and select Browserstack

Step 2 - Create account on Browserstack and get your username and accesskey

Step 3 - On Browserstack upload your application or use the demo application

Step 4 - If needed can use capabilities generator https://www.browserstack.com/app-automate/capabilities

Step 5 - Add the browserstack username, accesskey and desired capabilities in Appium Inspector

{

"appium:automationName": "UiAutomator2",

"platformName": "Android",

"appium:platformVersion": "9.0",

"appium:deviceName": "Google Pixel 3",

"appium:app": "bs://c700ce60cf13ae8ed97705a55b8e022f13c5827c"

}

Ref: https://appium.github.io/appium-inspector/2024.2/

Browserstack Capabilities Generator - https://www.browserstack.com/app-automate/capabilities

SauceLabs Capabilities Generator - https://saucelabs.com/products/platform-configurator

LambdaTest Capabilities Generator - https://www.lambdatest.com/capabilities-generator/

## appPackage and appActivity

appPackage and appActivity can be obtained in two ways,

### from command prompt

adb shell dumpsys window displays | grep -E ‘mCurrentFocus’

adb shell "dumpsys window windows | grep -E 'mCurrentFocus|mFocusedApp'”

### from APK file app

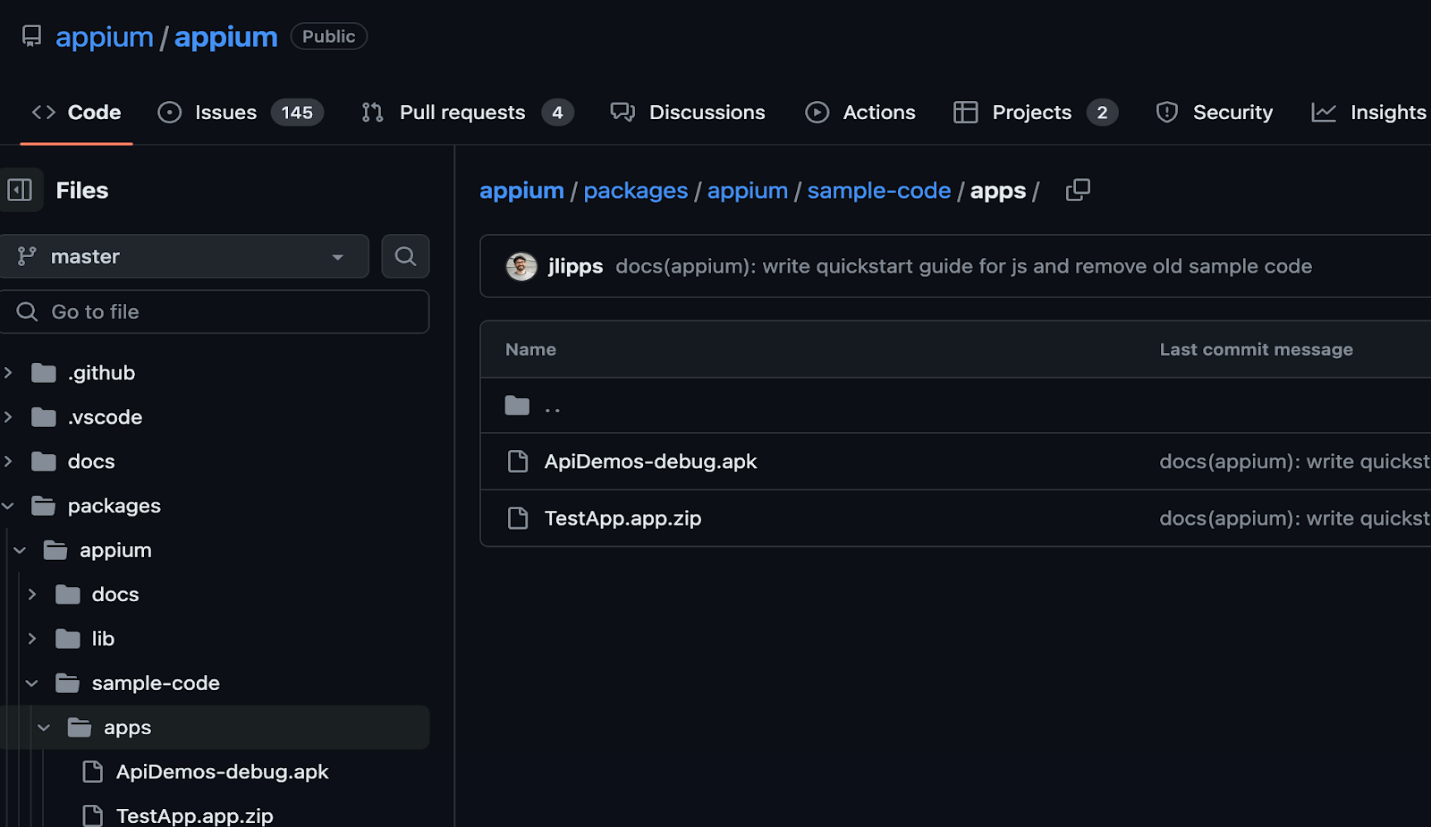
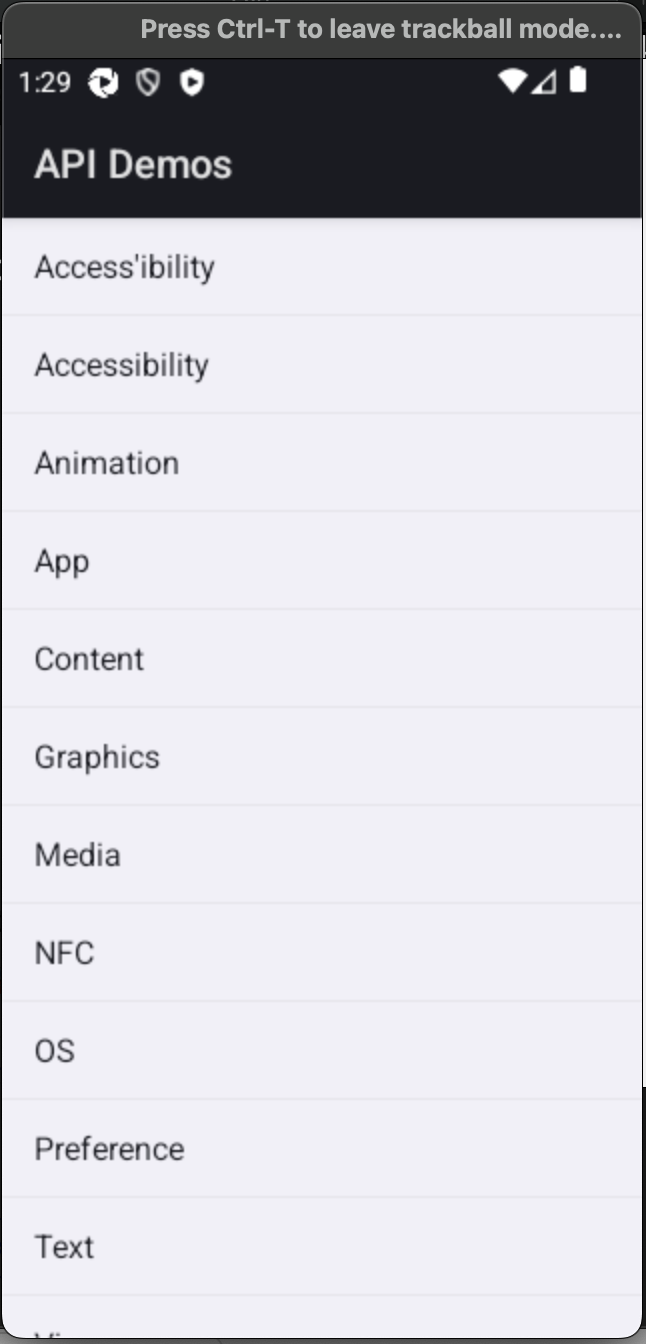
Can also install APK Info app on your device to get appPackage and appActivity of the apps installed on the device

