

## Appium tutorial

1. Instalar jdk 1.8.0\_202 e api do Android: 25 ou superior.
2. Configurar variáveis ambiente: JAVA\_HOME, ANDROID\_HOME, platform-tools, emulator, tools e tools\bin.
3. Instalar a versão 1.3.2 do appium, segue o link: <https://github.com/appium/appium-desktop/releases/tag/v1.3.2>, você deve instalar o .exe para desktop com sistema windows.
4. Instalar o eclipse e configurar um projeto Maven com jdk 1.8.0\_202.
5. Configurar as dependências do pom.xml, abaixo as dependências:

```
<dependencies>
  <dependency>

    <groupId>io.appium</groupId>
    <artifactId>java-client</artifactId>
    <version>5.0.4</version>

  </dependency>
</dependencies>
```

6. Executar o Appium clicando em: **start server v.1.7.2**.
7. Na nova tela do appium, clicar em **start inspector session**, o ícone:



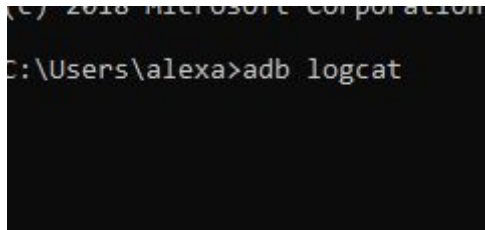
8. Na nova tela, clicar na aba **Automatic Server**
9. Executar o emulador do Android, depois abrir o cmd do windows para verificar o nome do adb device que você vai usar:

```
C:\Users\alexa>adb devices
List of devices attached
0047593311    device

C:\Users\alexa>
```

o meu é: 0047593311

10. Executar a aplicação que vai ser utilizada para testes no seu celular ou emulador e anotar a hora que você executou a aplicação.
11. Encontrar o pacote e o activity executando os comandos: **adb logcat**



- 12.
13. Esperar alguns segundos para ele carregar todos os processos.
14. Control F para localizar e procurar por **for activity**.
15. Depois de localizar **for activity**, o que vier depois do nome for activity é o package e o activity separados por uma "/" que vai ser utilizado para preencher os campos do desired capabilities do Appium. Exemplo do package do MVP google: **com.example.android.architecture.blueprints.todomvp.mock**  
Exemplo do activity do MVP google: **com.example.android.architecture.blueprints.todoapp.tasks.TasksActivity**
16. Configurar desired capabilities:  
**platformName** - Text - android  
**deviceName** - Text - 0047593311 obs:número do seu device  
**appPackage** - Text - com.example.android.architecture.blueprints.todomvp.moc  
**appActivity** - Text - com.example.android.architecture.blueprints.todoapp.tasks.TasksActivity
17. Depois de configurar o desired clicar em **Start Session**
18. O celular ou emulador e a nova tela do appium ficarão sincronizadas para capturar todos os cliques e cadastros. Nesta nova tela é possível criar uma simulação automática com código em java que vai ser utilizado no Eclipse com o projeto maven.
19. Iniciar o record que tem o ícone de um olho. Criar os passos para executar a tarefa. Com os passos criados, automaticamente é gerado um código java que vai servir para automatizar a execução das tarefas usando o Maven para gerenciar os passos.
20. Appium também fornece o código para se conectar com o celular, exemplo: **DesiredCapabilities desiredCapabilities = new DesiredCapabilities();**  
**desiredCapabilities.setCapability("deviceName", "0047593311");**  
**desiredCapabilities.setCapability("platformName", "android");**

```
desiredCapabilities.setCapability("appPackage",  
"com.example.android.architecture.blueprints.todoapp.mock");  
desiredCapabilities.setCapability("appActivity",  
"com.example.android.architecture.blueprints.todoapp.tasks.TasksActivity");  
desiredCapabilities.setCapability("automationName", "UiAutomator2");  
desiredCapabilities.setCapability("noReset", true);  
desiredCapabilities.setCapability("eventTimings", true);
```

Este código vai ser utilizado no Maven para sincronizar com o servidor do appium e executar os passos com o código gerado pelo Appium em java para criar, deletar e editar.

21. Tutorial caso fique dúvida:

<https://www.youtube.com/watch?v=i1tQ1pjEFWw>

22. Os teste são executados todos pelo Maven no eclipse, o servidor do appium precisa estar aberto.

23. Código fonte do projeto:

<https://github.com/alexandrefcesar/tcc-codigo-amstras>

24. Pasta do Maven é “teste”, Pasta das aplicações Android recebem o nome de MVP e MVVM, Pasta de resultados está em “Amstras”.