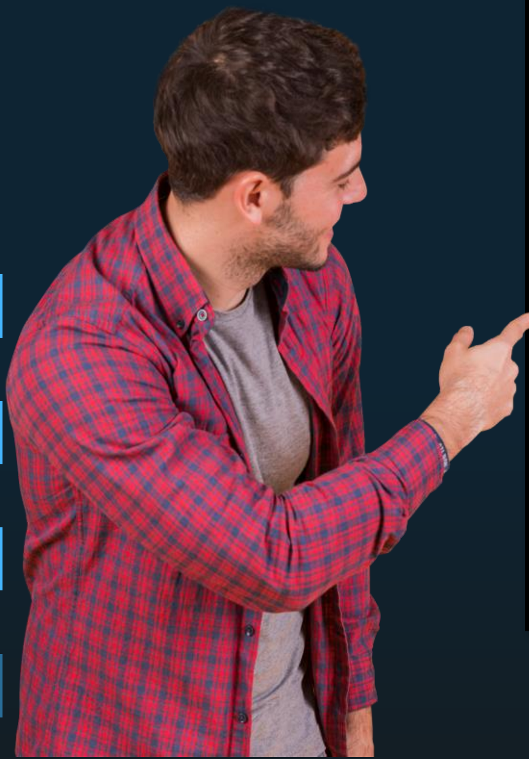
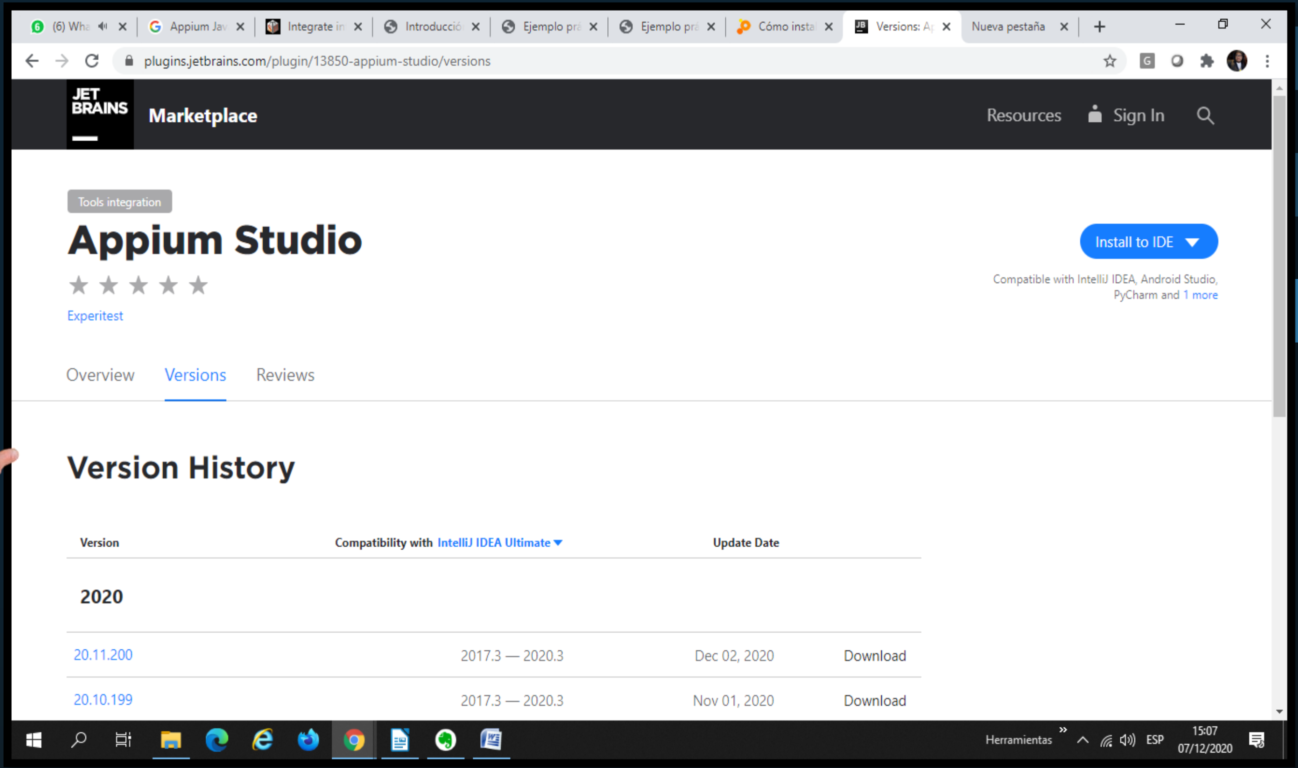
Appium Studio

