**Run JDI test for Appium from scratch**

To have Appium up and running on your system, following components need to be installed:

1. [JDK](#_JDK_installation)
2. [Test Application](#_Test_Application)
3. [Node JS](#_Node_JS)
4. [Android SDK](#_Android_SDK)
5. [Appium Server](#_Appium_Server)

Since everything is installed, it is possible to create your [first test with JDI with Appium](#_Create_your_first)

## **JDK installation**

1. Download jdk from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Setup JAVA\_HOME system variable to ‘C:\Program Files\Java\jdk1.8.0\_60 (or the path your java is installed)
3. Add ‘%JAVA\_HOME%\bin’ to path system variable

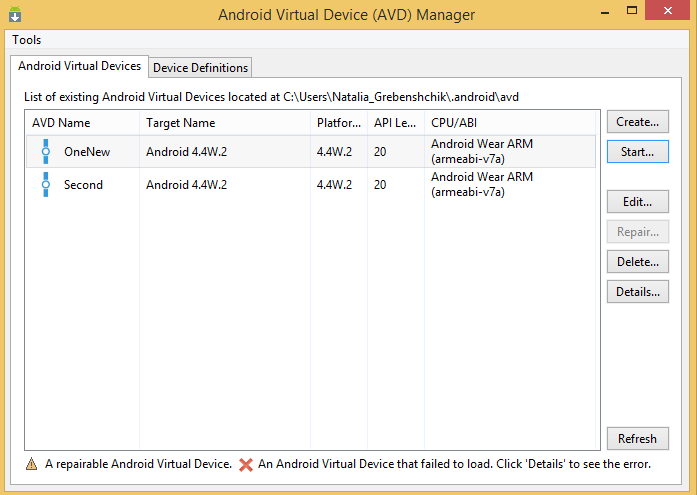
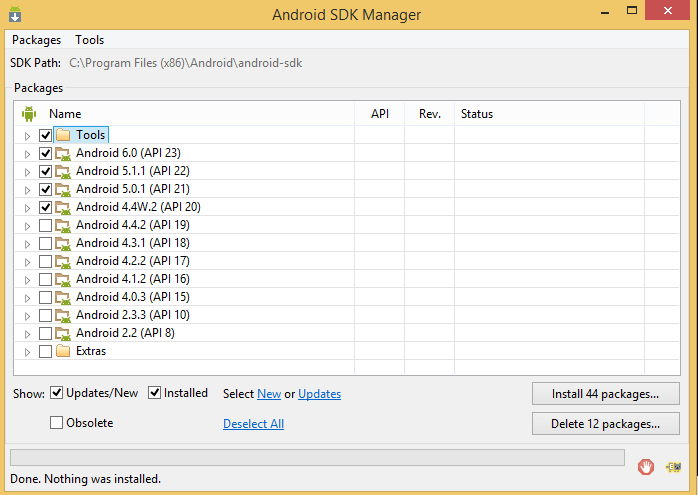
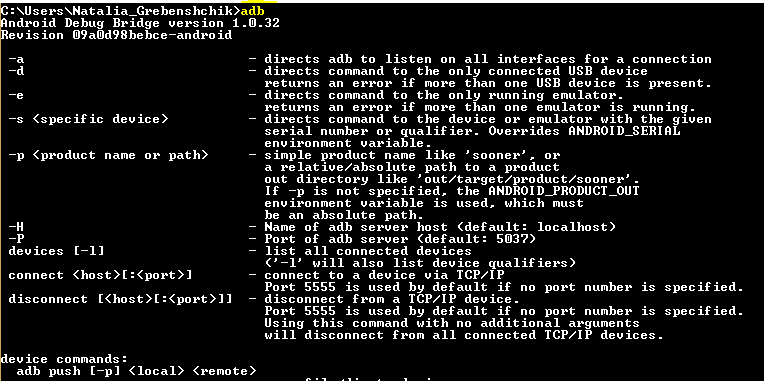
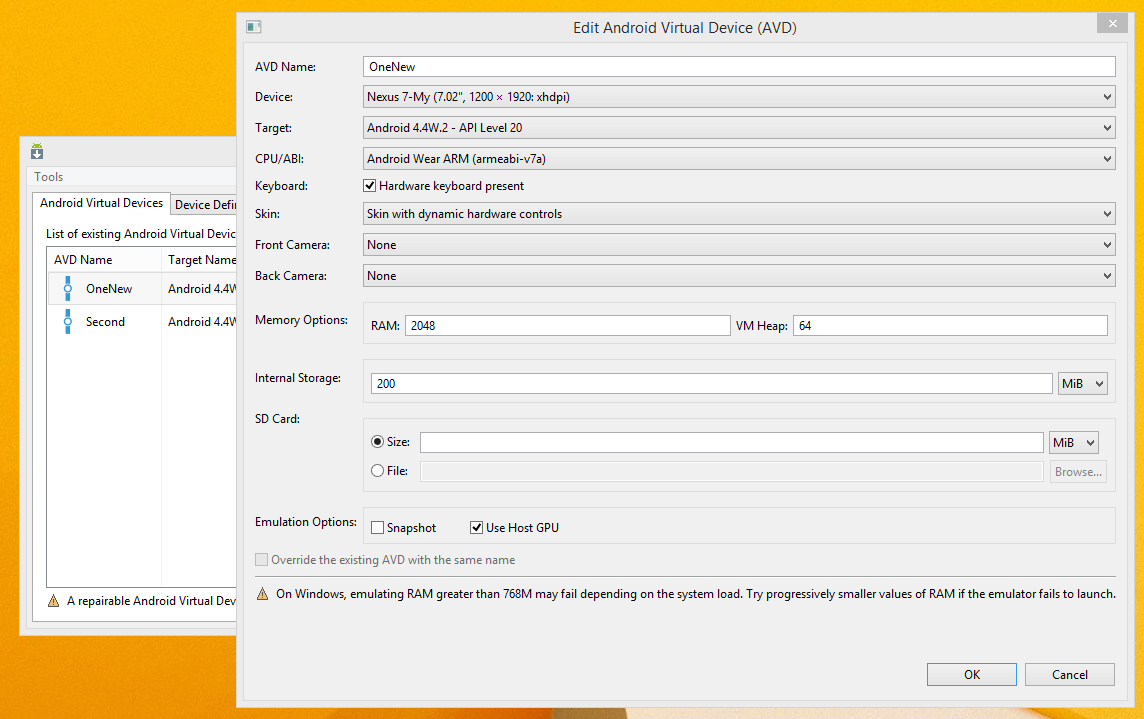
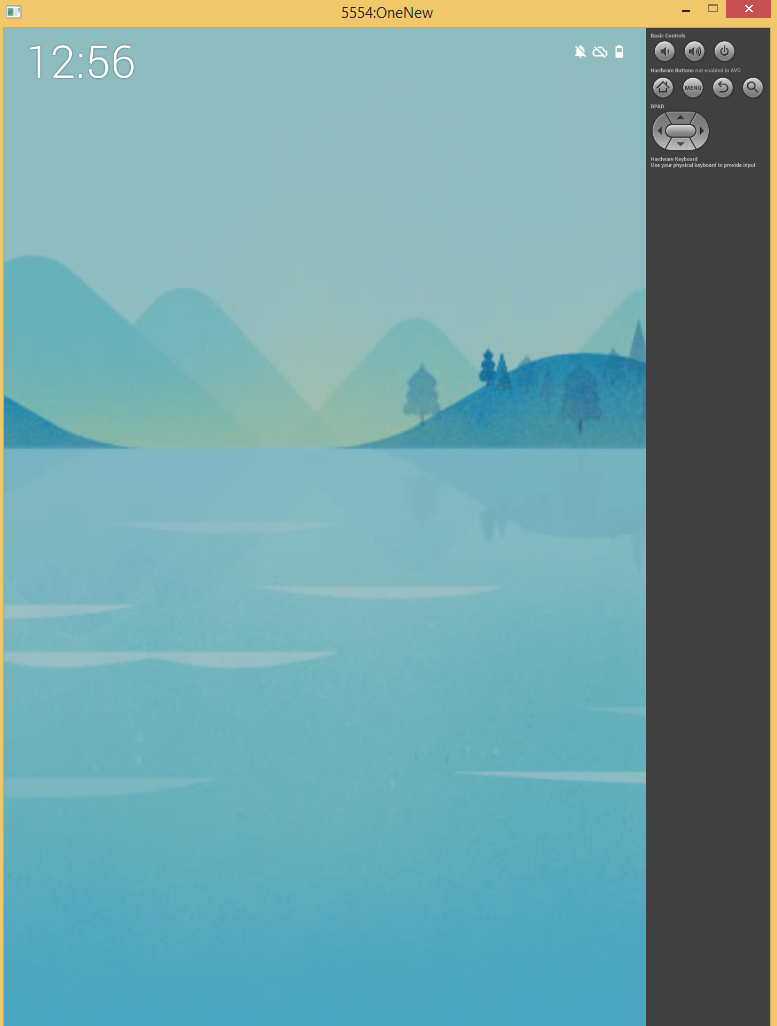
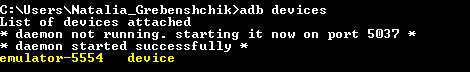
## **Test Application**

