A MOBILE LABS GUIDEBOOK

All About Appium: Get Up and Running in One-Hour or Less

It's All About Appium, Isn't It?

When is the last time that you've had a conversation about mobile test automation that did not include Appium? Based on the excitement and fervor about Appium, it's probably been a while. As interest in Appium continues to grow and more mobile QA teams begin exploring this most intriguing open-source testing tool, industry professionals continue to learn more and become more confident in their Appium expertise.

From our Mobile Labs customers, we've learned that although there is a lot of interest around leveraging Appium for test automation, that getting started with Appium has its special set of unique challenges.

Here is just a sample of the things we've learned so far:

We've discovered that sometimes it's hard to get started testing with Appium because it's expensive or difficult to purchase Apple computers due to the Appium requirements for iOS testing. We've found that troubles often come from writing test scripts to use for running Appium tests. We've also observed that while a familiarity with Selenium for mobile web testing helps testers get up to speed with Appium faster, that they often still struggle when executing Appium tests.

Honestly, Team Mobile Labs is just as fascinated with Appium as our customers and industry professionals are. This interest in exploring Appium and all its nuances is what inspired our mission of making Appium easier to use by offering world-class Appium support. As the first step in our Mobile Labs Appium journey, we added a built-in, instant-on Appium server to our mobile device cloud, GigaFoxTM to offer better performance when running Appium tests, in addition to making it easier to get started with Appium. Believe it or not, GigaFox makes it possible to write your first test and execute it within an hour. Seriously.

We are also fortunate to have several in-house Appium experts on our support team that enjoy helping our customers navigate these Appium challenges, from setup to scripting. No matter where testers are in their Appium journey, it is an adventure that we can all embark on together. As we all learn more about Appium, we can help each other. Now that's a great way to make learning fun for everyone!

So, in the spirit of education, let's review some handy Appium basics that will give you a solid foundation on your Appium journey.

MOBILE LABS

Crash Course: Appium 101

LESSON #1

How Did We Get Here Anyway? From Selenium to Appium

While figuring out where you're going in your Appium adventures, it can be useful and interesting to explore what has happened in the recent past in our world to get you where you are today. Or in other words, where did Appium come from?

How did we go from mobile web testing to mobile app testing, and how did this trend result in the birth of Appium? The answer is actually…Selenium.
Sadly, it is often difficult to remember life before mobile. From bulky car phones in bags, to flip phones, to today's smartphones and tablets, it is amazing how quickly technology has evolved and its influence on consumer behavior.
As the types of available devices expanded, and internet connectivity continued to speed up, so did the abilities of mobile devices to connect, entertain and captivate the minds of consumers via the mobile web. And then, the explosion of mobile apps happened, which was a game changer for testing teams. As a result, simply testing mobile web was not enough – teams needed a solution to test these robust mobile apps.

With mobile apps and mobile web, consumers are able to do virtually anything on their mobile devices. Thanks to the rapid growth of mobile apps, testers found themselves with an entirely new platform for testing.
Before the explosion of mobile, web served as the primarily platform for testing. Testers relied on tools such as MicroFocus' UFT (formerly HP) and Selenium. But recently, Appium has made a huge impact in enterprise mobility for testers interested in test automation for mobile apps. When it comes to generating industry buzz, Appium has been a hot topic from thought leaders and vendors at enterprise mobility events worldwide.
But, how exactly did we get to this point? How did we go from mobile web testing to mobile app testing, and how did this trend result in the creation of Appium?

To answer this question, it is important to take a look at the history of web testing and Selenium.

History Time: Selenium and Web Testing

Before Selenium, when it came to web testing tools, UFT was the clear market leader for testers. As a commercial tool, UFT offered a robust support community and testers benefited from being connected to a larger ecosystem of tools formerly available from HP. Today, testers using UFT are now a part of the MicroFocus ecosystem, thanks to its acquisition of UFT. Today, testers leveraging UFT automation enjoy using a stable tool that has stood the test of time and is ideal for testers who are not as savvy when it comes to writing scripts and complex programming.

Although many testers embrace commercials tools there are other testers who thrive in an open-source environment and chafe against the restraints from commercial tools like UFT. Therefore, Selenium was developed to bring more freedom to web testing.