Guia #1

Victor Portugal





Tools integration

Appium Studio

★ ★ ★ ★ ★

Experitest

Install to IDE ▾

Compatible with IntelliJ IDEA, Android Studio, PyCharm and 1 more

Overview Versions Reviews

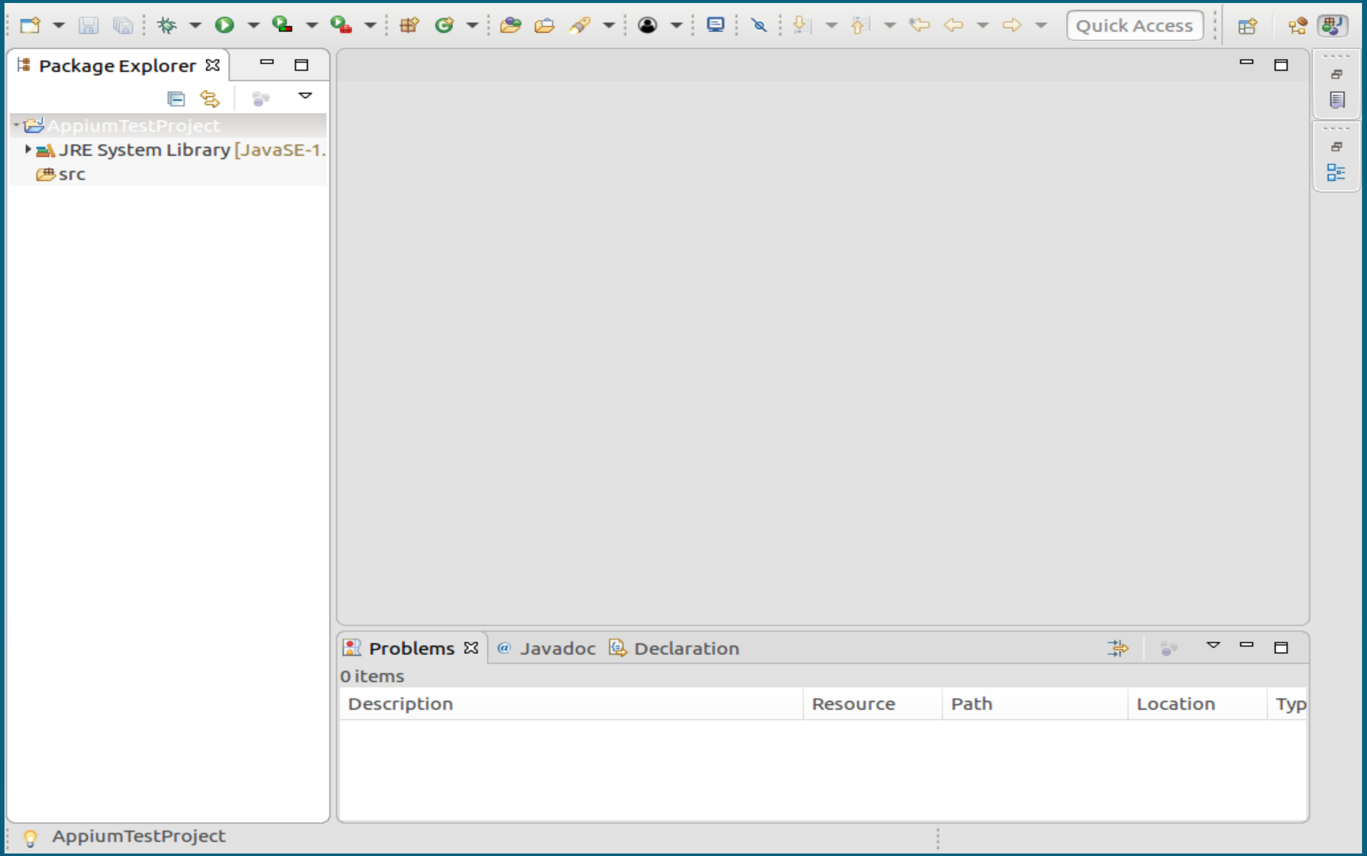
Version History

Version	Compatibility with IntelliJ IDEA Ultimate ▾	Update Date	
2020			
20.11.200	2017.3 — 2020.3	Dec 02, 2020	Download
20.10.199	2017.3 — 2020.3	Nov 01, 2020	Download

