

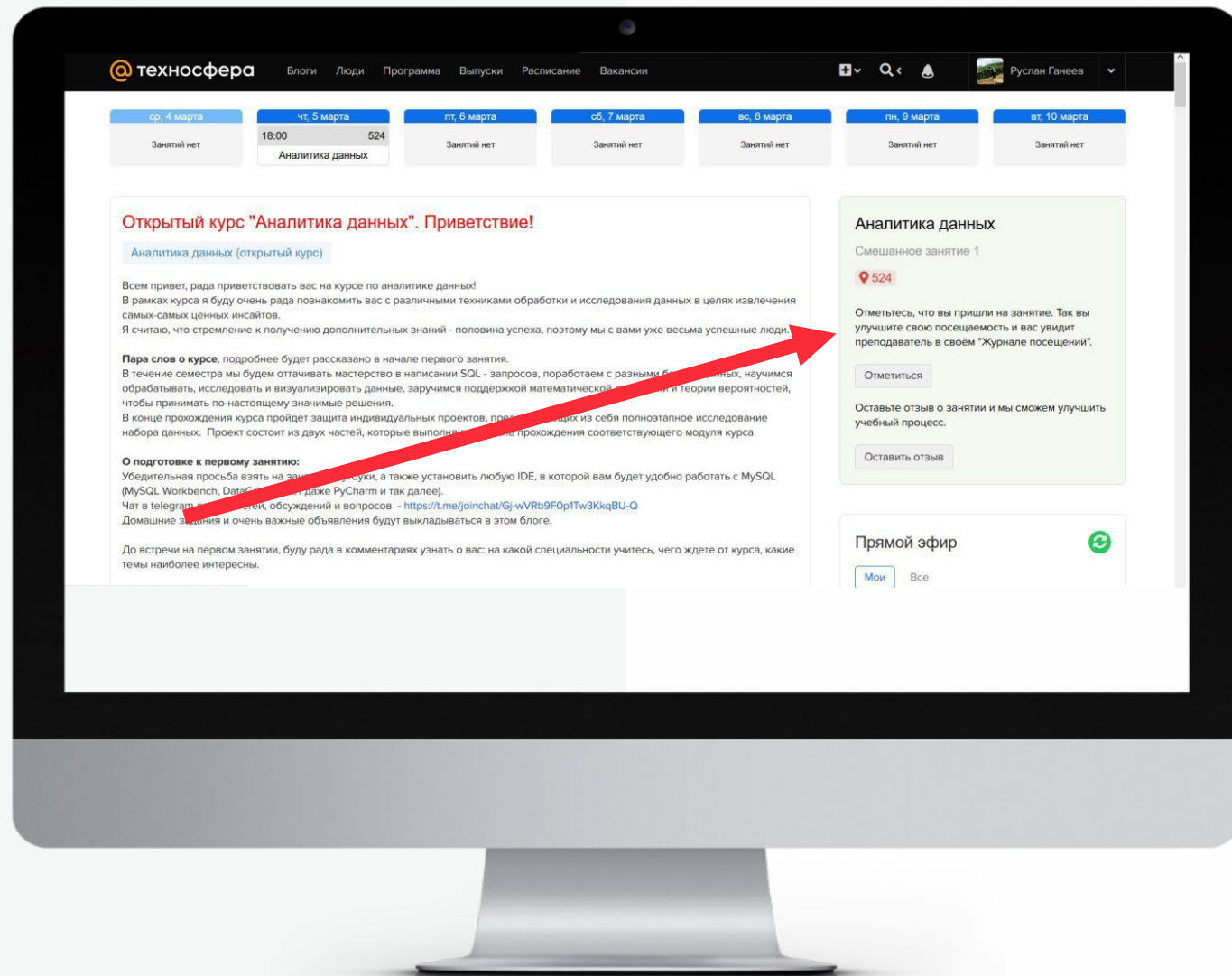


x @ mail.ru
group


Популярные библиотеки

Куликов Никита






Напоминание
отметиться
на портале



Содержание занятия

1. Утилитарные библиотеки
2. Картинки
3. DI
4. Тесты



Утилитарные библиотеки

1. Timber
2. Flipper
3. RxJava2Debug

Timber | Gradle

```
implementation 'com.jakewharton.timber:timber:4.7.1'
```

Timber | Как использовать

```
if (BuildConfig.DEBUG) {  
    Timber.plant(new Timber.DebugTree());  
}
```

```
Timber.i("Test")
```

