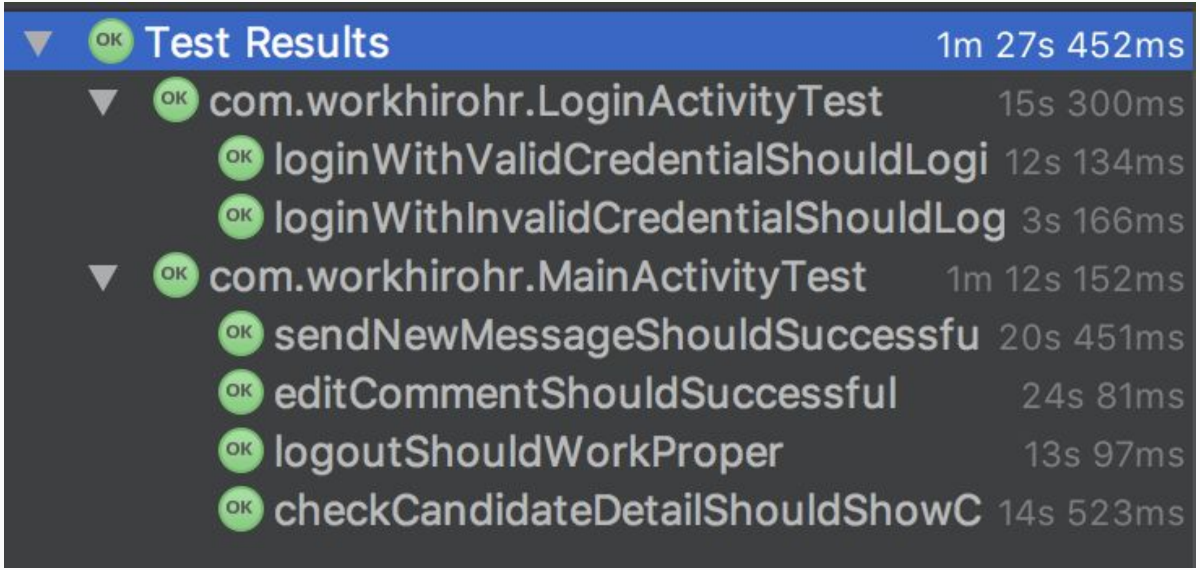
UI Testing with Kakao (vol.2)