Selenium prides itself on being a tool that simply automates web browsers. As an open-source solution, testers are empowered to harness Selenium as a means of automating web applications for testing. Testers with the appropriate skill sets and hunger for innovation can really make the tool their own through customization.
As Selenium began to rise in popularity it pushed UFT out of the #1 spot for web testing tools. In fact, Selenium captured such mindshare in the industry that it is actually the core technology used by other browser automation tools, APIs and testing frameworks thanks to its powerful capabilities.

In addition, Selenium supports many different scripting languages, while commercial tools usually support only one language. This freedom in scripting languages opened up the world for testers by bringing much needed flexibility when writing test scripts.

Selenium also utilizes the web browser's native API when testing and easily integrates with other open- source tools for additional options for web testing and integration with other tools in the testing lab.
It is important to note that unlike commercial solutions, Selenium requires testers to have programming skills to write tests. For teams without deep programming skills, Selenium may be a challenge to use. For teams that have programming capabilities, Selenium allows testers to write tests in their language of choice and to their desired specifications.

AT-A-GLANCE: BENEFITS OF SELENIUM

- Open-source
- Supports a variety of scripting languages
- Utilizes the web browser's native API
- Easily integrates with other open source tools

Even though Selenium is still a robust testing tool used today for web testing, the growth of mobile apps required a new tool for testing on this platform. Thus, Appium was born to fulfill this need and has its roots in the Selenium framework.

Appium and the Open-Source Community

Like Selenium, Appium was developed by the open-source community to test mobile apps. Basically, Appium is "Selenium for Apps," as it is based on Selenium's Webdriver technology. Appium functions by using the native APIs of iOS and Android to communicate to the apps or web browser on a mobile device.

For history buffs, you can check out this website to get a comprehensive history of Appium. But in a nutshell, Appium was born from Co-Creator Dan Cuellar's need to test an iOS app while he was a test manager at Zoosk in 2011. To accomplish this task, he built iOSAuto, using the same underlying philosophy as Selenium by leveraging a native API. In 2012, Cuellar met Selenium's co-creator, Jason Huggins and together they created Appium using Selenium Webdriver's wire protocol over HTTP.

Although Appium in its earliest form has been around since 2012, it is only recently that Appium has really begun to resonate on a wide-scale in enterprise mobility. As a testing tool for apps, it offers many of the same benefits that Selenium does for web testing. Today, Appium has an active, open-source community that works to expand the tool's capabilities.

Also, many thought leaders and vendors in enterprise mobility have started to add Appium support capabilities into their current solutions. As Appium continues to grow in popularity, expect to see more features and functions from vendors and more insight from thought leaders around this testing framework.
Perhaps one of the most interesting things about the relationship and connection between Selenium and Appium lies in the power of pairing the two solutions for testing – bringing a powerful suite of tools to today's testing labs for web and mobile app testing.

Selenium + Appium = A Powerful Toolkit for Testing

For testers that are familiar with Selenium, and have used the tool, getting up to speed with Appium for mobile app testing is an easy learning curve. Because Appium is built on Selenium's framework, the skills are easily transferable and intuitive.

For teams that are using Selenium and Appium, testers can also leverage current web browser test assets (provided the object names are the same). It is important to note that for both native and hybrid apps while the structure of the tests are similar, the type of objects in the tests are different.

By leveraging both tools in today's modern testing lab, testers have the best of both worlds. If they are familiar with Selenium and use it for web testing, then bringing in Appium to test mobile apps is a logical next step. If teams are using Selenium for web testing, then these scripts can be easily converted for Appium using a mobile web browser.

As an extension of Selenium that was built using its capable technology, it is easy to understand why the excitement around Appium continues to grow, with Selenium still maintaining its popularity as a trusted tool for web testing.
But, if you're ready to jump in and get started with Appium, what is the best way to make this transition easier? Let's consider the benefits of a mobile device cloud with a built-in Appium server to give your team a leg up in getting started and testing with Appium.

Crash Course: Appium 101

LESSON #2

3 Benefits of Appium Integration with a Mobile Device Cloud

It's no secret that some enterprise mobility teams are still struggling to get Appium up and running. But what if there was a way to make using Appium easier? The answer lies in leveraging Appium with a mobile device cloud.
Because Appium is open-source, the framework itself is constantly evolving and changing. Thanks to its dedicated community of developers, Appium is a living, breathing entity that has the potential to be

shaped by the professionals who use it daily. This is truly an exciting notion for enterprise mobility and a great opportunity to embrace innovation.
Despite the buzz around Appium and its potential to enhance mobile test automation strategies, some teams are still struggling to get Appium up and running. Having each team member in a testing organization configure Appium on their workstation can be both time-consuming and challenging. But what if there was a way to make using Appium easier? The answer lies in leveraging Appium with a mobile device cloud.