## Sample APK’s

**Here are some sample APK files that we can use for learning test automation**

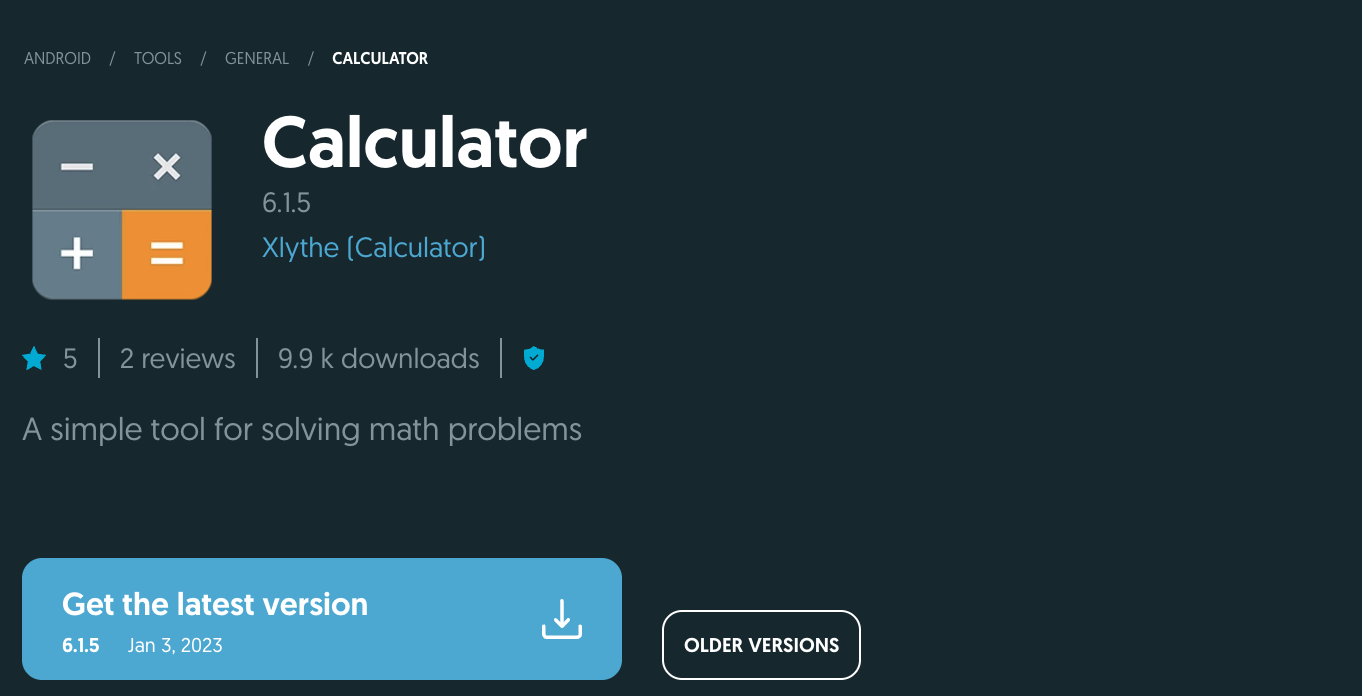
Can get some sample APK files from appium