Timber | Что внутри?

```
StackTraceElement[] stackTrace = new Throwable().getStackTrace();  
if (stackTrace.length <= CALL_STACK_INDEX) {  
    throw new IllegalStateException(  
        "Synthetic stacktrace didn't have enough elements: are you  
using proguard?");  
}  
return createStackElementTag(stackTrace[CALL_STACK_INDEX]);
```

RxJava2Debug | Gradle

```
implementation 'com.akaita.java:rxjava2-debug:1.4.0'
```


RxJava2Debug | Как использовать

```
public class MyApplication extends Application {  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
  
        // Enable RxJava assembly stack collection, to make RxJava crash reports clear and unique  
        // Make sure this is called AFTER setting up any Crash reporting mechanism as Crashlytics  
        RxJava2Debug  
            .enableRxJava2AssemblyTracking(new String[]{"com.example.myapp"});  
    }  
}
```

```
1  FATAL EXCEPTION: main
2  Process: com.akaita.fgas.debug, PID: 27300
3  ▼ io.reactivex.exceptions.OnErrorNotImplementedException: The mapper function returned a null value.
4      at io.reactivex.internal.functions.Functions$OnErrorMissingConsumer.accept(Functions.java:704)
5      at io.reactivex.internal.functions.Functions$OnErrorMissingConsumer.accept(Functions.java:701)
6      at io.reactivex.internal.observers.LambdaObserver.onError(LambdaObserver.java:74)
7      at io.reactivex.internal.operators.observable.ObservableObserveOn$ObserveOnObserver.checkTerminated(ObservableObserveOn.java:276)
8      at io.reactivex.internal.operators.observable.ObservableObserveOn$ObserveOnObserver.drainNormal(ObservableObserveOn.java:172)
9      at io.reactivex.internal.operators.observable.ObservableObserveOn$ObserveOnObserver.run(ObservableObserveOn.java:252)
10     at io.reactivex.android.schedulers.HandlerScheduler$ScheduledRunnable.run(HandlerScheduler.java:109)
11     at android.os.Handler.handleCallback(Handler.java:751)
12     at android.os.Handler.dispatchMessage(Handler.java:95)
13     at android.os.Looper.loop(Looper.java:154)
14     at android.app.ActivityThread.main(ActivityThread.java:6077)
15     at java.lang.reflect.Method.invoke(Native Method)
16     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:866)
17     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:756)
18  ▼ Caused by: java.lang.NullPointerException: The mapper function returned a null value.
19     at io.reactivex.internal.functions.ObjectHelper.requireNonNull(ObjectHelper.java:39)
20     at io.reactivex.internal.operators.observable.ObservableMap$MapObserver.onNext(ObservableMap.java:59)
21     at io.reactivex.internal.operators.observable.ObservableSubscribeOn$SubscribeOnObserver.onNext(ObservableSubscribeOn.java:58)
22     at io.reactivex.internal.operators.observable.ObservableScalarXMap$ScalarDisposable.run(ObservableScalarXMap.java:248)
23     at io.reactivex.internal.operators.observable.ObservableJust.subscribeActual(ObservableJust.java:35)
24     at io.reactivex.Observable.subscribe(Observable.java:10838)
25     at io.reactivex.internal.operators.observable.ObservableSubscribeOn$SubscribeTask.run(ObservableSubscribeOn.java:96)
26     at io.reactivex.Scheduler$DisposeTask.run(Scheduler.java:452)
27     at io.reactivex.internal.schedulers.ScheduledRunnable.run(ScheduledRunnable.java:61)
28     at io.reactivex.internal.schedulers.ScheduledRunnable.call(ScheduledRunnable.java:52)
29     at java.util.concurrent.FutureTask.run(FutureTask.java:237)
30     at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:272)
31     at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1133)
32     at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:607)
33     at java.lang.Thread.run(Thread.java:761)
34
35
```



```
1 FATAL EXCEPTION: main
2 Process: com.akaita.fgas.debug, PID: 22538
3 java.lang.Throwable: The mapper function returned a null value.
4     at io.reactivex.internal.functions.ObjectHelper.requireNonNull(ObjectHelper.java:39)
5     at io.reactivex.internal.operators.observable.ObservableMap$MapObserver.onNext(ObservableMap.java:59)
6     at hu.akarnokd.rxjava2.debug.ObservableOnAssembly$OnAssemblyObserver.onNext(ObservableOnAssembly.java:55)
7     at io.reactivex.internal.operators.observable.ObservableSubscribeOn$SubscribeOnObserver.onNext(ObservableSubscribeOn.java:58)
8     at hu.akarnokd.rxjava2.debug.ObservableOnAssembly$OnAssemblyObserver.onNext(ObservableOnAssembly.java:55)
9     at io.reactivex.internal.operators.observable.ObservableScalarXMap$ScalarDisposable.run(ObservableScalarXMap.java:248)
10    at io.reactivex.internal.operators.observable.ObservableJust.subscribeActual(ObservableJust.java:35)
11    at io.reactivex.Observable.subscribe(Observable.java:10838)
12    at hu.akarnokd.rxjava2.debug.ObservableOnAssemblyScalarCallable.subscribeActual(ObservableOnAssemblyScalarCallable.java:41)
13    at io.reactivex.Observable.subscribe(Observable.java:10838)
14    at io.reactivex.internal.operators.observable.ObservableSubscribeOn$SubscribeTask.run(ObservableSubscribeOn.java:96)
15    at io.reactivex.Scheduler$DisposeTask.run(Scheduler.java:452)
16    at io.reactivex.internal.schedulers.ScheduledRunnable.run(ScheduledRunnable.java:61)
17    at io.reactivex.internal.schedulers.ScheduledRunnable.call(ScheduledRunnable.java:52)
18    at java.util.concurrent.FutureTask.run(FutureTask.java:237)
19    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:272)
20    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1133)
21    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:607)
22    at java.lang.Thread.run(Thread.java:761)
23 Caused by: java.lang.Throwable
24     at com.example.myapp.TopActivity.onResume(TopActivity.java:205)
25     at android.app.Instrumentation.callActivityOnResume(Instrumentation.java:1269)
26     at android.app.Activity.performResume(Activity.java:6766)
27     at android.app.ActivityThread.performResumeActivity(ActivityThread.java:3377)
28     at android.app.ActivityThread.handleResumeActivity(ActivityThread.java:3440)
29     at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2713)
30     at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1460)
31     at android.os.Handler.dispatchMessage(Handler.java:102)
32     at android.os.Looper.loop(Looper.java:154)
33     at android.app.ActivityThread.main(ActivityThread.java:6077)
34     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:866)
35     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:756)
```