Patipon

Growth Session #11 - January 18-19 2018

Achievements and progress

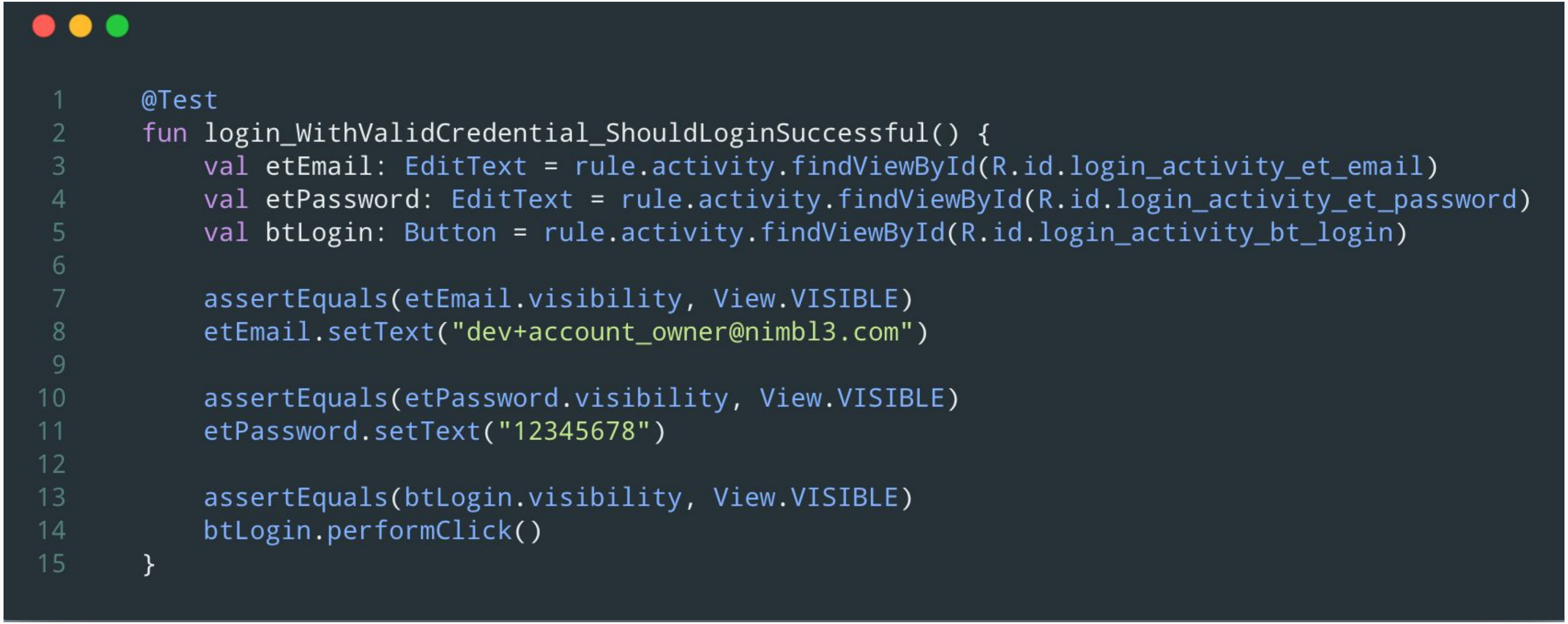
- Fix clearing of data between each test case
- Write more test cases
    - Edit message
    - New comment
    - See candidate detail
    - Logout



```
▼ OK Test Results                                        1m 27s 452ms
    ▼ OK com.workhirohr.LoginActivityTest                  15s 300ms
           OK loginWithValidCredentialShouldLogi           12s 134ms
           OK loginWithInvalidCredentialShouldLog           3s 166ms
    ▼ OK com.workhirohr.MainActivityTest                 1m 12s 152ms
           OK sendNewMessageShouldSuccessfu                20s 451ms
           OK editCommentShouldSuccessful                  24s 81ms
           OK logoutShouldWorkProper                       13s 97ms
           OK checkCandidateDetailShouldShowC              14s 523ms
```

Compare with default way

Instrumentation UI testing in with standard way:

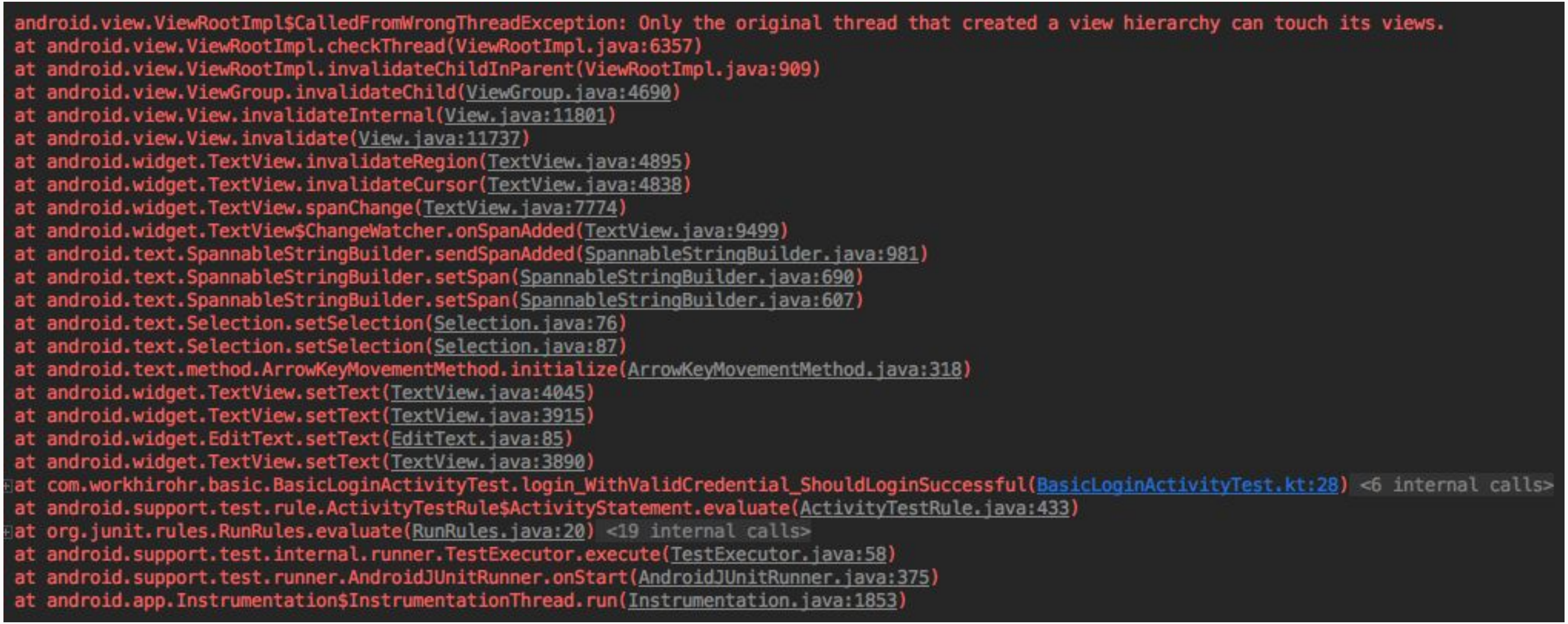


```
@Test
fun login_WithValidCredential_ShouldLoginSuccessful() {
    val etEmail: EditText = rule.activity.findViewById(R.id.login_activity_et_email)
    val etPassword: EditText = rule.activity.findViewById(R.id.login_activity_et_password)
    val btLogin: Button = rule.activity.findViewById(R.id.login_activity_bt_login)

    assertEquals(etEmail.visibility, View.VISIBLE)
    etEmail.setText("dev+account_owner@nimbl3.com")

    assertEquals(etPassword.visibility, View.VISIBLE)
    etPassword.setText("12345678")

    assertEquals(btLogin.visibility, View.VISIBLE)
    btLogin.performClick()
}
```

