

О JUnit 5 в контексте Android разработки

JUnit 5 был представлен в июле 2016 года, и данный релиз принес довольно много нововведений. Предыдущая версия - JUnit4 была стандартом для юнит-тестирования на Android еще с первых дней выхода этой ОС на мировой рынок. В данном материале речь пойдет об основах фреймворка, мы продемонстрируем как можно настроить его для дальнейшей комфортной работы с проектом на ОС Android. Кстати стоит отметить, обновленный фреймворк состоит сразу из нескольких различных по назначению компонентов - и это пожалуй самое главное отличие.

JUnit Platform – является основой данной для JUnit 5, ее функционал дает возможность без проблем запускать всевозможные тест-фреймворки JVM. Кстати, здесь есть Console Launcher с возможностью быстрого запуска JUnit 5 из командной строки. JUnit Jupiter - основа платформы. Дает массу возможностей для написания всевозможных решений, создания тестов. И конечно же не обошлось без JUnit Vintage - это фреймворк разработанный специально для выполнения тестов из предыдущих версий. Так вы сможете без проблем запустить на нем все тесты для JUnit4 и даже JUnit 3. Эта троица (JUnit Platform, JUnit Jupiter и JUnit Vintage) и являет собой JUnit 5. Хотя платформа и претерпела довольно существенные изменения, но тесты написанные для JUnit4 по-прежнему работают без нареканий.

Далее речь пойдет о настройке JUnit 5 под ОС Android. И пожалуй начать стоит с официальной документации - (<https://junit.org/junit5/docs/current/user-guide>). Так в секции 4.2.1 довольно подробно описано как произвести настройку JUnit 5 для Gradle. В том случае, если четко следовать всем предписаниям, взять и просто добавить плагин Gradle для JUnit 5, то такая сборка будет постоянно вылетать, естественно о нормальной работе в данном случае не может быть и речи:

```

0leksii-MBP:JUnit5 oleksii$ ./gradlew Test
Starting a Gradle Daemon, 1 incompatible Daemon could not be reused, use --status for details

FAILURE: Build failed with an exception.

* Where:
Build file '/Users/oleksii/AndroidProjects/JUnit5/build.gradle' line: 30

* What went wrong:
A problem occurred evaluating root project 'JUnit5'.
> Could not find method test() for arguments [build_ajjd6vyn42ssdk9lk18k3yf6k$_run_closure3@7d05e69e] on root project 'JUnit5' of type org.gradle.api.Project.

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 4s

```

Причина проста - JUnit 5 стабильно функционирует исключительно с Gradle под Java, а он определяет иную иерархию задач нежели например плагин Gradle для Android. В том случае если в своем проекте мы будем работать с чистой версией Java, то эта настройка будет работать стабильно, но при запуске JUnit-тестов уже на Android-проекте, сбои в работе возникнут сразу же. Поэтому JUnit 5, в чистом виде, для дальнейшей работы с ОС Android не подходит. Но решение этой проблемы уже есть. Достаточно убрать все сделанные ранее настройки, а затем просто создать уже новые - это позволит плагину работать как компоненту системы:

```

buildscript {
    dependencies {
        classpath "de.mannodermaus.gradle.plugins:android-junit5:1.2.0.0"
    }
}

apply plugin: "de.mannodermaus.android-junit5"
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'

    // (Required) Writing and executing Unit Tests on the JUnit Platform
    testImplementation "org.junit.jupiter:junit-jupiter-api:5.2.0"
    testRuntimeOnly "org.junit.jupiter:junit-jupiter-engine:5.2.0"

    // (Optional) If you also have JUnit 4-based tests
    testImplementation "junit:junit:4.12"
    testRuntimeOnly "org.junit.vintage:junit-vintage-engine:5.2.0"
    testImplementation("org.assertj:assertj-core:3.11.1")
}

```

Если все сделать правильно, то JUnit 5 заработает, а значит в дальнейшем появится возможность использовать его для создания тестов. К слову, все же, требуется учитывать зависимость JUnit 4.12, поскольку без него test runner Android Studio попросту не запустится.

Для проверки работоспособности напишем сразу несколько тестов, например «Hello world!» - один на JUnit 4:

```
import org.junit.Test;
import static org.assertj.core.api.Assertions.assertThat;

public class JUnit4Test {
    @Test
    public void test() {
        assertThat( actual: true).isTrue();
    }
}
```

