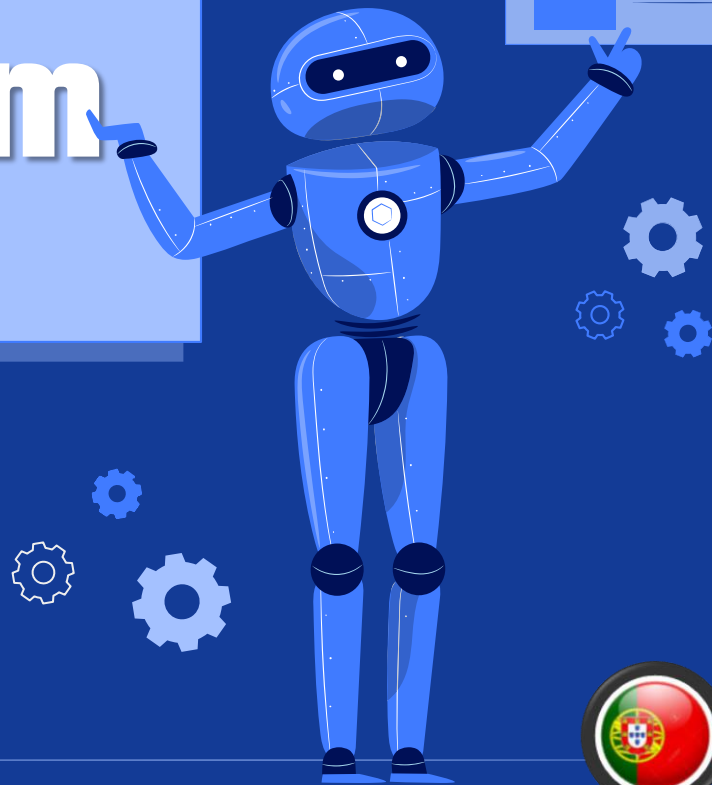


Appium

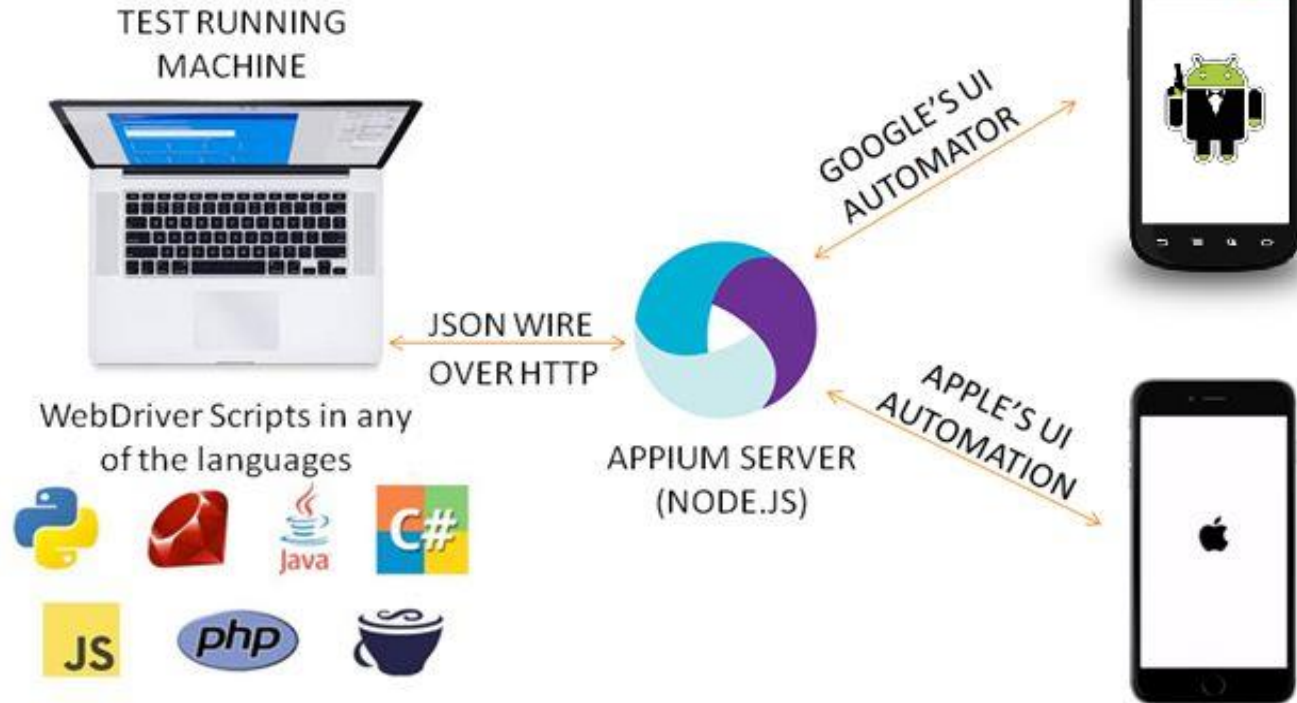
Victor Portugal





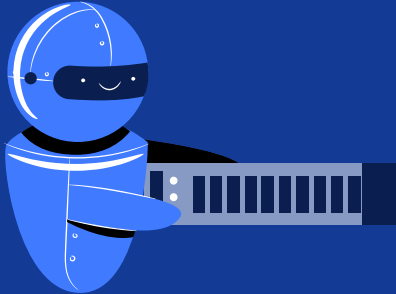
- Appium es una herramienta open-source para la automatización de aplicaciones web nativas e híbridas en las plataformas móviles iOS y Android, y en la plataforma de escritorio Windows.
- Es cross-platform por lo que es posible crear pruebas en diversas plataformas utilizando la misma API. Esto permite la reutilización de código entre conjuntos de casos de prueba definidos para una aplicación desarrollada para varias plataformas.







Appium emplea el WebDriver de Selenium, que es un entorno de pruebas para aplicaciones web y que especifica un protocolo cliente-servidor conocido como JSON Wire Protocol. Esto permite al cliente el uso de marcos de pruebas escritos en cualquier lenguaje, enviando las solicitudes HTTP apropiadas al servidor.