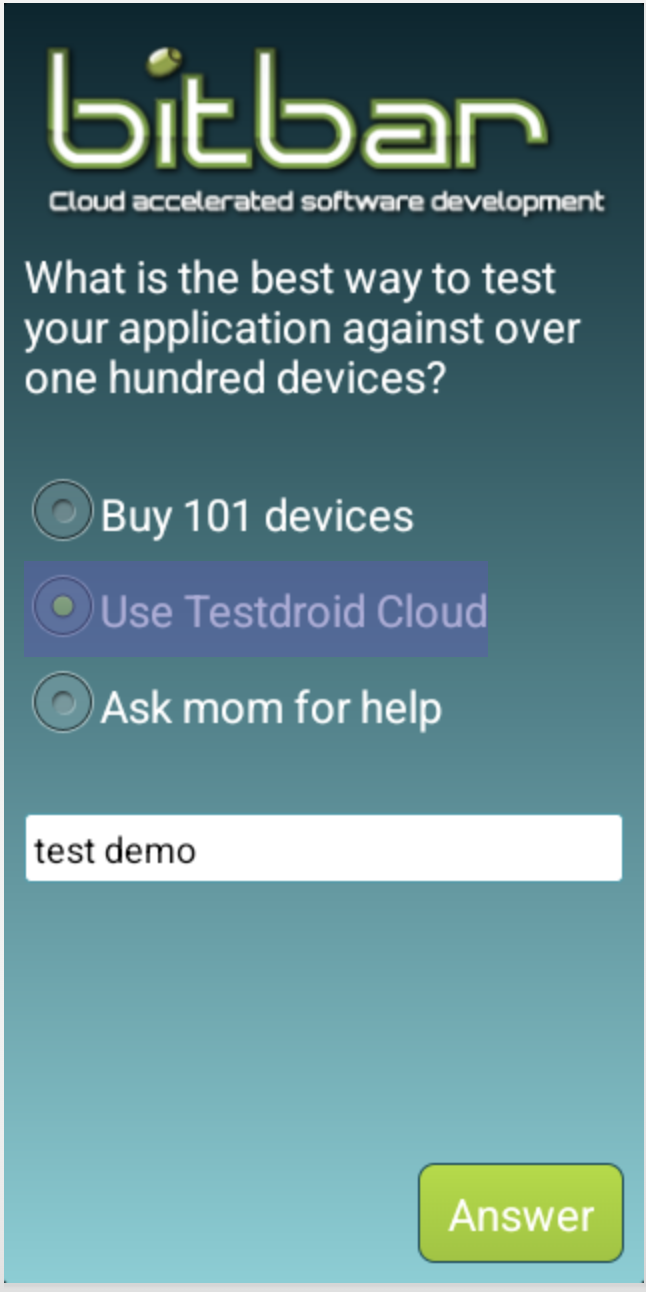
[https://github.com/appium/appium/tree/master/packages/appium/sample-code/apps](https://www.google.com/url?q=https://github.com/appium/appium/tree/master/packages/appium/sample-code/apps&sa=D&source=editors&ust=1717941557097918&usg=AOvVaw3C2E8Xr96692jWivH_e-up)

This contains both Android (.apk files) and iOS (.app files)  
  


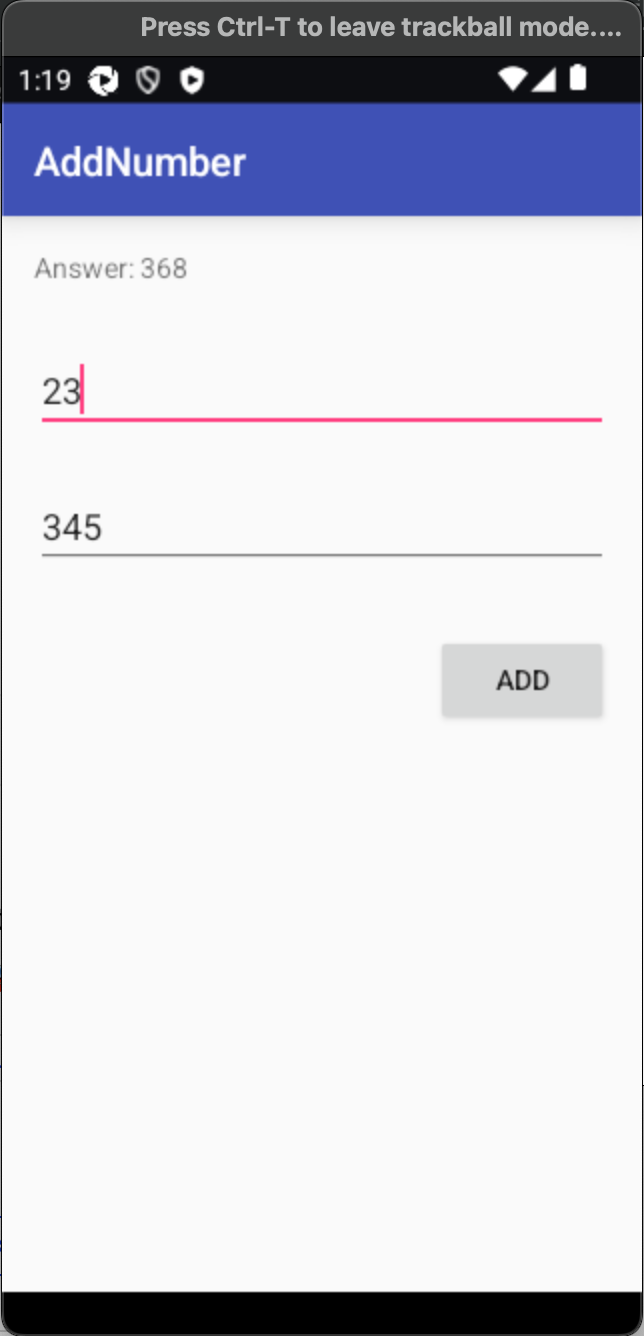
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Can search for APK files for apps like Calculator