Compare with default way

⇒ Then run it

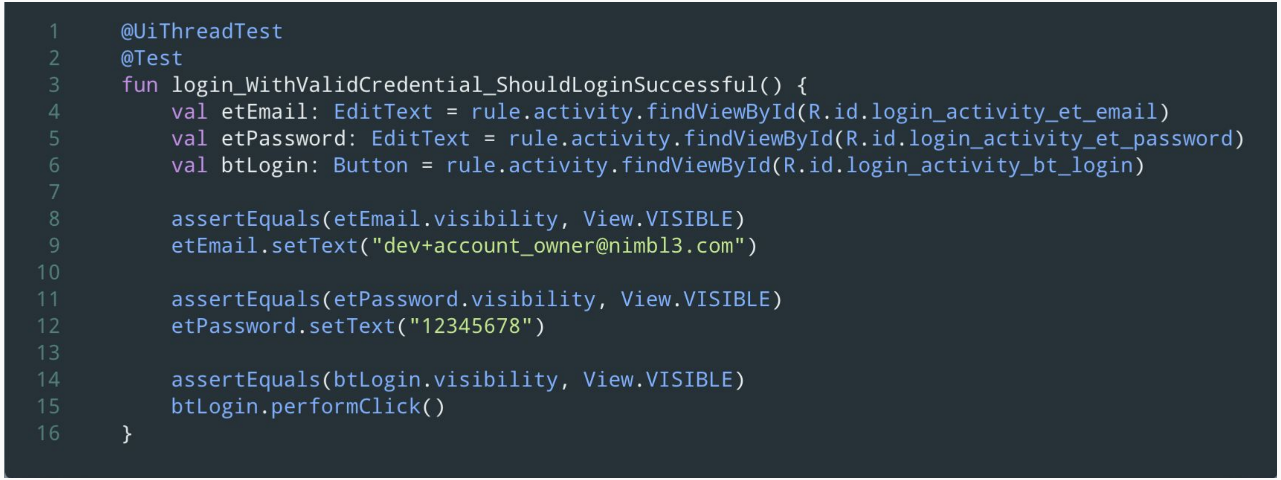
Compare with default way



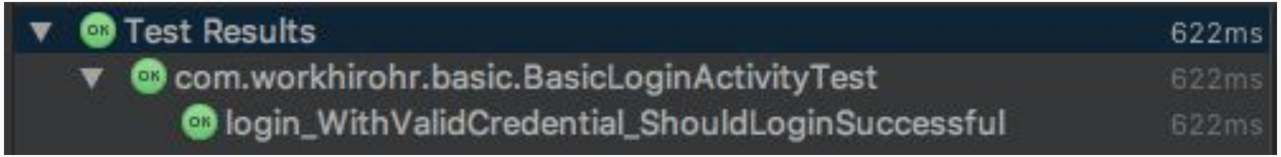
```
android.view.ViewRootImpl$CalledFromWrongThreadException: Only the original thread that created a view hierarchy can touch its views.
at android.view.ViewRootImpl.checkThread(ViewRootImpl.java:6357)
at android.view.ViewRootImpl.invalidateChildInParent(ViewRootImpl.java:909)
at android.view.ViewGroup.invalidateChild(ViewGroup.java:4690)
at android.view.View.invalidateInternal(View.java:11801)
at android.view.View.invalidate(View.java:11737)
at android.widget.TextView.invalidateRegion(TextView.java:4895)
at android.widget.TextView.invalidateCursor(TextView.java:4838)
at android.widget.TextView.spanChange(TextView.java:7774)
at android.widget.TextView$ChangeWatcher.onSpanAdded(TextView.java:9499)
at android.text.SpannableStringBuilder.sendSpanAdded(SpannableStringBuilder.java:981)
at android.text.SpannableStringBuilder.setSpan(SpannableStringBuilder.java:690)
at android.text.SpannableStringBuilder.setSpan(SpannableStringBuilder.java:607)
at android.text.Selection.setSelection(Selection.java:76)
at android.text.Selection.setSelection(Selection.java:87)
at android.text.method.ArrowKeyMovementMethod.initialize(ArrowKeyMovementMethod.java:318)
at android.widget.TextView.setText(TextView.java:4045)
at android.widget.TextView.setText(TextView.java:3915)
at android.widget.EditText.setText(EditText.java:85)
at android.widget.TextView.setText(TextView.java:3890)
at com.workhirohr.basic.BasicLoginActivityTest.login_WithValidCredential_ShouldLoginSuccessful(BasicLoginActivityTest.kt:28) <6 internal calls>
at android.support.test.rule.ActivityTestRule$ActivityStatement.evaluate(ActivityTestRule.java:433)
at org.junit.rules.RunRules.evaluate(RunRules.java:20) <19 internal calls>
at android.support.test.internal.runner.TestExecutor.execute(TestExecutor.java:58)
at android.support.test.runner.AndroidJUnitRunner.onStart(AndroidJUnitRunner.java:375)
at android.app.Instrumentation$InstrumentationThread.run(Instrumentation.java:1853)
```

Compare with default way

If you want interact with any View you need to add `@UiThreadTest` to switch to the UI thread



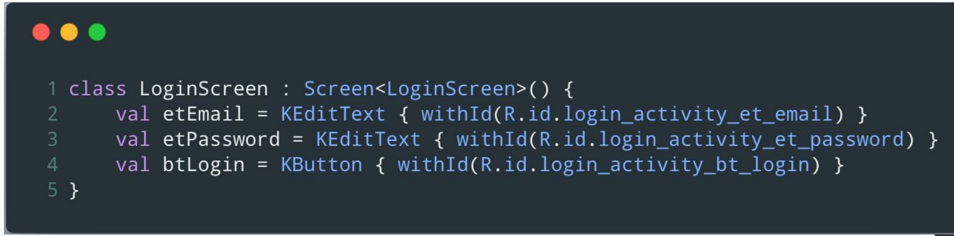
```
@UiThreadTest
@Test
fun login_WithValidCredential_ShouldLoginSuccessful() {
    val etEmail: EditText = rule.activity.findViewById(R.id.login_activity_et_email)
    val etPassword: EditText = rule.activity.findViewById(R.id.login_activity_et_password)
    val btLogin: Button = rule.activity.findViewById(R.id.login_activity_bt_login)

    assertEquals(etEmail.visibility, View.VISIBLE)
    etEmail.setText("dev+account_owner@nimbl3.com")

    assertEquals(etPassword.visibility, View.VISIBLE)
    etPassword.setText("12345678")

    assertEquals(btLogin.visibility, View.VISIBLE)
    btLogin.performClick()
}
```