Es importante destacar que Appium es “multiplataforma”: le permite escribir pruebas en múltiples plataformas (iOS, Android, Windows), utilizando la misma API. Esto permite la reutilización de códigos entre las salas de pruebas de iOS, Android y Windows.

El funcionamiento es el siguiente :

01

- La prueba que se está ejecutando en la máquina es enviada al servidor

02

- Este envía los comandos al framework para pruebas correspondiente a cada plataforma.

03

- En Android es Google 's UI Automator, en iOS Apple 's UI Automation.

Filosofía de Appium



Appium fue diseñado para
satisfacer las
necesidades de
automatización móvil de
acuerdo con una filosofía
descrita por los
siguientes cuatro
principios:

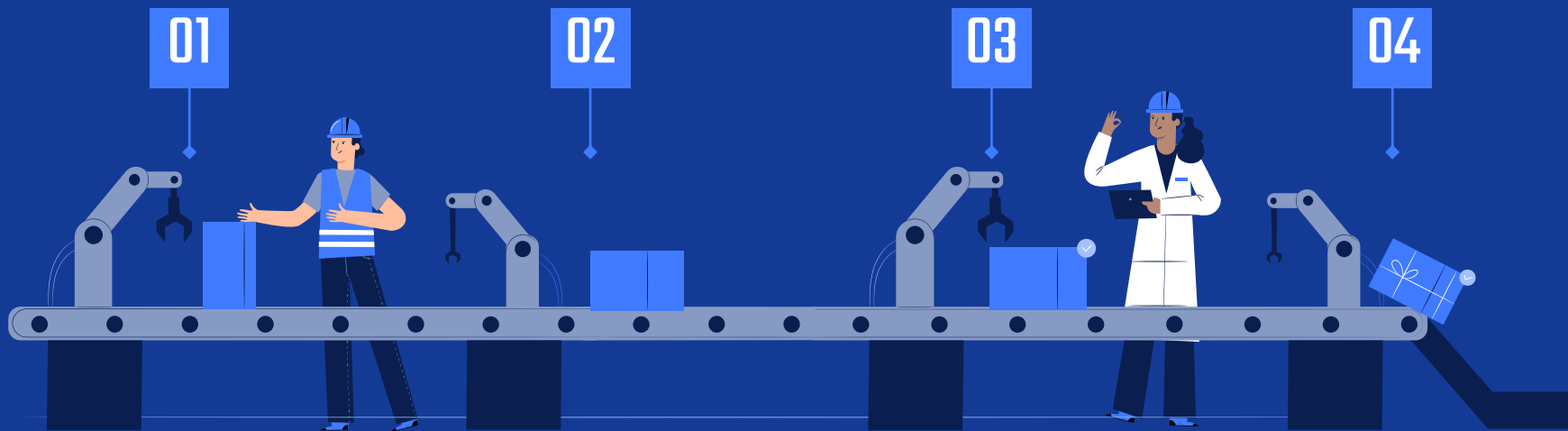


No debería
tener que
volver a
compilar su
aplicación o
modificarla de
ninguna manera
para
automatizarla

No debe estar
encerrado en
un idioma o
marco
específico para
escribir y
ejecutar sus
pruebas.

