

SEG2105 – Tutorial

Android Testing and Espresso Framework.

Create an Application

Required app has a login activity consisting of

- User First name
- Last Name
- Email ID
- Password
- Submit button
- Message

Download the app from the following link

<https://drive.google.com/open?id=10g1B2TpW0T5Vpv0ky1aR6ChP-tGB0uF5>

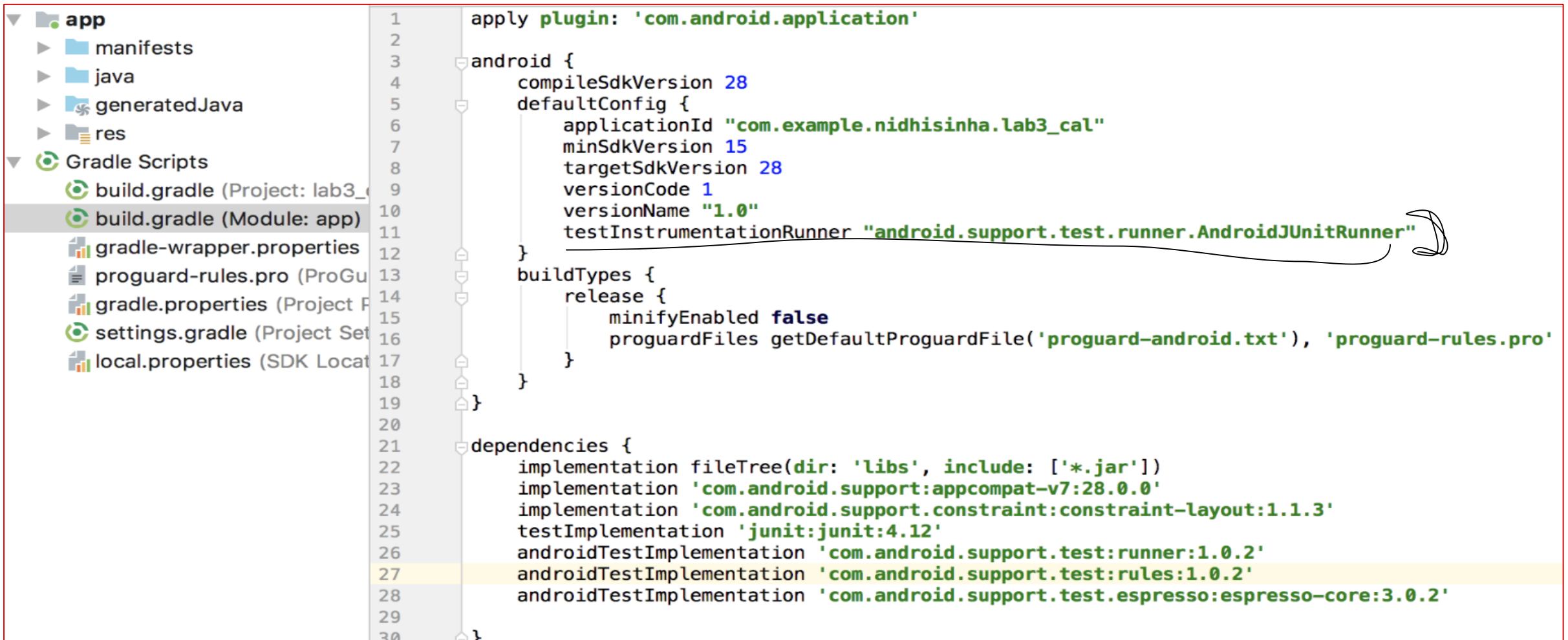
Exercise

- Write a JUnit test case that validate the name, lastname and password
- Write a espresso test case where we will validate the when the User types invalid name, textview should contain the text "Name is invalid"
- Write a espresso test case where we will validate the when the User types invalid last name, textview should contain the text "Last name is invalid"