1. Android applications have ‘.apk’ extension.

## **Node JS**

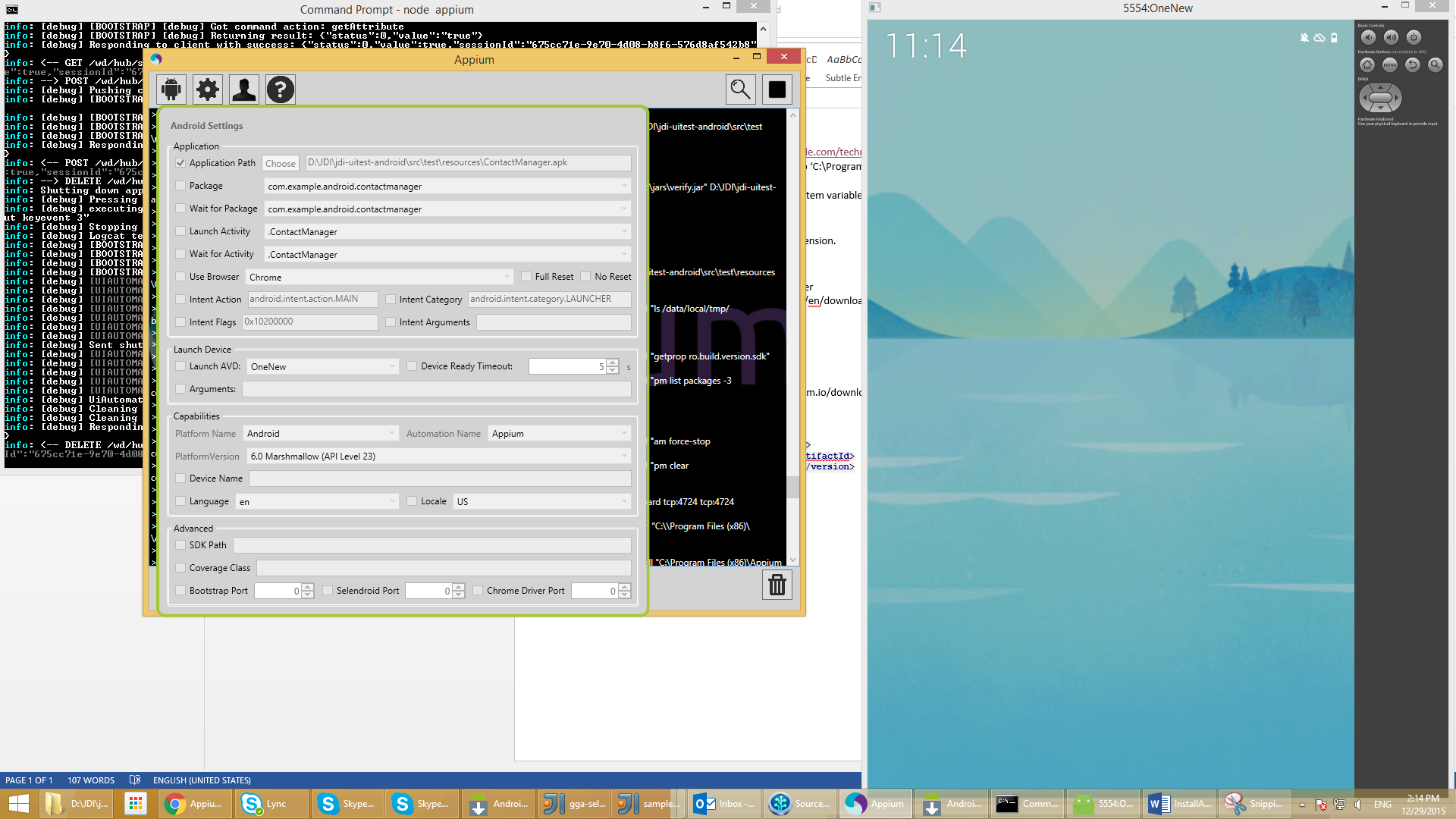
1. Node JS is used for start Appium server
2. Download it from [https://nodejs.org/en/download](https://nodejs.org/en/download/)