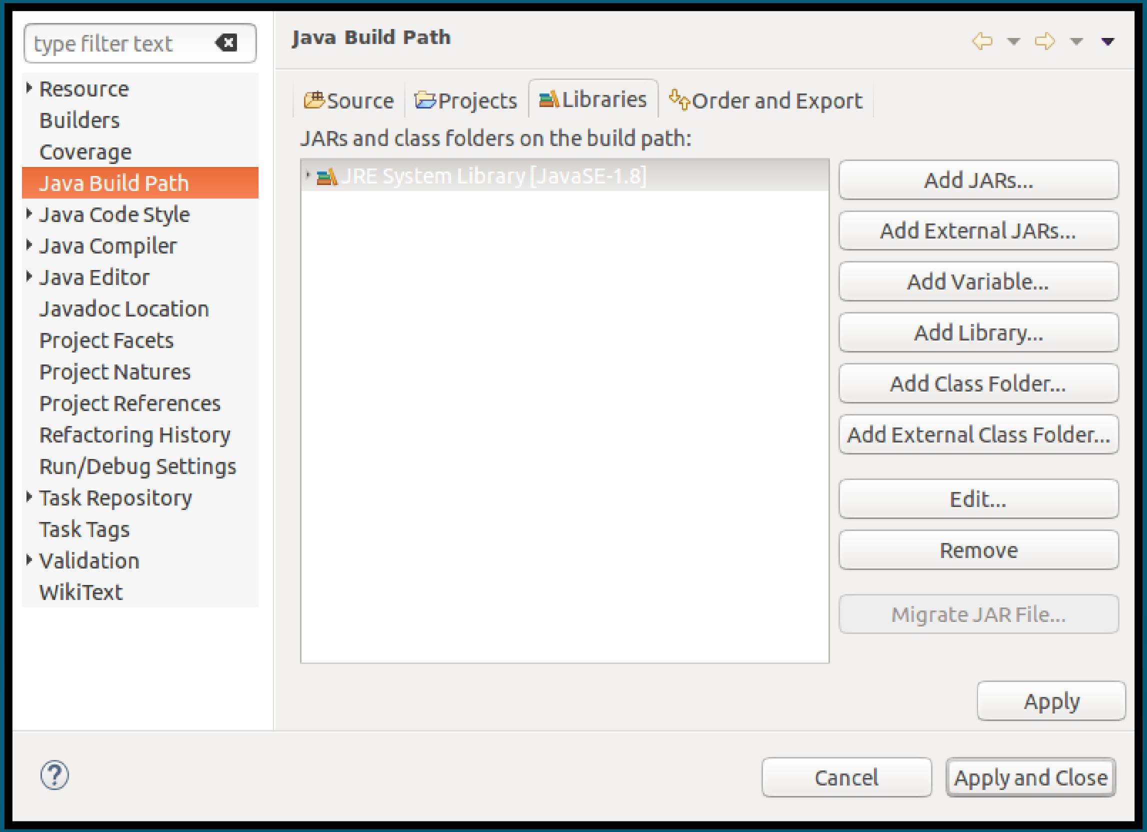
Configuración del entorno de desarrollo

Antes de empezar a preparar el entorno de desarrollo debemos descargar *Selenium Standalone Server* desde su página de descargas y *Appium Java Client* desde Maven.

Creamos un nuevo proyecto Java:



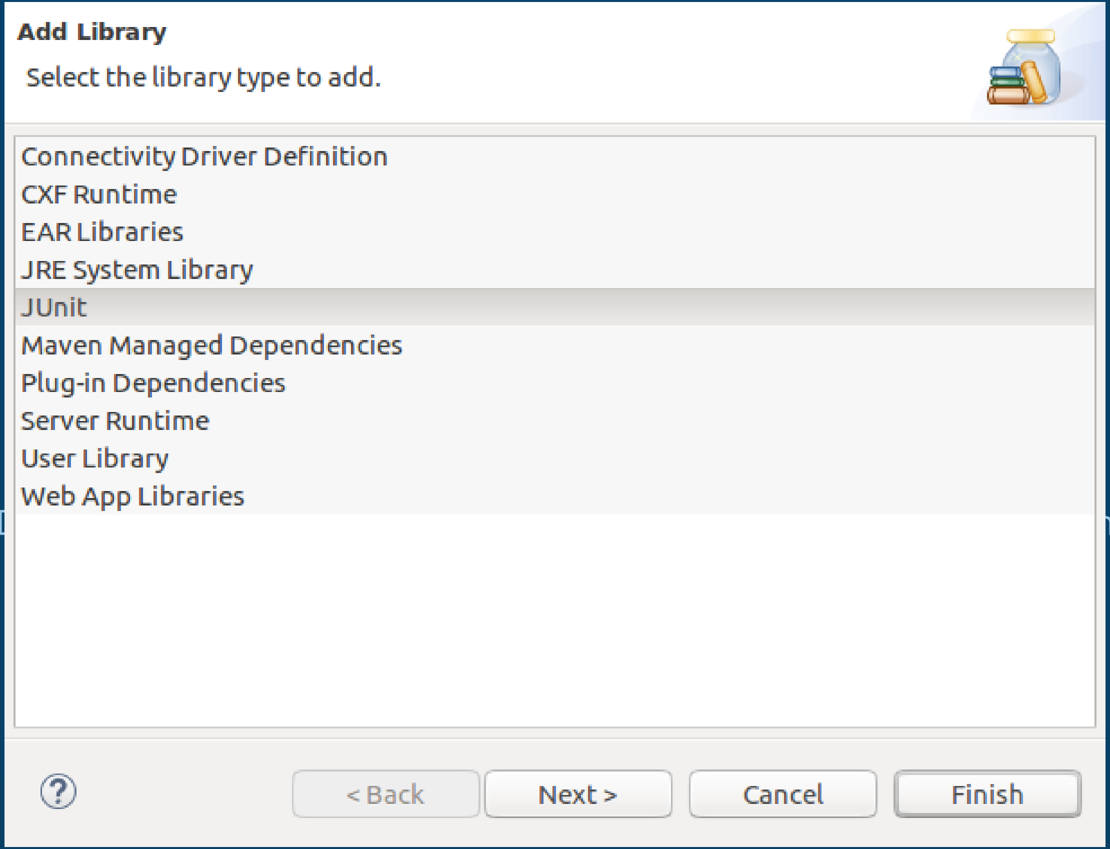
Accedemos a la configuración del *Build Path* haciendo *Click derecho sobre el proyecto > Build Path > Configure Build Path > Pestaña Libraries:*