At Mobile Labs, we are currently developing solutions to help busy mobility teams make working with Appium easier than ever. By adding a built-in Appium server to our private mobile device cloud, GigaFox, we have developed a solution to further streamline mobile development, automated testing using Appium, and device sharing. But, how can you benefit from this integration?
Here are the top 3 ways that your team can benefit by leveraging a device cloud with Appium.

Top 3 Benefits of Leveraging Appium with a Device Cloud

#1: Avoid headaches of setting up Appium

Thanks to increased mobile and digital demand, enterprise mobility teams are busier than ever. Mobile developers, testers and QA are all faced with a unique set of challenges as they constantly strive to create high-quality mobile apps and mobile websites. Although leveraging automated testing using Appium would benefit the team in the long run by being able to test faster, getting started with Appium is a bit of a learning curve. As enterprise mobility teams who have already installed Appium can attest, switching to Appium can be a daunting prospect due to its complex installation process.

To begin using Appium, enterprise mobility teams will need to install Appium, along with all its dependencies on each workstation. The process of wading through Appium documentation, configuring dependencies, and troubleshooting the inevitable road blocks for each testing machine can be a challenge even for the most technically savvy user. Having an entire team spend hours or even days of work just to get things set up can be a waste of valuable time.

To relieve this pressure and to make it easier for teams to get started using Appium, our GigaFox's built- in Appium server now comes with its dependencies already configured, set up and installed. By integrating Appium with our device cloud solution, GigaFox can be a "one stop shop" for mobile development, testing and device sharing through one streamlined platform. We take care of the Appium maintenance so you can focus on testing.

#2: Use your current Appium Scripts

Another benefit of GigaFox's built-in Appium server relates to scripts. Testers can still write their scripts in Eclipse using JAVA or their language of choice. But what makes leveraging GigaFox and Appium together so revolutionary for scripting is that testers can reuse existing scripts by simply adding parameters and pointing the script to the GigaFox server. Users can run up to eight tests simultaneously on eight devices.

Also, adding a device to a workstation is an easy process. Once the device is added, testers can then access it for testing remotely. This prevents testers from having to connect real devices to their workstation each time a test needs to be run. Removing the headache of device sharing speeds up the process of testing for increased DevOps and productivity.

Considering implementing parallel testing? Here's a fun activity for you. Check out our ROI Calculator for Parallel Testing on our website and see for yourself the time and money you can save with this approach.

#3: Test on iOS and Android

Worried that you won't be able to test iOS devices on your Windows workstations? With a standard Appium setup, you'd be right. However, with GigaFox with integrated Appium, you can test iOS or Android devices on Windows, iOS or even on Linux. This is possible thanks to GigaFox's built-in Appium server, where Appium is running on its server and is accessed and viewed by enterprise mobility professionals over the web.

At Mobile Labs, we are committed to helping enterprise mobility teams streamline the process of mobile development, testing and device sharing. That mission is why we created our mobile device cloud solutions. But, to empower mobile developers, testers and QA, we have taken this mission one step further by making it easy for teams to leverage cutting-edge tools like Appium. By adding a built-in Appium server to GigaFox, it is our goal to make working with Appium as easy as possible.

Crash Course: Appium 101

LESSON #3

Are you Appium Ready? Top 6 Ways to Leverage Appium

What does it mean to be Appium ready? With all of the buzz in enterprise mobility around test automation leveraging Appium, what can enterprise mobility teams do to be successful?

For mobile developers, testers and quality assurance professionals who are considering using Appium or who are just getting started, clearly investing in a mobile device cloud with built-in Appium support is an option worth exploring. A mobile device cloud not only helps enterprise mobility teams manage and share devices, but this centralized platform is also helpful for streamlining automated testing and continuous delivery initiatives. For teams interested in using Appium for automated testing, a mobile device cloud with a built-in Appium server will make getting started with Appium easy and more efficient. Furthermore, a mobile device cloud with built-in Appium support will provide an extra cushion against any hiccups that may occur when using an open-source platform for testing.

Still not convinced? Here are six benefits of pairing Appium with a mobile device cloud.