[https://en.uptodown.com/android/search](https://www.google.com/url?q=https://en.uptodown.com/android/search&sa=D&source=editors&ust=1717941557098888&usg=AOvVaw3dptH92drv0GudBnE0lu5u)  
  

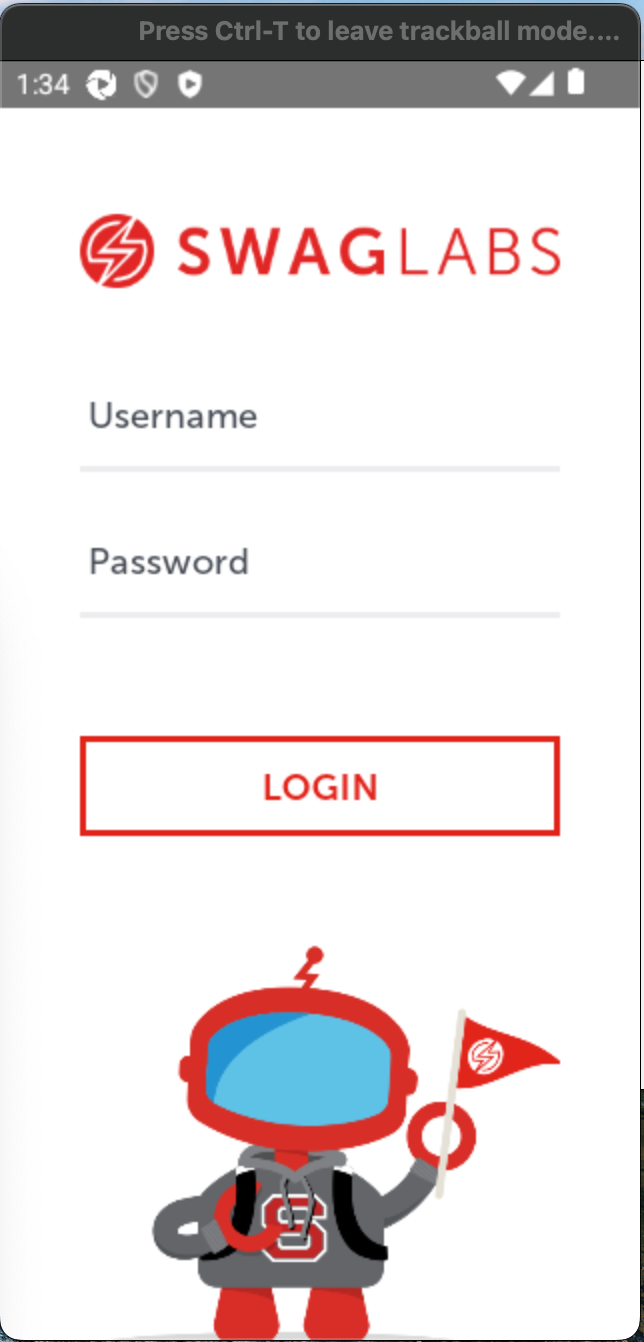

Bitbar sample APK <https://github.com/bitbar/test-samples/blob/master/apps/android/bitbar-sample-app.apk>  