JARs and class folders on the build path:

JRE System Library [JavaSE-1.8]

Añadimos los ficheros .jar de *Selenium Standalone Server* y *Appium Java Client* haciendo *Click en Add External JARs > Seleccionar los ficheros > Aceptar*. Necesitaremos también la biblioteca de *JUnit*, por lo que debemos hacer *Click en Add Library > Seleccionar JUnit > Next > Finish*.



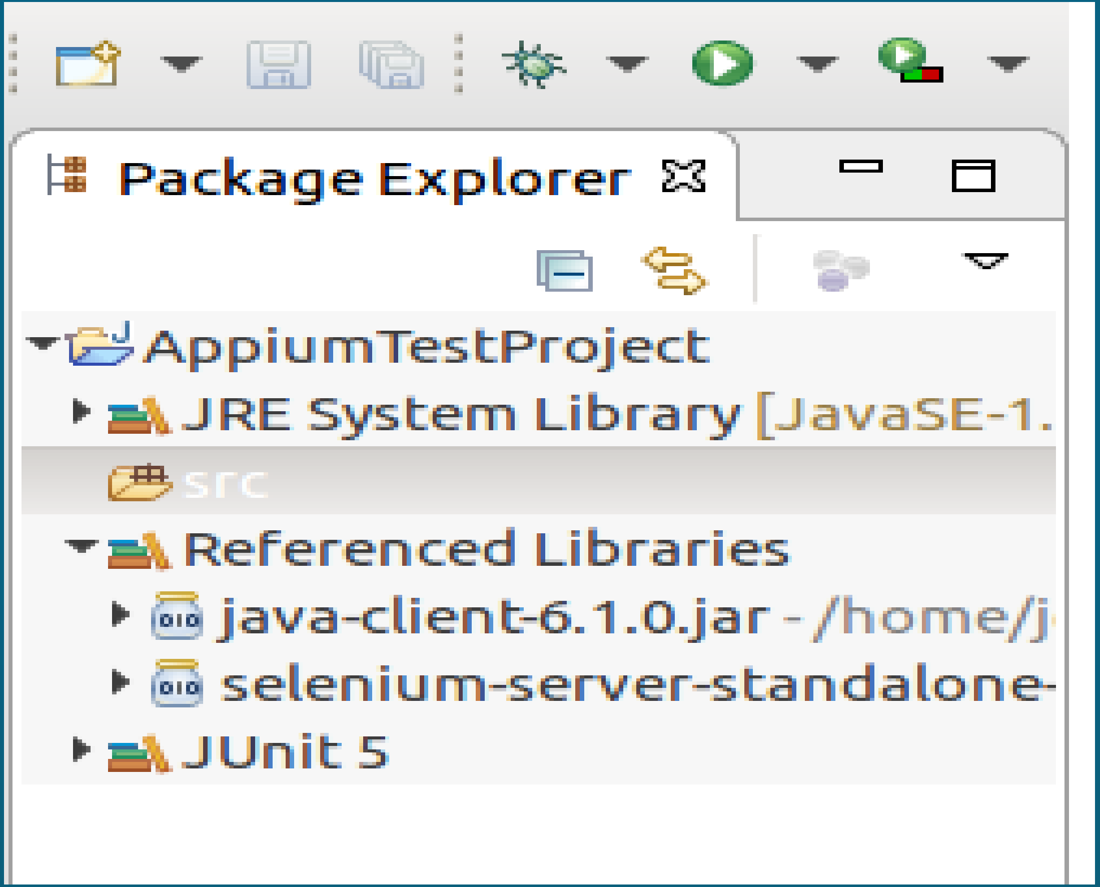
Add Library

Select the library type to add.