#1: Take Advantage of a Built-in Open-Source Appium Server

At Mobile Labs, we are committed to the open source Appium project. We believe that Appium should not behave differently if your mobile testing team wants to take scripts to another vendor or if they want to use their own Appium server. Our goal is to make working with Appium easier. One way we are

doing this is by bundling the Appium server into our completely re-engineered mobile device cloud, GigaFox By building an Appium server into the GigaFox infrastructure, we are not changing the way Appium works from a scripting standpoint. We are also not providing enterprise mobility professionals with an Appium compatible server that uses a different server technology from Appium. By leveraging a real, built-in Appium server we are able to respond and to account for any new features as Appium continues to evolve.

#2: World-class Appium Customer Support

Appium is an open source project that depends on the community of users to contribute updates. This means that there is no official support and that bug fixes will be patched when community developers pick them up and contribute the fix back to the project. Fortunately, Appium has corporate sponsors who are invested in its success and who dedicate resources. Therefore, bug fixes usually come quickly, and we can be assured that the entire project will not be abandoned anytime soon. For Mobile Labs' customers, because we have included Appium as a part of GigaFox, enterprise mobility teams have access to Mobile Labs' world class customer support for any Appium issues. Mobile Labs provides 24/7 customer support and our in-house experts will work to fix any issues that may occur. In fact, our team will even contribute bug fixes back to the Appium open source project to help benefit mobile developers, testers and QA teams who may experience similar issues.

#3: Windows Support for iOS Automation

The Appium open source project requires enterprise mobility teams to have a Mac environment to run Appium tests on iOS devices. At its core, Appium ensures that apps do not have to be modified and therefore can use the native automation technologies provided by the OS vendor. For iOS, this means that testers must use Apple's processes and tools that only run on a Mac. In addition, the Appium server for iOS devices should also be run on a Mac. Appium scripts for iOS can run from any environment if they point to a Mac Appium server. Many organizations do not support Mac environments, as the procurement and security restrictions on Apple hardware can cause issues when trying to run Appium tests on iOS. GigaFox uses a Mac server, meaning that enterprise mobility teams will not have to make additional Apple purchases to work with Appium.

#4: Simplified Startup

Enterprise mobility teams who are considering Appium or who are just getting started may have noticed that setting up an Appium environment requires a lot of steps. From iOS provisioning to server configurations, Mobile Labs has made getting started with Appium easy for teams with our built-in Appium server for GigaFox. We have also provided capabilities to select an appropriate device for testing, in case the device used to script a test is being used by another developer, tester or QA professional. This simply means that GigaFox will retain a device that matches the app's OS version criteria or preset OS version criteria without having to specify a particular device. GigaFox also handles the iOS provisioning to make sure Appium will run on each iOS device.

#5: Parallel Testing and Appium Administration

Once enterprise mobility teams have figured out how to get one test running against one device on an Appium server, the next step is to figure out how to run additional tests against many devices on one or more operating systems. Starting the Appium server for real devices can be a challenge because teams must make sure that all parameters and ports are configured properly. Appium is starting to make this easier by giving parameters for different ports in the startup capabilities.

But this is not the only issue to be concerned with when executing parallel Appium tests. There are other dependencies that are needed to make sure that Appium is ready to run on various devices at the same time. Some of these dependencies include leveraging additional open-source tools that must be running on the server and must be properly configured to run against the same device as the Appium server. When it comes to running devices in parallel, this can be daunting when making sure that startup scripts are working properly before running tests.

Finally, when it comes to running a script, it is necessary to change the port for each device for each run. The consequence of multiple ports is that testers will need to handle the differences in scripts each time the test is run against a different device. This can cause a bunch of unnecessary code maintenance. Appium has provided some welcome relief as testers can register Appium nodes and point tests to a Selenium grid. However, this is a separate area that enterprise mobility teams will need to maintain and configure.

With GigaFox, Mobile Labs takes care of the Appium server because all scripts point to GigaFox for execution. GigaFox handles the Appium server startup automatically. This means that enterprise mobility teams just need to worry about writing and executing scripts. The startup scripts and server configuration are handled by Mobile Labs and GigaFox, bringing necessary relief to a busy team of mobile developers, testers and QA teams to focus on more pressing issues.

#6: Selenium Web Testing on Real Devices

Appium is popular because of the success of Selenium as a web automation tool. Many organizations have made huge investments in Selenium scripts for their websites. Now, it is easy to run these scripts against a variety of real devices to determine if the site loads and functions properly on a real device.