Un marco de
automatización
móvil no debe
reinventar la
rueda cuando se
trata de API de
automatización.

¡ Un marco de
automatización
móvil debe ser de
código abierto,
tanto en espíritu
como en práctica,
así como de nombre!



Diseño de Appium

Entonces, ¿cómo vive la estructura del proyecto Appium esta filosofía? Cumplimos con el requisito n.º 1 mediante el uso de marcos de automatización proporcionados por el proveedor bajo el capó. De esta forma, no necesitamos compilar en su aplicación ningún código o framework específico o de terceros de Appium. Esto significa que estás probando la misma aplicación que estás enviando. Los marcos proporcionados por el proveedor que utilizamos son:





...
iOS 9.3 y superior: XCUITest de Apple
iOS 9.3 y versiones anteriores: UIAutomation de Apple
Android 4.2+: UiAutomator / UiAutomator2 de Google
Android 2.3+: Instrumentación de Google . (El soporte de instrumentación se proporciona agrupando un proyecto por separado, Selendroid)
Windows: WinAppDriver de Microsoft

- Cumplimos con el requisito n. ° 2 ajustando los marcos proporcionados por el proveedor en una API, la API de WebDriver.





WebDriver (también conocido como “Selenium WebDriver”) especifica un protocolo cliente-servidor (conocido como el protocolo JSON Wire Protocol). Dada esta arquitectura cliente-servidor, un cliente escrito en cualquier idioma puede usarse para enviar las solicitudes HTTP apropiadas al servidor. Ya hay clientes escritos en todos los lenguajes de programación populares. Esto también significa que puede utilizar el corrector de prueba y el marco de prueba que desee; las bibliotecas cliente son simplemente clientes HTTP y se pueden mezclar en su código de la forma que desee. En otras palabras, los clientes de Appium & WebDriver no son técnicamente “frameworks de prueba”, son “bibliotecas de automatización”.

¡Puede gestionar su entorno de prueba de la forma que desee!

01.

Cumplimos con el requisito n.º 3 de la misma manera: WebDriver se ha convertido en el estándar de facto para la automatización de navegadores web, y es un borrador de trabajo del W3C . ¿Por qué hacer algo totalmente diferente para dispositivos móviles?

02.

En cambio, hemos extendido el protocolo con métodos adicionales de API útiles para la automatización móvil.

Debería ser obvio que el requisito # 4 es un hecho, estás leyendo esto porque Appium es de código abierto .



Arquitectura cliente / servidor

Appium es en esencia un servidor web que expone una API REST. Recibe conexiones de un cliente, escucha los comandos, ejecuta esos comandos en un dispositivo móvil y responde con una respuesta HTTP que representa el resultado de la ejecución del comando. El hecho de que tengamos una arquitectura de cliente / servidor abre muchas posibilidades: podemos escribir nuestro código de prueba en cualquier idioma que tenga una API de cliente http, pero es más fácil usar una de las bibliotecas de cliente de Appium. Podemos poner el servidor en una máquina diferente a la que se ejecutan nuestras pruebas. Podemos escribir un código de prueba y confiar en un servicio en la nube como Sauce Labs para recibir e interpretar los comandos.

Automatización de sesión siempre se realiza en el contexto de una sesión. Los clientes inician una sesión con un servidor de formas específicas para cada biblioteca, pero todos terminan enviando una solicitud POST /session al servidor, con un objeto JSON llamado el objeto 'capacidades deseadas'. En este punto, el servidor iniciará la sesión de automatización y responderá con una ID de sesión que se utiliza para enviar más comandos.

Clientes de Appium

- Hay bibliotecas de clientes (en Java, Ruby, Python, PHP, JavaScript y C #) que son compatibles con las extensiones de Appium para el protocolo WebDriver. Cuando usa Appium, desea utilizar estas bibliotecas de cliente en lugar de su cliente WebDriver normal.



Appium Desktop



Hay un contenedor GUI alrededor del servidor Appium que se puede descargar para cualquier plataforma. Viene incluido con todo lo necesario para ejecutar el servidor de Appium, por lo que no necesita preocuparse por el nodo. También viene con un Inspector, que le permite verificar la jerarquía de su aplicación. Esto puede ser útil al escribir pruebas.