Once the app is build, add the dependencies

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 28
5      defaultConfig {
6          applicationId "com.example.nidhisinha.lab3_cal"
7          minSdkVersion 15
8          targetSdkVersion 28
9          versionCode 1
10         versionName "1.0"
11     }
12     buildTypes {
13         release {
14             minifyEnabled false
15             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
16         }
17     }
18 }
19
20 dependencies {
21     implementation fileTree(dir: 'libs', include: ['*.jar'])
22     implementation 'com.android.support:appcompat-v7:28.0.0'
23     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
24     testImplementation 'junit:junit:4.12'
25     androidTestImplementation 'com.android.support.test:runner:1.0.2'
26     androidTestImplementation 'com.android.support.test:rules:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28 }
29
```

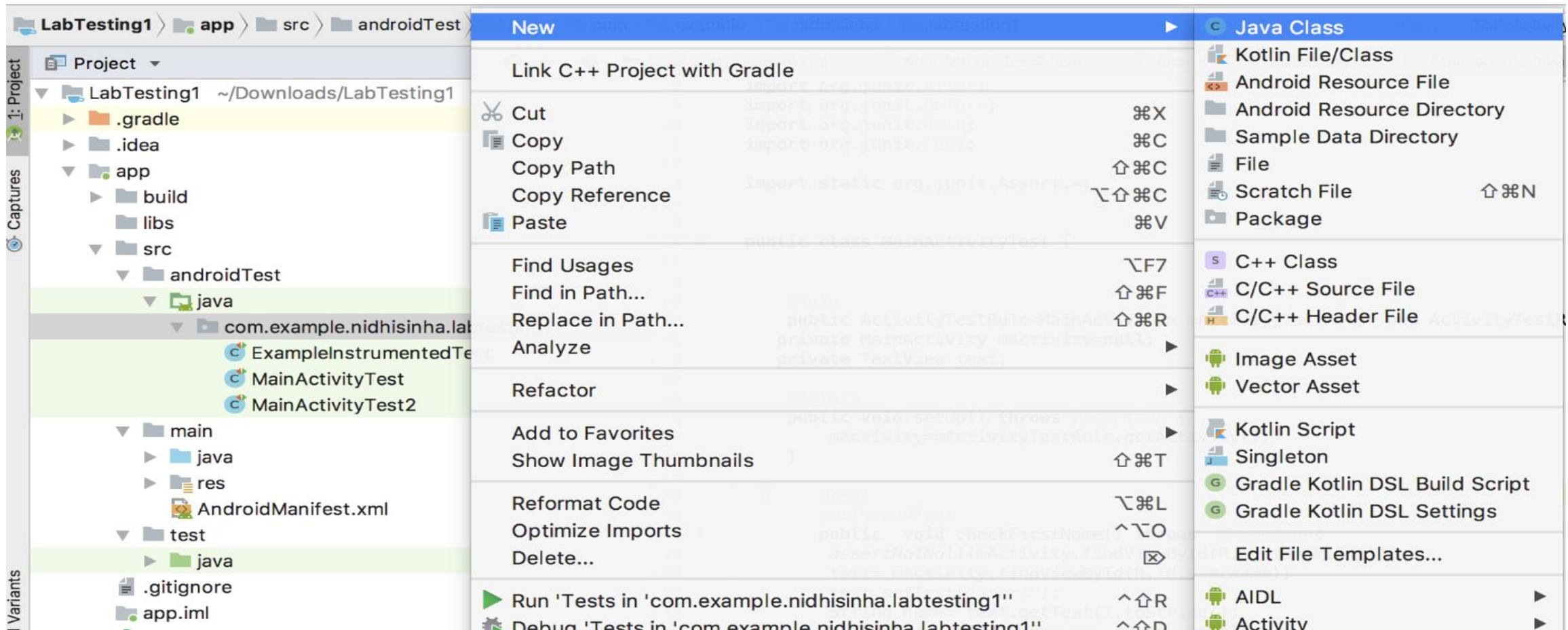
Set the instrumentation runner



```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 28
5      defaultConfig {
6          applicationId "com.example.nidhisinha.lab3_cal"
7          minSdkVersion 15
8          targetSdkVersion 28
9          versionCode 1
10         versionName "1.0"
11         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12     }
13     buildTypes {
14         release {
15             minifyEnabled false
16             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
17         }
18     }
19 }
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test:rules:1.0.2'
28     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
29 }
30
```

Add the test class under android tests

Right click on android test and create a new java class.



JUnit test case for User First name Sample

```
public class MainActivityTest {  
  
    @Rule  
    public ActivityTestRule<MainActivity> mActivityTestRule= new ActivityTestRule<MainActivity>(MainActivity.class);  
    private MainActivity mActivity=null;  
    private TextView text;  
  
    @Before  
    public void setUp() throws Exception {  
        mActivity=mActivityTestRule.getActivity();  
    }  
  
    @Test  
    @UiThreadTest  
    public void checkFirstName() throws Exception{  
        assertNotNull(mActivity.findViewById(R.id.textView1));  
        text= mActivity.findViewById(R.id.username);  
        text.setText("user1");  
        String name= text.getText().toString();  
        assertEquals("user",name);  
    }  
}
```


Right click on android test and run



Add another test class for Espresso UI test

Eg. to validate correct error message on UI when wrong email is entered

```
public class MainActivityTest2 {  
  
    @Rule  
    public ActivityTestRule<MainActivity> mActivityTestRule = new  
ActivityTestRule<>(MainActivity.class);  
  
    @Test  
    public void emailIsInvalid() {  
        onView(withId(R.id.username)).perform(typeText("user"), closeSoftKeyboard());  
        onView(withId(R.id.Lastname)).perform(typeText("test"), closeSoftKeyboard());  
        onView(withId(R.id.email)).perform(typeText("email@"), closeSoftKeyboard());  
        onView(withId(R.id.loginBtn)).perform(click());  
        onView(withText("Invalid Email")).check(matches(isDisplayed()));  
    }  
}
```

Eg. to validate correct error message on UI when wrong email is entered

```
@Test  
public void firstIsInvalid() {  
    onView(withId(R.id.username)).perform(typeText("user1"), closeSoftKeyboard());  
    onView(withId(R.id.loginBtn)).perform(click());  
    onView(withText("Invalid First Name")).check(matches(isDisplayed()));  
}
```

Run the test

Once the test class is created,
right click on android test and run.