Flipper | Gradle

```
repositories {  
    jcenter()  
}  
  
dependencies {  
    debugImplementation 'com.facebook.flipper:flipper:0.66.0'  
    debugImplementation 'com.facebook.soloaders:soloaders:0.9.0'  
  
    releaseImplementation 'com.facebook.flipper:flipper-noop:0.66.0'  
}
```

Flipper | Manifest

```
<activity  
  android:name="com.facebook.flipper.android.diagnostics.FlipperDiagnosticActivity"  
  android:exported="true"/>
```

Flipper | Как использовать

```
import com.facebook.flipper.android.AndroidFlipperClient;
import com.facebook.flipper.android.utils.FlipperUtils;
import com.facebook.flipper.core.FlipperClient;
import com.facebook.flipper.plugins.inspector.DescriptorMapping;
import com.facebook.flipper.plugins.inspector.InspectorFlipperPlugin;

public class MyApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        SoLoader.init(this, false);

        if (BuildConfig.DEBUG && FlipperUtils.shouldEnableFlipper(this)) {
            final FlipperClient client = AndroidFlipperClient.getInstance(this);
            client.addPlugin(new InspectorFlipperPlugin(this, DescriptorMapping.withDefaults()));
            client.start();
        }
    }
}
```

Pixel_2_API_28

Notifications

Pixel_2_API_28

Crash Reporter

Logs

Flipper

Example Plugin

Images

Layout

LeakCanary

Network

Shared Preferences View

Plugin not showing?