## **Android SDK**

1. Download Android SDK from <http://developer.android.com/intl/ru/sdk/index.html>
2. Install it.
3. Add system variables ANDROID\_HOME as ‘C:\Program Files (x86)\Android\android-sdk’ (or the path your Android SDK installed)
4. Append path variable with % ANDROID\_HOME%\platform-tools
5. Test that installation is fine:
   1. There is AVD installed  
      
   2. There is Android SDK manage installed. Install packages you need.
   3. Open cmd, type ‘adb’. You will see all available Android SDK’s command   
      
6. Create virtual android device on AVD by supplying device setting.  
   
7. Start android emulator from AVD Manager  
   
8. Verify emylator using ‘adb device’ command  
   

## **Appium Server**

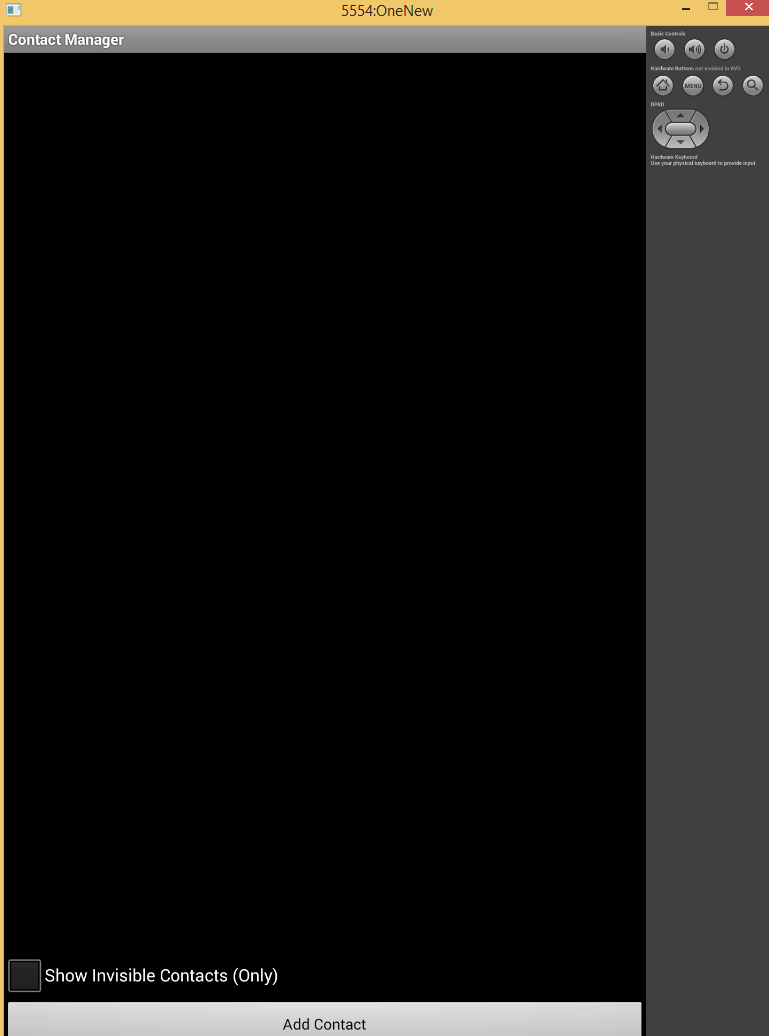
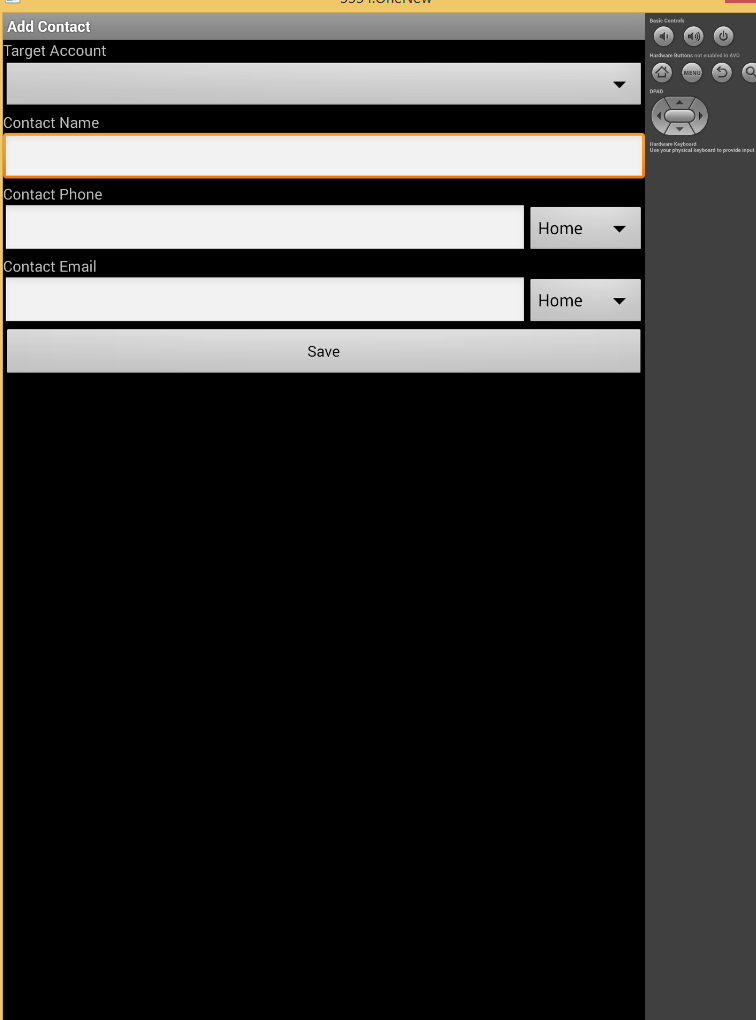
1. Download Appium from <http://appium.io/downloads.html>
2. Install locally. Program view is