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Browserstack<https://github.com/browserstack/app-testing-examples/blob/master/app-debug.apk>  


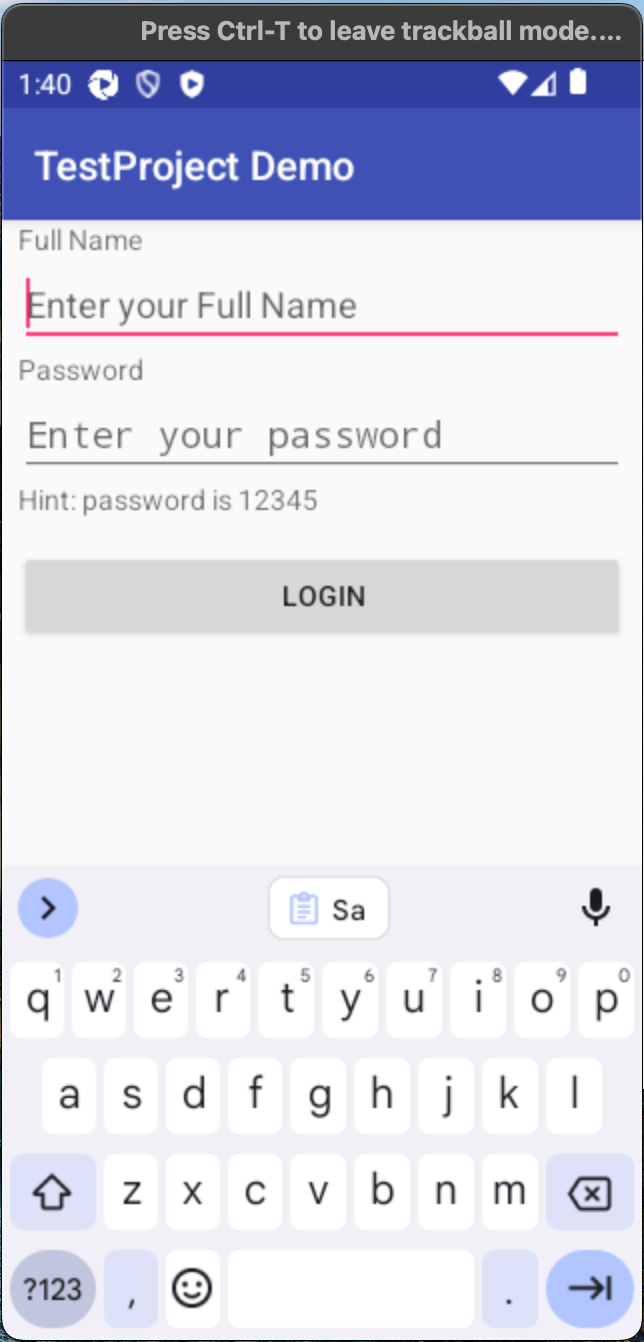
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SauceLabs demo Apps  
[https://github.com/saucelabs/sample-app-mobile/releases](https://www.google.com/url?q=https://github.com/saucelabs/sample-app-mobile/releases&sa=D&source=editors&ust=1717941557100659&usg=AOvVaw1H0TFS06e_s40jAi225yYP)

Includes both android and ios apps  
  


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TestProject demo app  
[https://github.com/testproject-io/android-demo-app](https://www.google.com/url?q=https://github.com/testproject-io/android-demo-app&sa=D&source=editors&ust=1717941557101198&usg=AOvVaw35PuKuqNFeYHUMu2cDCsO6)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Katalon Studio Sample apk files

https://github.com/katalon-studio-samples/katalon-mobile-automation/tree/master/Data%20Files