Search

com.facebook.flipper.sample

MainActivity

PhoneWindow

DecorView

LinearLayout

ViewStub id=@android:id/action_mode_bar_stub

FrameLayout

ActionBarOverlayLayout id=@id/decor_content_parent

ContentFrameLayout id=@android:id/content

LithoView

RootComponent key=1

Text key=1

Text key=2

Text key=3

TextDrawable

Text key=4

Text key=5

ActionBarContainer id=@id/action_bar_container

Props

accessibleClickableSpans: ☐

breakStrategy: 0

clickableSpanExpandedOffset: 0

clipToBounds: ☒

customEllipsisText: undefined

ellipsize: undefined

extraSpacing: 0

glyphWarming: ☐

highlightColor: ☐ rgba(0, 0, 0, 0.00)

highlightEndOffset: -1

highlightStartOffset: 0

hyphenation: 0

isSingleLine: ☐

justificationMode: 0

letterSpacing: 0

lineHeight: 1

linkColor: 0

maxEms: -1

maxLines: 1

maxTextWidth: 0

minEms: -1

minLines: 1

minTextWidth: 0

minimallyWidth: 0

minimallyWidth: 0

shadowColor: 0

shadowDx: 0

shadowDy: 0

shadowRadius: 0

shouldIncludeFontPadding: ☒

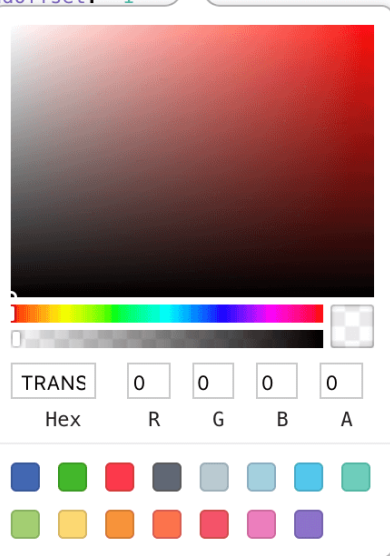
spacingMultiplier: 1

spanListener: undefined

text: "Trigger Notification"

textAlignment: "ALIGN_NORMAL"

textColor: ☐ rgba(0, 0, 0, 0.00)



Flipper | Что внутри?

```
adb forward tcp:6100 tcp:7100
```




Картинки

1. Glide

Glide | Gradle

```
repositories {  
    google()  
    jcenter()  
}  
  
dependencies {  
    implementation 'com.github.bumptech.glide:glide:4.11.0'  
    annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'  
}
```

Glide | Proguard

```
-keep public class * implements com.bumptech.glide.module.GlideModule
-keep class * extends com.bumptech.glide.module.AppGlideModule {
    <init>(...);
}
-keep public enum com.bumptech.glide.load.ImageHeaderParser$** {
    **[] $VALUES;
    public *;
}
-keep class com.bumptech.glide.load.data.ParcelFileDescriptorRewinder$InternalRewinder {
    *** rewind();
}

# for DexGuard only
-keepresourcexmlelements manifest/application/meta-data@value=GlideModule
```

Glide | Как использовать

```
ImageView imageView = (ImageView) findViewById(R.id.my_image_view);  
Glide.with(this).load("http://goo.gl/gEgYUd").into(imageView);
```

Glide | Что внутри

```
public class RequestManager  
    implements ComponentCallbacks2, LifecycleListener
```



DI

1. Dagger
2. koin

Dagger | Gradle

```
// Add Dagger dependencies
dependencies {
    api 'com.google.dagger:dagger:2.x'
    api 'com.google.dagger:dagger-android:2.x'
    api 'com.google.dagger:dagger-android-support:2.x' // if you use the support libraries
    annotationProcessor 'com.google.dagger:dagger-android-processor:2.x'
    annotationProcessor 'com.google.dagger:dagger-compiler:2.x'
}
```

Dagger | Как использовать

```
@Component()  
public interface ApplicationComponent {  
  
}
```