1. It is used for elements location
2. Dependency for maven project is  
   <**dependency**>  
    <**groupId**>io.appium</**groupId**>  
    <**artifactId**>java-client</**artifactId**>  
    <**version**>${appium.version}</**version**>  
   </**dependency**>
3. To start Appium server
   1. open cmd and navigate ‘c:\Program Files (x86)\Appium\node\_modules\appium\bin’ dir (or path Appium is installed)
   2. execute ‘node appium’



**Create your first test with JDI**

1. Contact Manager application is used as example in this document. It has 2 pages – Start page and Add Contact page.  
    
2. Create @JApp pages list to support Cascad initialisation

@JApp  
**public class** EpamJDIAndroidApp **extends** Site {  
 @JPage(title = **"Add Contact"**)  
 **public static** AddContactForm *addContactForm*;  
 @JPage(title = **"Contacts"**)  
 **public static** StartPage *startPage*;  
}

1. Create test.properties with driver settings   
   **driver**=**android  
   timeout.wait.element**=**5  
   timeout.wait.pageLoad**=**5  
   appFilePath** = **jdi-uitest-android\\src\\test\\resources\\ContactManager.apk  
   deviceName** = **emulator-5554  
   platformVersion** = **4.4  
   appPackage** = **com.example.android.contactmanager  
   appActivity** = **.ContactManager**
2. Now it is good to create page objects classes
   1. Start page is AppPage  
        
      **public class** StartPage **extends** AppPage {  
       @FindBy(className = **"android.widget.Button"**)  
       **public** Button **addContact**;  
      }
   2. Add Contact page might be a Form (not a page), should have ContactDetails as init class