Simulators and emulators can be a good source for testing, but Apple and Google recommend testing against real devices. Keep in mind that the hardware does make a difference. If working with a responsive website, it is more important to test across a variety of devices with various OS versions and different screen sizes. Appium will run the Selenium scripts on native browsers (Safari and Chrome) on each device. Using GigaFox allows teams to have the variety of devices needed for testing, while making it easy to run Selenium scripts against the built-in Appium server.

How-To: Getting Started With Appium in One Hour or Less

So, you've decided to take the plunge and embark on automated testing with Appium. As you've discovered, by leveraging a mobile device cloud solution your testing lab with a built-in Appium server, your team gets to skip the long process of building and setting up for Appium. Taking advantage of a

mobile device cloud with a built-in Appium server will eliminate a cumbersome set up process and have you testing within an hour of installing of your mobile device cloud.

When you combine the easy set-up with a high performing Appium server, it's a combination that really cannot be beat.
Now that you've decided to embrace Appium and get started testing, here's some things to consider on your Appium journey.

Crash Course: Appium 101

LESSON #4

5 Things to Consider When Getting Started with Appium

So, You've Installed Appium, But Now What?
Although many enterprise mobility teams have installed Appium, they still aren't sure what to do next.

To get you going and off to a running start, here's 5 things to consider when beginning your Appium journey.

Top 5 Things to Consider When Getting Started with Appium

#1: Appium is Open Source

Because Appium is an open-source test automation framework for mobile, there is no official support for the product. Issues are logged in GitHub and are fixed by the Appium developers or contributors. Therefore, issues may not be resolved as quickly as older more mature products. Appium has had frequent releases, with some improving the way objects are found. These releases may require more maintenance to scripts to keep up with the changes. These changes may also introduce bugs that break your script runs. Make sure you plan for this in your testing environment

#2: Understand Appium Requirements for your App

Apps may require specific provisioning that require you to register a device with your development program. iOS applications must be built for development, which means that the device must be registered to your Apple Developer program and signed with development profiles and certificates. If you are currently testing an app signed with an Enterprise profile and certificate, you will want to make sure your development team knows the new requirements for Appium.

If you plan on using real devices and simulators, you may need different builds of your application. For instance, iOS requires an app that runs on a simulator to be compiled differently than the app that runs on an iPhone or iPad.

Check out the following chart to get a better idea of Appium requirements for mobile apps compared to some other tools you may be familiar with.

	Robotium	uiautomater	Espresso	Appium	Calabash
Android	Yes	Yes	Yes	Yes	Yes
iOS	No	No	No	Yes	Yes
Mobile Web	Yes	Limited x.y Cicks	No	Yes (Android & iOS)	Yes (Android)
Scripting Language	Java	Java	Java	Almost Any	Ruby
Test Creation Tools	Testdroid Recorder	UI Automator Viewer	Hierarchy Viewer	Appium.app	CLI
Supported API Levels	All	16 =>	8,10,15 =>	All	All
Community	Contributors	Google	Google	Active	Pretty Quiet

#3: Determine Which Programming Language and Testing Framework to Use

Unlike a more traditional testing tool like HPE Unified Functional Testing (UFT) where everything is provided in a nice environment that handles scripting, execution, and reporting, you must handle these things separately with Appium. Appium comes with client libraries for all major programming languages such as JQuery and HTML5. The first choice you will make with Appium is to choose your language and then the IDE that can handle the chosen language. For instance, you might use Eclipse or IntelliJ to handle writing your scripts in Java. Once you have a language, you will need to choose the testing framework to execute. JUnit or TestNG are popular choices for Java. Finally, you must configure reports to show progress. Most testing frameworks will give you the results at the end.

#4: Understand Which Type of Computer Can Run your Real Devices

Appium can be used for iOS and Android, but if you plan on testing iOS apps or mobile web with Safari, you are going to need an Apple computer. The reason is that Appium, at its core, does not want to modify an application for testing purposes. Therefore, the technology used to interact with iOS comes from Apple provided services. These services are required by license to run on an Apple computer. It is important to plan for this if your testing organization only has Windows computers. Android devices can run on any computer as its services are not tied to an operating system.