Connectivity Driver Definition
CXF Runtime
EAR Libraries
JRE System Library
JUnit
Maven Managed Dependencies
Plug-in Dependencies
Server Runtime
User Library
Web App Libraries

Para guardar todos los cambios hacemos *Click en Apply and Close*. El resultado en el proyecto debe verse como a continuación:

.



Package Explorer

AppiumTestProject

JRE System Library [JavaSE-1.

src

Referenced Libraries

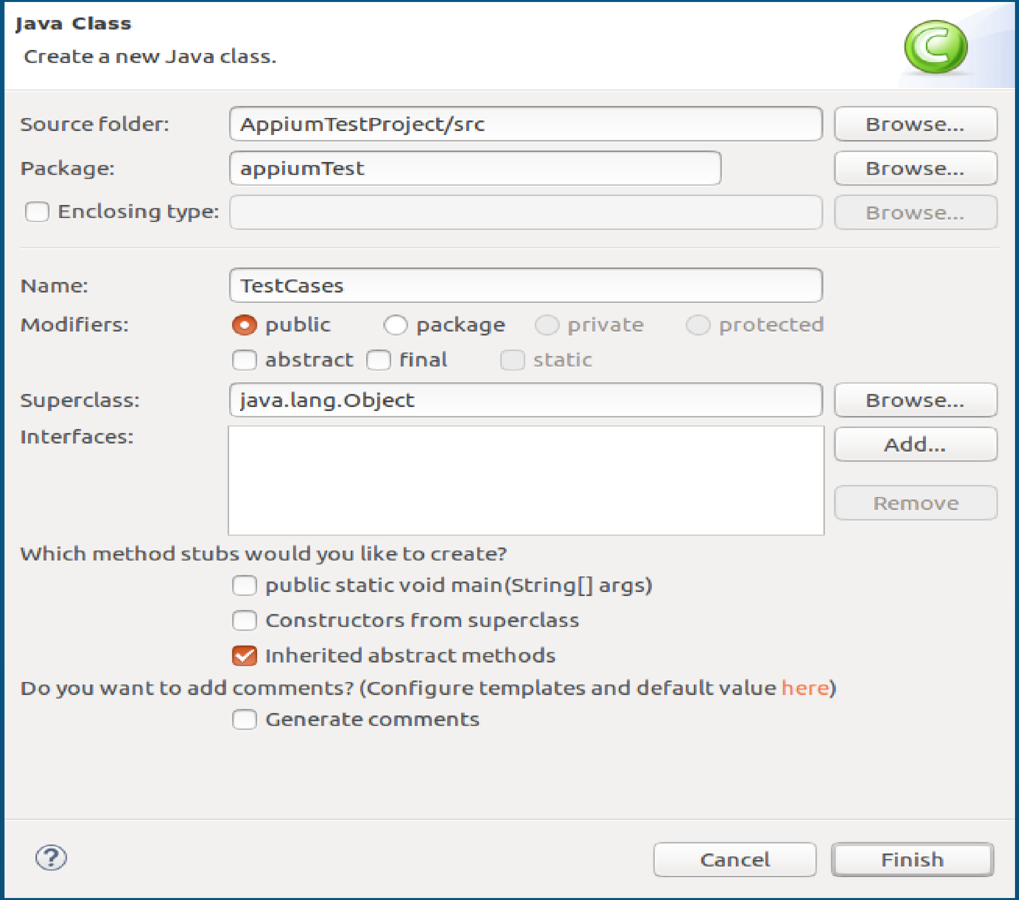
java-client-6.1.0.jar - /home/j

selenium-server-standalone-

JUnit 5

-Implementación de los casos de prueba

Una vez configurado el entorno, nos disponemos a implementar los casos de prueba. Creamos una nueva clase dentro de *src*:



Java Class

Create a new Java class.

Source folder: AppiumTestProject/src

Package: appiumTest

□ Enclosing type:

Name: TestCases

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

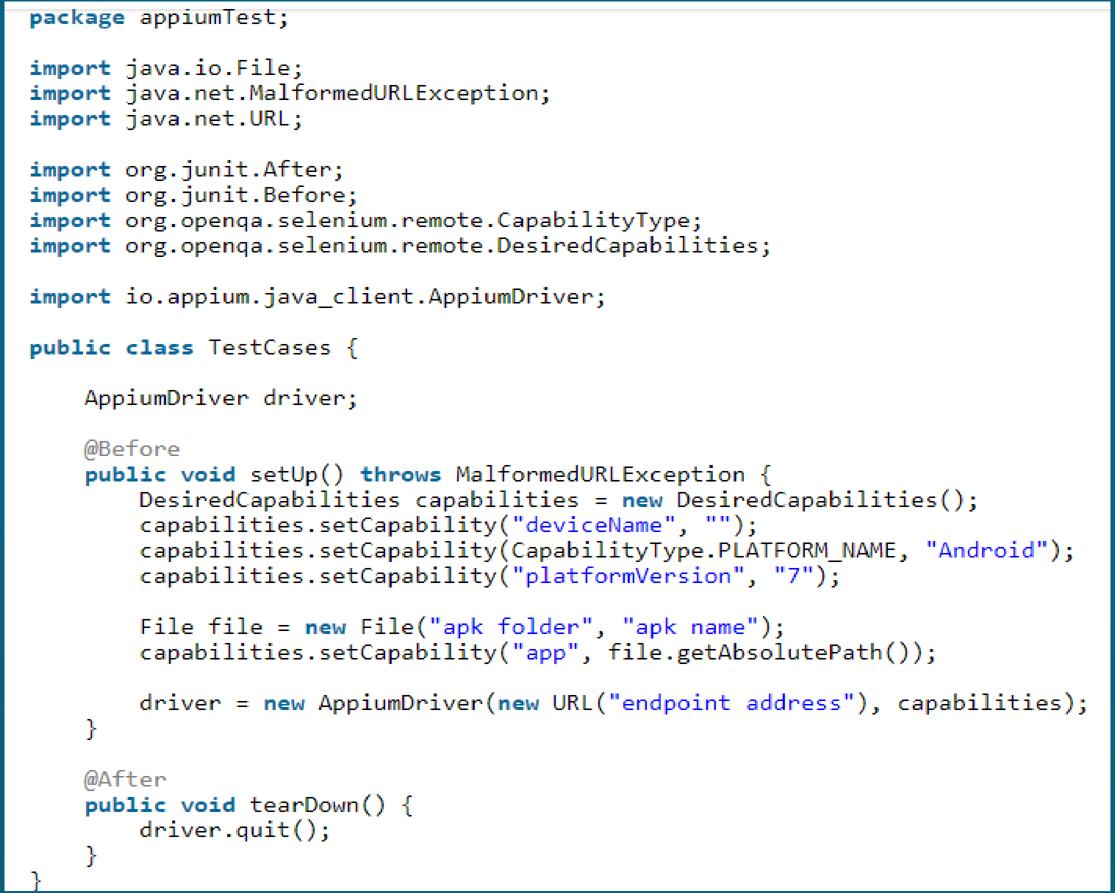
□ public static void main(String[] args)

□ Constructors from superclass

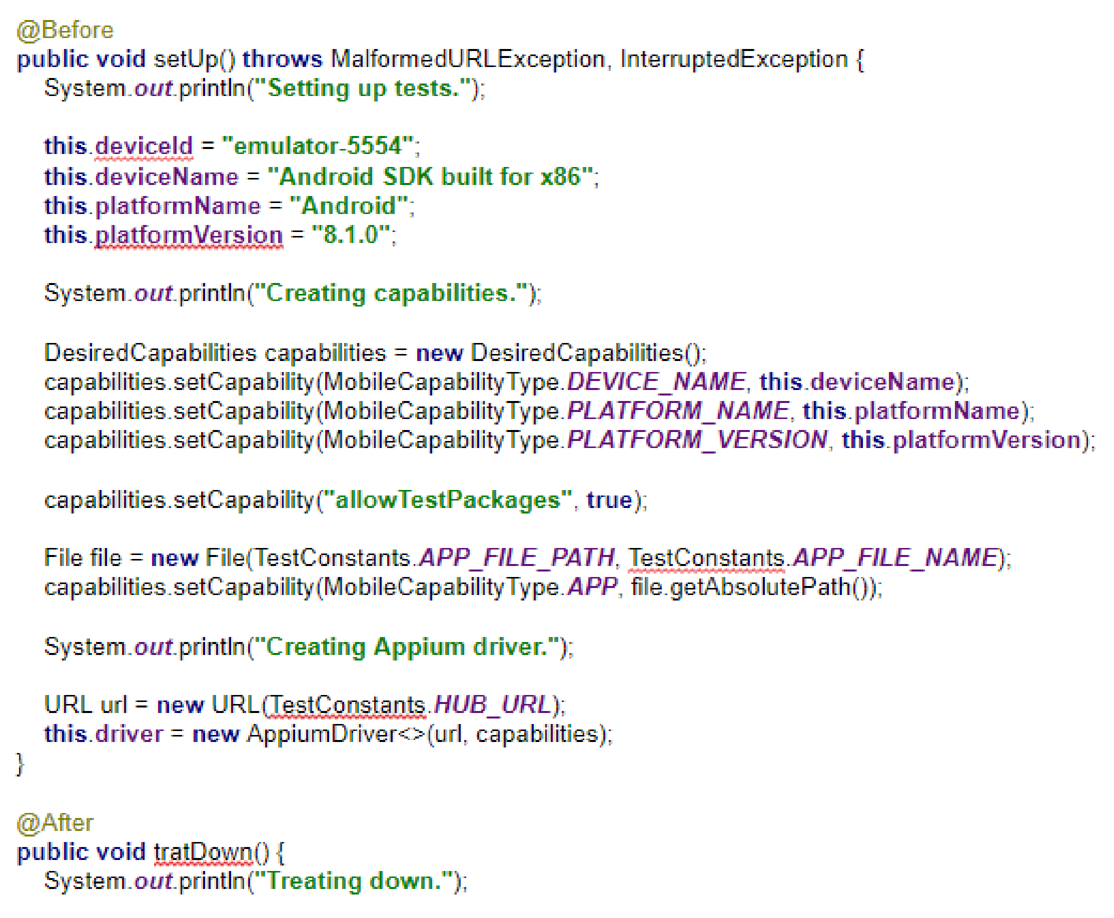
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