Dagger | Как использовать

```
@Module
public class ApplicationModule {
    private Context context;

    public ApplicationModule(Context context) {
        this.context = context;
    }

    @Provides
    @Singleton
    public Context provideEndpoint() {
        return context;
    }
}
```

Dagger | Как использовать

```
@Component(modules = {  
    ApplicationModule.class  
})  
public interface ApplicationComponent {  
  
}
```

Dagger | Как использовать

```
DaggerApplicationComponent  
    .applicationModule(ApplicationModule(this))  
    .create()
```

Dagger | Как использовать

```
@Component()  
public interface ApplicationComponent {  
    void inject(MyClass myObject);  
}
```

Dagger | Как использовать

```
public class MyClass {  
  
    @Inject  
    public Context context;  
  
    Application.getAppComponent().inject(this);  
}
```

Dagger | Как использовать

```
@Component()  
public interface ApplicationComponent {  
    void plus(NetworkComponent cmpnt);  
}
```

Dagger | Что внутри

#031

Koin | Gradle

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath "org.koin:koin-gradle-plugin:$koin_version"  
    }  
}  
  
apply plugin: 'koin'
```


Koin | Gradle

```
// Koin for Kotlin  
implementation "org.koin:koin-core:$koin_version"  
// Koin extended & experimental features  
implementation "org.koin:koin-core-ext:$koin_version"  
// Koin for Unit tests  
testImplementation "org.koin:koin-test:$koin_version"  
// Koin for Android  
implementation "org.koin:koin-android:$koin_version"  
// Koin Android Scope features  
implementation "org.koin:koin-android-scope:$koin_version"  
// Koin Android ViewModel features  
implementation "org.koin:koin-android-viewmodel:$koin_version"  
// Koin Android Experimental features  
implementation "org.koin:koin-android-ext:$koin_version"
```

Koin | Как использовать

```
// Given some classes  
class Controller(val service : BusinessService)  
class BusinessService()  
  
// just declare it  
val myModule = module {  
    single { Controller(get()) }  
    single { BusinessService() }  
}
```

Тесты

1. Roboelectric
2. Mockk
3. Espresso

Roboelectric | Gradle

```
testImplementation "org.robolectric:robolectric:4.4"
```

Roboelectric | Как пользоваться

```
@RunWith(AndroidJUnit4.class)
public class MyActivityTest {

    @Test
    public void clickingButton_shouldChangeResultsViewText() throws Exception {
        Activity activity = Roboelectric.setupActivity(MyActivity.class);

        Button button = (Button) activity.findViewById(R.id.press_me_button);
        TextView results = (TextView) activity.findViewById(R.id.results_text_view);

        button.performClick();
        assertEquals(results.getText().toString(), equalTo("Testing Android Rocks!"));
    }
}
```

Mockk | Gradle

```
testImplementation "io.mockk:mockk:{version}"
```

Mockk | Как пользоваться

```
val car = mockk<Car>()

every { car.door(DoorType.FRONT_LEFT).windowState() } returns WindowState.UP

car.door(DoorType.FRONT_LEFT) // returns chained mock for Door
car.door(DoorType.FRONT_LEFT).windowState() // returns WindowState.UP

verify { car.door(DoorType.FRONT_LEFT).windowState() }

confirmVerified(car)
```

Espresso | Как пользоваться

```
// Щелчок кнопки у первой активности
public void onClick(View view) {
    EditText nameEditText = (EditText) findViewById(R.id.editTextName);
    Intent intent = new Intent(MainActivity.this, SecondActivity.class);
    intent.putExtra("name", nameEditText.getText().toString());