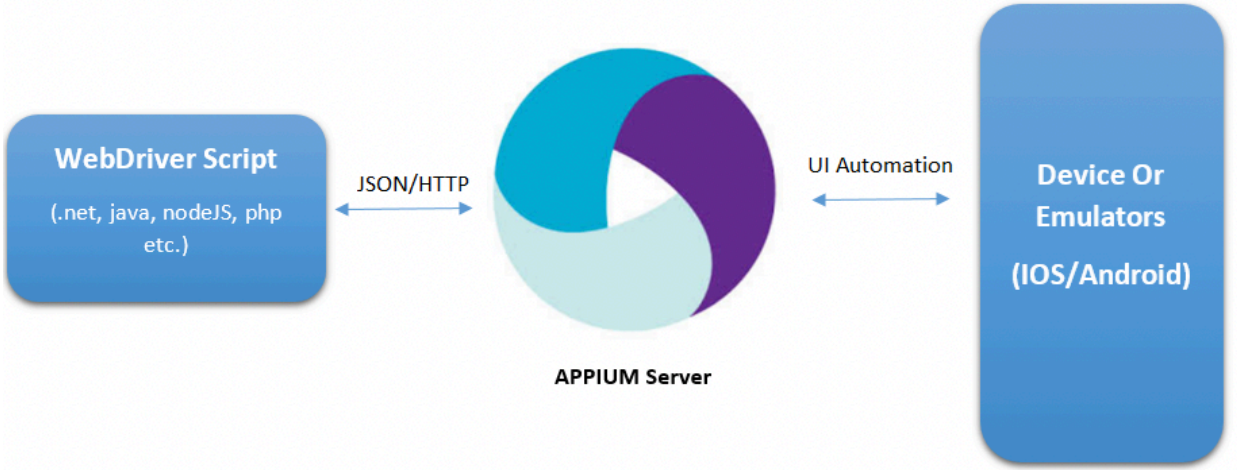
#5: Realize How the Appium Server Runs

The Appium server implements the JSON wire protocol that is the standard interface for Selenium. When a test is executed, Appium interprets the methods into the native automation calls provided by iOS and Android. There are several additional components involved in this process that must be properly configured for an Appium server to execute the steps of a test. When wanting to execute multiple scripts at the same time, the configurations will change, such as ports and folder paths. It's important to understand these complexities to begin executing your full testing library.

When comparing Appium vs Selenium, Appium is not as straightforward as its desktop counterpart Selenium. Understanding what is involved is vital to making your automation strategy work with

Appium. At Mobile Labs, we have taken some of these things into consideration and our device cloud, GigaFox is Appium ready. GigaFox has a built in Appium server taking away the complexities of managing devices and using them in your Appium framework.

The chart illustrates how the Appium server runs.



In Conclusion: Why GigaFox?

As Appium continues to grow and gain traction in enterprise mobility, more and more mobile development and QA teams will begin to embrace this framework in their testing labs. Sure, teams could set up their own workstations and Appium servers in-house, but why work harder if you don't have to?

A mobile device cloud such as Mobile Labs' GigaFox, that features a built-in Appium server, has many benefits for a team that is interested in Appium. In fact, due to GigaFox's speed and performance, we believe any team that implements our mobile device cloud infrastructure will become an Appium superstar in no time.

In a nutshell, GigaFox:

- Improves Appium performance and reliability
- Runs more Appium concurrent tests
- Makes iOS provisioning fast and easy

In addition to amazing Appium prowess, GigaFox can streamline device sharing and management among all members of the team regardless of location. Offered on-premises or hosted in Mobile Labs' secure data center, your team has the power to build the Appium-powered mobile testing labs of your dreams. Why not get started today?

About Mobile Labs

Mobile Labs remains the leading supplier of in-house mobile device clouds that connect remote, shared mobile devices to Global 2000 mobile web, gaming, and app engineering teams. Its patented GigaFoxTM is offered on-premises or hosted, and solves mobile device sharing and management challenges during development, debugging, manual testing, and automated testing. A pre-installed and pre-configured Appium server provides "instant on" Appium test automation. GigaFox enables scheduling, collaboration, user management, security, mobile DevOps, and continuous automated testing for teams spread across the globe and can connect cloud devices to an industry-leading number of third-party tools. For more information please visit mobilelabsinc.com.

Available on-premises or hosted solutions

Free Trials Available

Contact Mobile Labs

info@mobilelabsinc.com
1 (404) 214-5804
www.mobilelabsinc.com

MOBILE LABS