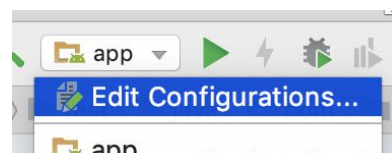
Второй уже на новой версии - JUnit 5:

```
import org.junit.jupiter.api.Test;
import static org.assertj.core.api.Assertions.assertThat;

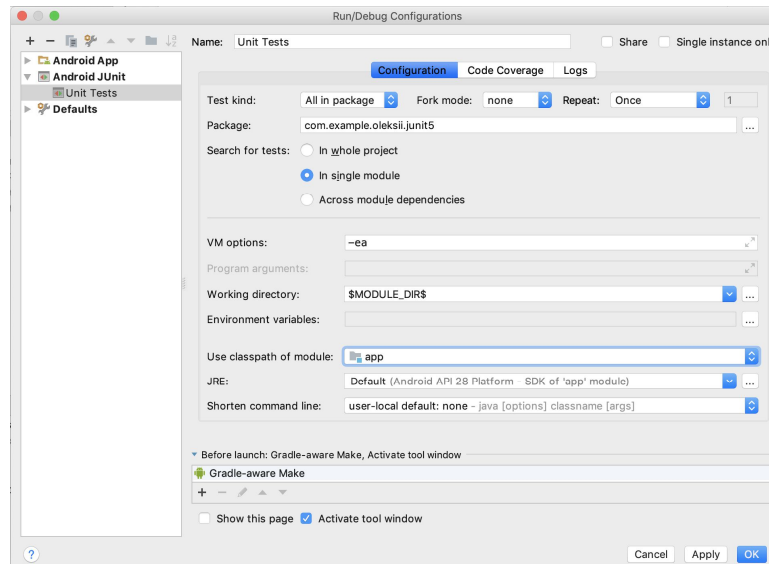
public class JUnit5Test {

    @Test
    public void test() {
        assertThat( actual: true).isTrue();
    }
}
```

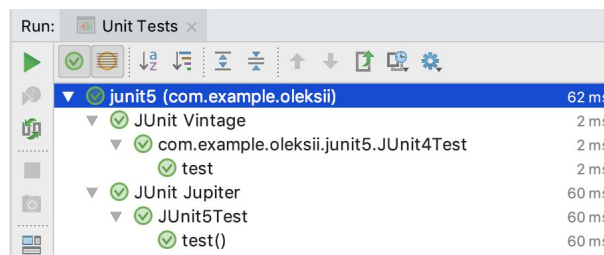
Как видим, не считая выделенной строки отличий между этими двумя тестами нет. Можем приступить к конфигурации исполнителя тестов в Android Studio. Для этого, сначала выбираем «Edit Configurations...» в размещенном на панели инструментов списке:



Создаем конфигурацию Android JUnit:



При запуске подобной конфигурации все тесты должны работать стабильно:



Упоминания заслуживает тот факт, что этот тест написан на JUnit 4, выполнялся на JUnit Vintage, в случае с тестом написанным на JUnit 5 - уже на JUnit Jupiter. В целом, никаких проблем дальше не возникнет, единственное что могут быть проблемы при запуске через командную строку:

```
:app:compileReleaseUnitTestSources
:app:assembleReleaseUnitTest
:app:junitPlatformTestRelease
Feb 18, 2017 4:46:04 PM org.junit.platform.launcher.core.ServiceLoaderTestEngineRegistry loadTestEngines
INFO: Discovered TestEngines with IDs: [junit-jupiter]

Test run finished after 83 ms
[ 1 tests found ]
[ 0 tests skipped ]
[ 1 tests started ]
[ 0 tests aborted ]
[ 1 tests successful ]
[ 0 tests failed ]
[ 0 containers failed ]

:app:testDebugUnitTest
:app:testReleaseUnitTest
:app:test SKIPPED

BUILD SUCCESSFUL

Total time: 6.197 secs
- JUnit5
```

В отчете мы видим что выполнялся лишь один тест, хотя по сути создано было два. Дело в том, что в ходе проведения тестов обнаружен был только junit-jupiter. Избежать этой проблемы довольно просто, достаточно добавить junit-vintage в build.gradle файл:

```
testRuntimeOnly "org.junit.vintage:junit-vintage-engine:5.2.0"
```

При запуске с Android Studio все тесты будут функционировать стабильно, плюс, теперь вы сможете спокойно работать через командную строку:

```
:app:compileReleaseUnitTestSources
:app:assembleReleaseUnitTest
:app:junitPlatformTestRelease
Feb 18, 2017 4:48:11 PM org.junit.platform.launcher.core.ServiceLoaderTestEngineRegistry loadTestEngines
INFO: Discovered TestEngines with IDs: [junit-jupiter, junit-vintage]

Test run finished after 96 ms
[      2 tests found      ]
[      0 tests skipped   ]
[      2 tests started   ]
[      0 tests aborted   ]
[      2 tests successful ]
[      0 tests failed    ]
[      0 containers failed]

:app:testDebugUnitTest
:app:testReleaseUnitTest
:app:test SKIPPED

BUILD SUCCESSFUL

Total time: 5.756 secs
  JUnit5
```

По итогу, в данном материале представлены основные моменты работы с JUnit 5. Кратко перечислены проблемы которые могут возникнуть во время работы и способы их исправления.

Литература:

1. JUnit 5 // JUnit 5. URL: <https://junit.org/junit5/> (дата обращения: 20.10.2018).
2. JUnit 5: Getting Started // Styling Android. URL: <https://blog.stylingandroid.com/junit-5-getting-started/> (дата обращения: 20.10.2018).