    startActivity(intent);
}

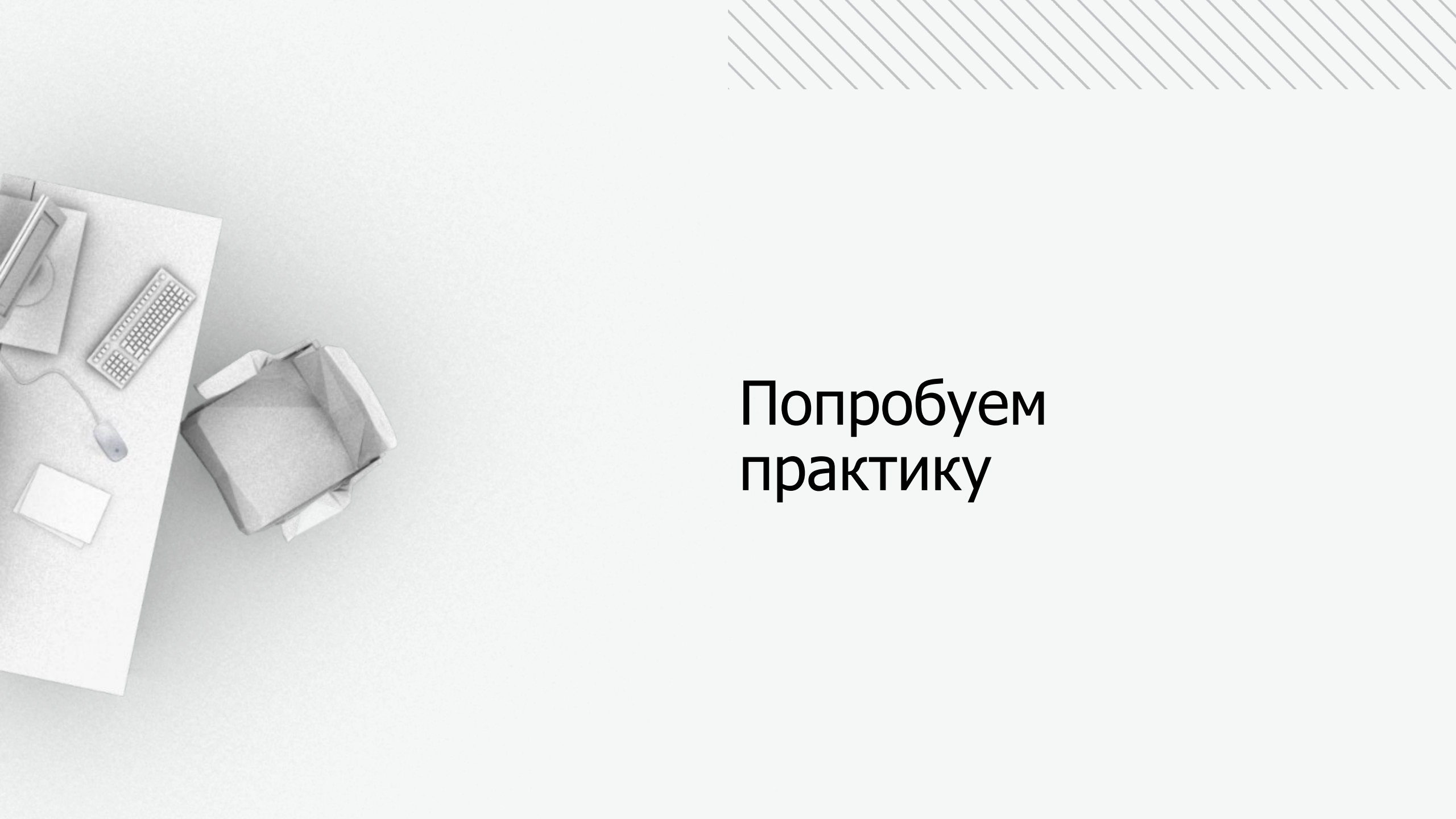
// Запускается вторая активность
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);

    String user = "";
    user = getIntent().getExtras().getString("name");

    TextView infoTextView = (TextView) findViewById(R.id.textViewInfo);
    infoTextView.setText("Hello " + user);
}
```


Espresso | Как пользоваться

```
public class MainActivityTest {  
  
    @Rule  
    public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(  
        MainActivity.class);  
  
    @Before  
    public void setUp() throws Exception {  
  
    }  
  
    @Test  
    public void onClick() throws Exception {  
        onView(withId(R.id.editTextName)).perform(typeText("Barsik"));  
        onView(withId(R.id.button)).perform(click());  
        onView(withId(R.id.textViewInfo)).check(matches(withText("Hello Barsik")));  
    }  
}
```



Попробуем
практику

**СПАСИБО
ЗА ВНИМАНИЕ**