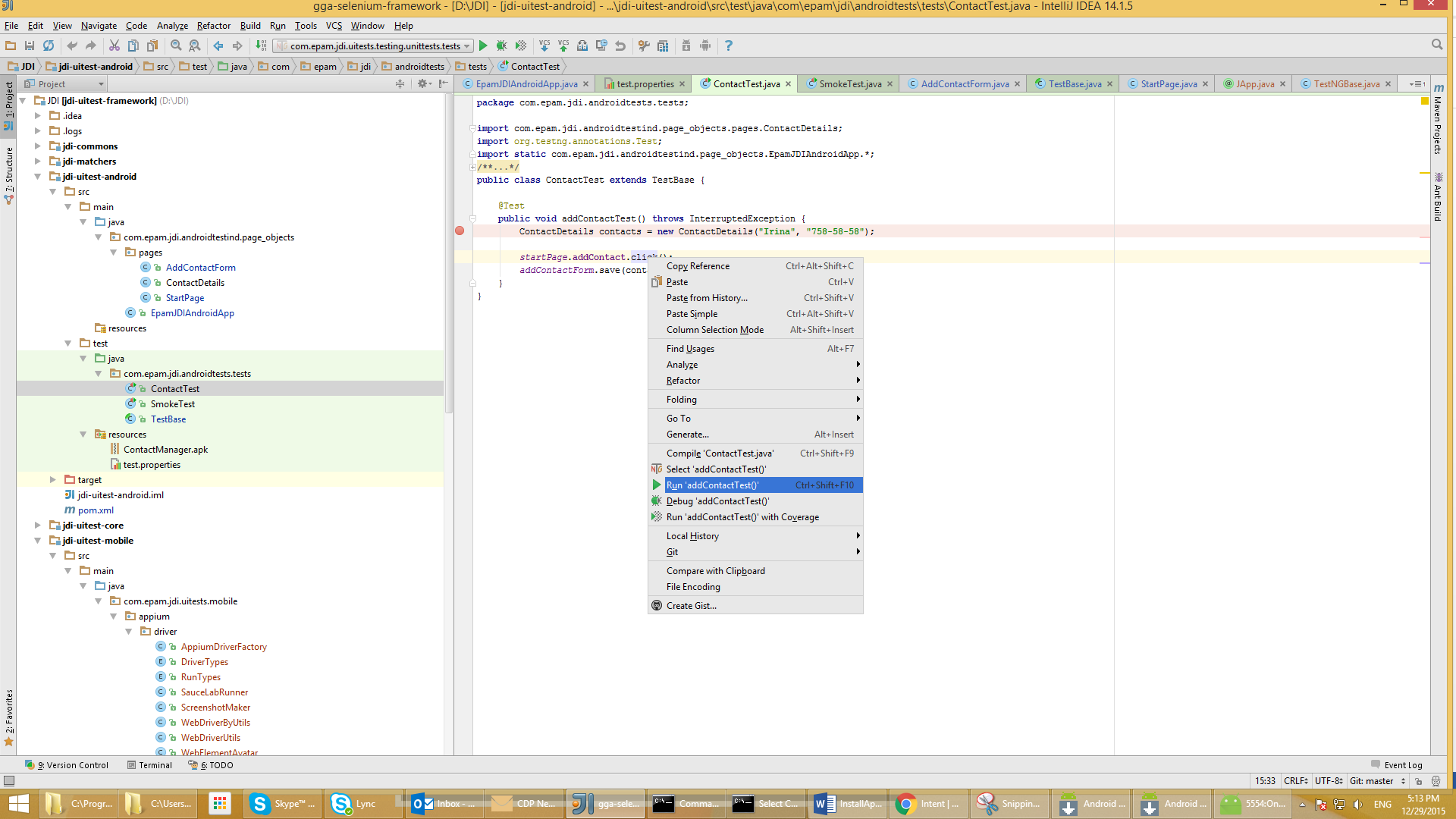
**public class** AddContactForm **extends** Form<ContactDetails>{  
  
 @FindBy (xpath = **"//\*[@\*='com.example.android.contactmanager:id/contactNameEditText']"**)  
 **public** TextField **contactName**;  
  
 @FindBy (xpath = **"//\*[@\*='com.example.android.contactmanager:id/contactPhoneEditText']"**)  
 **public** TextField **phone**;  
  
 @FindBy (className = **"android.widget.Button"**)  
 **public** Button **save**;  
  
}

1. Create Before and After methods for starting and closing driver

**public abstract class** TestBase **extends** TestNGBase {  
  
 @BeforeMethod  
 **public static void** setUp() **throws** Exception {  
 *Init*(EpamJDIAndroidApp.**class**);  
 *logger*.info(**"Run Tests"**);  
 }  
  
 @AfterMethod  
 **public static void** teaDown(){  
 ((AndroidDriver)*driverFactory*.getDriver()).quit();  
 }  
}

1. No everything is set to write test  
     
   **public class** ContactTest **extends** TestBase {  
     
    @Test  
    **public void** addContactTest() **throws** InterruptedException {  
    ContactDetails contacts = **new** ContactDetails(**"Irina"**, **"758-58-58"**);  
     
    *startPage*.**addContact**.click();  
    *addContactForm*.save(contacts);  
    }  
   }  
   Test does click ‘Add’ button on start page and populate & submit Contact form on ‘Add Contact’ page.
2. To run test verify you have:
   1. One emulator started



* 1. Appium server started  
       
     
  2. Run you test as testing  
     

1. There will be standard testing result  
   