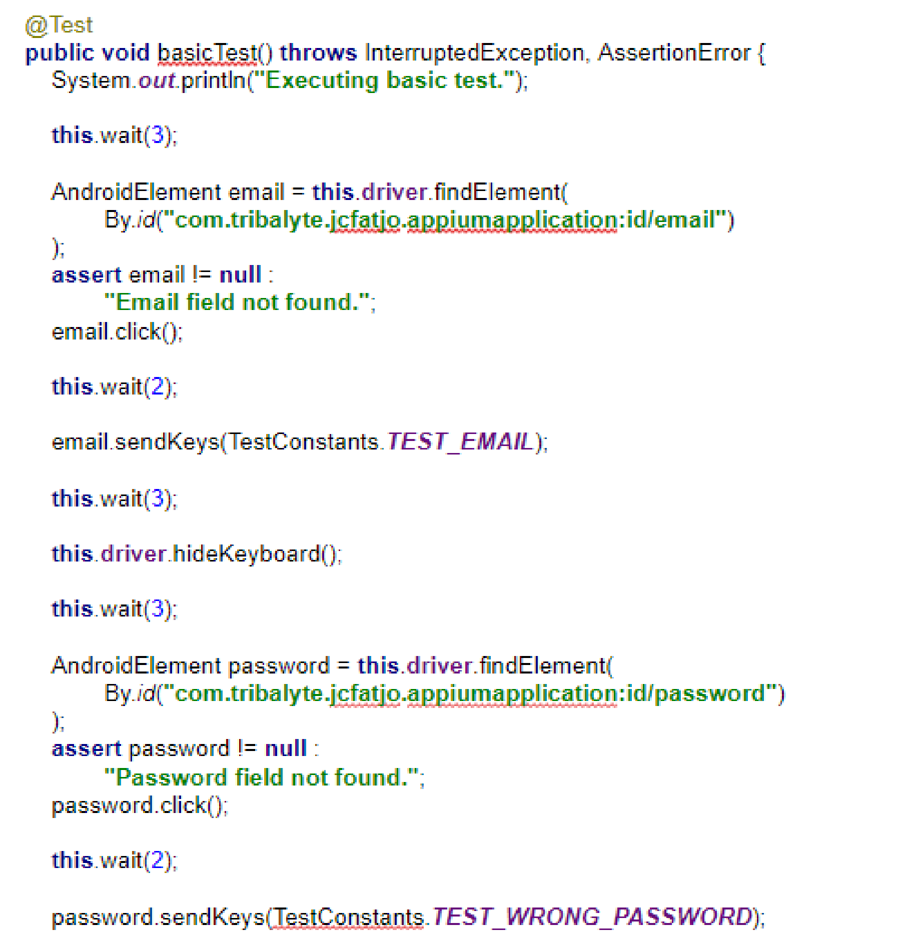
□ Generate comments



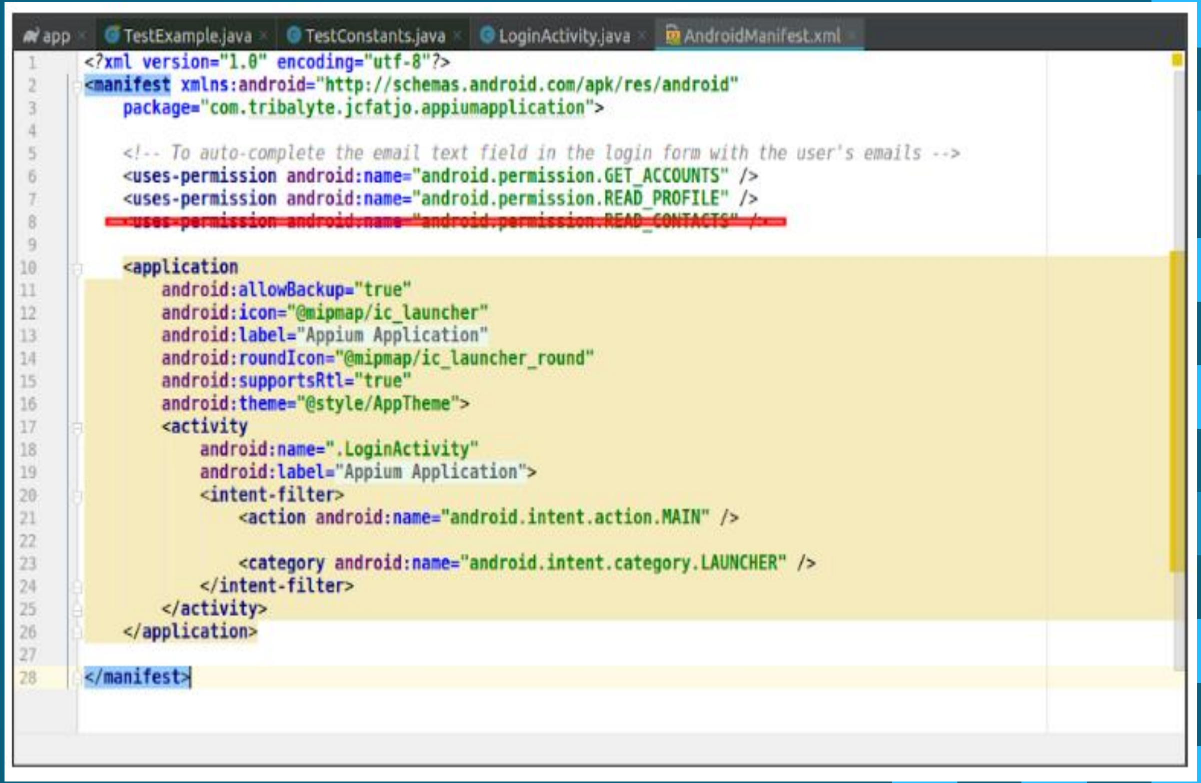
```java
package appiumTest;

import java.io.File;
import java.net.MalformedURLException;
import java.net.URL;

import org.junit.After;
import org.junit.Before;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.remote.DesiredCapabilities;

import io.appium.java_client.AppiumDriver;

public class TestCases {

    AppiumDriver driver;

    @Before
    public void setUp() throws MalformedURLException {
        DesiredCapabilities capabilities = new DesiredCapabilities();
        capabilities.setCapability("deviceName", "");
        capabilities.setCapability(CapabilityType.PLATFORM_NAME, "Android");
        capabilities.setCapability("platformVersion", "7");

        File file = new File("apk folder", "apk name");
        capabilities.setCapability("app", file.getAbsolutePath());

        driver = new AppiumDriver(new URL("endpoint address"), capabilities);
    }

    @After
    public void tearDown() {
        driver.quit();
    }
}
```



```
@Before
public void setUp() throws MalformedURLException, InterruptedException {
    System.out.println("Setting up tests.");

    this.deviceId = "emulator-5554";
    this.deviceName = "Android SDK built for x86";
    this.platformName = "Android";
    this.platformVersion = "8.1.0";

    System.out.println("Creating capabilities.");

    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities.setCapability(MobileCapabilityType.DEVICE_NAME, this.deviceName);
    capabilities.setCapability(MobileCapabilityType.PLATFORM_NAME, this.platformName);
    capabilities.setCapability(MobileCapabilityType.PLATFORM_VERSION, this.platformVersion);

    capabilities.setCapability("allowTestPackages", true);