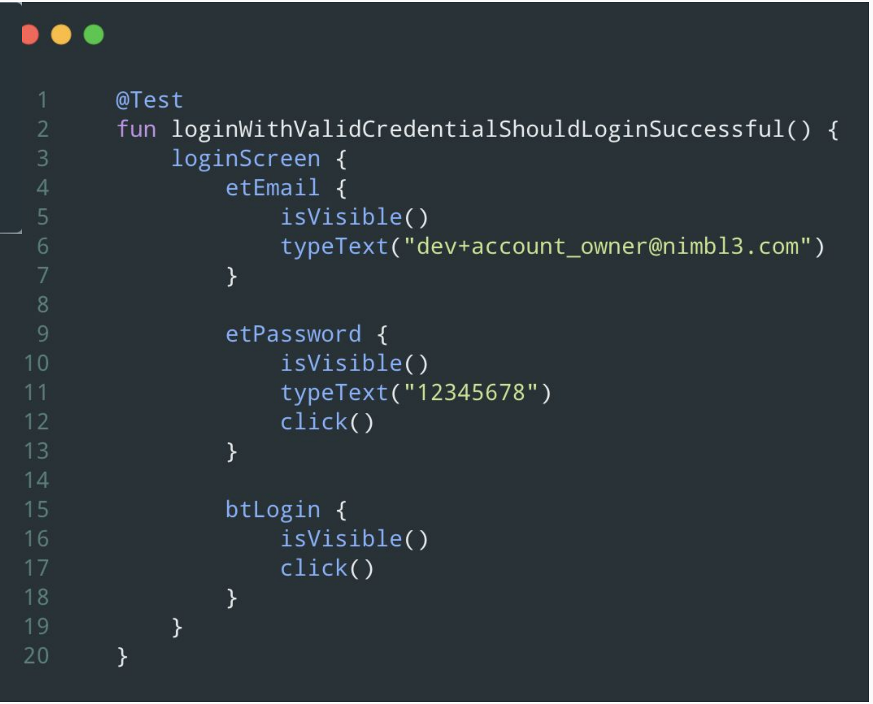
```
▼ OK Test Results                                              622ms
    ▼ OK com.workhirohr.basic.BasicLoginActivityTest           622ms
        OK login_WithValidCredential_ShouldLoginSuccessful     622ms
```

Now with Kakao

Just need to Declare Screen and use it



```kotlin
class LoginScreen : Screen<LoginScreen>() {
    val etEmail = KEditText { withId(R.id.login_activity_et_email) }
    val etPassword = KEditText { withId(R.id.login_activity_et_password) }
    val btLogin = KButton { withId(R.id.login_activity_bt_login) }
}
```



```kotlin
@Test
fun loginWithValidCredentialShouldLoginSuccessful() {
    loginScreen {
        etEmail {
            isVisible()
            typeText("dev+account_owner@nimbl3.com")
        }

        etPassword {
            isVisible()
            typeText("12345678")
            click()
        }

        btLogin {
            isVisible()
            click()
        }
    }
}
```

Now with Kakao

⇒ And then...

Summary

Pros

- Easy to use
- More readable (with page object pattern)

Cons

Need to have knowledge about Instrumented tests and Espresso Library before using this library

No sacrifice,
No victory.

- Sam Witwicky

Thanks!

Contact Nimbl3

hello@nimbl3.com

399 Sukhumvit Road, Interchange 21
Klongtoey nua, Wattana
Bangkok 10110

20th Floor, Central Tower
28 Queen's Road
Central, Hong Kong

nimbl3.com