    File file = new File(TestConstants.APP_FILE_PATH, TestConstants.APP_FILE_NAME);
    capabilities.setCapability(MobileCapabilityType.APP, file.getAbsolutePath());

    System.out.println("Creating Appium driver.");

    URL url = new URL(TestConstants.HUB_URL);
    this.driver = new AppiumDriver<>(url, capabilities);
}

@After
public void tratDown() {
    System.out.println("Treating down.");
```



```
@Test
public void basicTest() throws InterruptedException, AssertionError {
    System.out.println("Executing basic test.");

    this.wait(3);

    AndroidElement email = this.driver.findElement(
            By.id("com.tribalyte.jcfatjo.appiumapplication:id/email")
    );
    assert email != null :
            "Email field not found.";
    email.click();

    this.wait(2);

    email.sendKeys(TestConstants.TEST_EMAIL);

    this.wait(3);

    this.driver.hideKeyboard();

    this.wait(3);

    AndroidElement password = this.driver.findElement(
            By.id("com.tribalyte.jcfatjo.appiumapplication:id/password")
    );
    assert password != null :
            "Password field not found.";
    password.click();

    this.wait(2);

    password.sendKeys(TestConstants.TEST_WRONG_PASSWORD);
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tribalyte.jcfatjo.appiumapplication">

    <!-- To auto-complete the email text field in the login form with the user's emails -->
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.READ_PROFILE" />
    ~~<uses-permission android:name="android.permission.READ_CONTACTS" />~~

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Appium Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".LoginActivity"
            android:label="Appium Application">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```