Влияем на тестовое окружение «без рук»

HUMANS

МТС

Алексей Рассказов

Humans IT

Екатерина Батеева

МТС

План

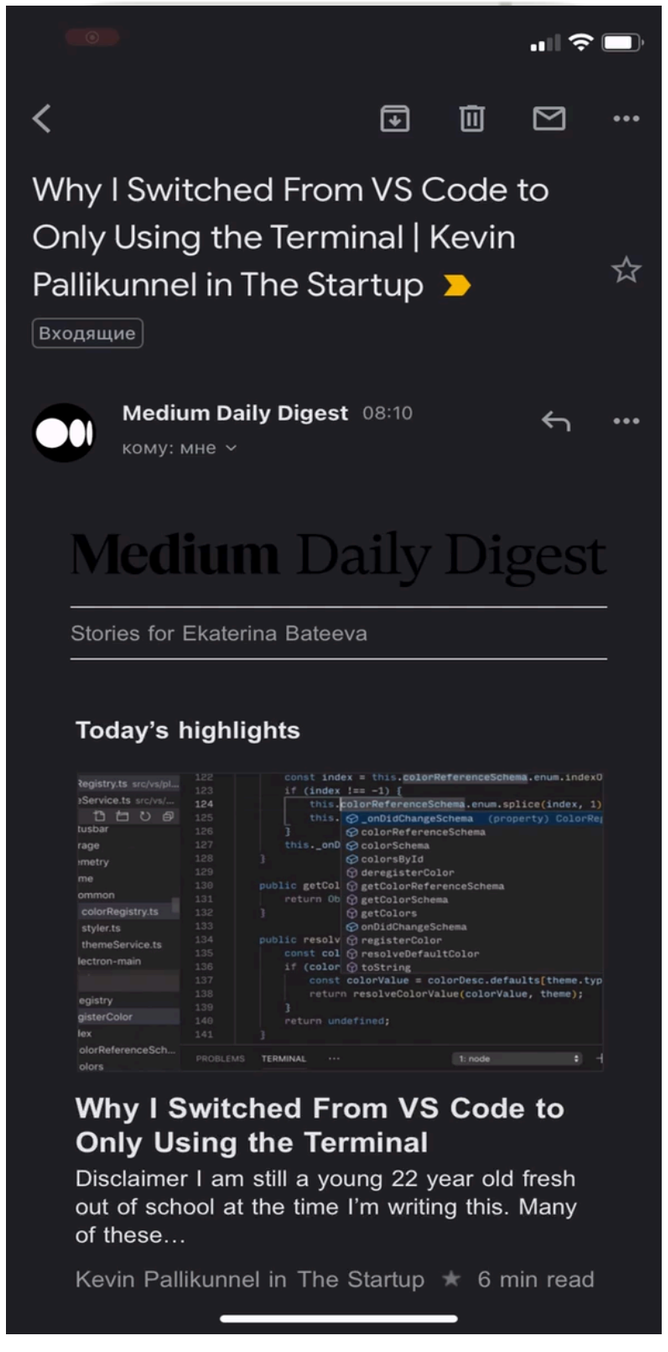
HUMANS

МТС

- Темная и светлая темы
- Управление устройством
- Пользовательский ввод
- Локализация
- Стартуем с экрана

МТС

Темная и светлая темы


Why I Switched From VS Code to Only Using the Terminal | Kevin Pallikunnel in The Startup
Входящие
Medium Daily Digest 08:10
кому: мне
Medium Daily Digest
Stories for Ekaterina Bateeva
Today's highlights
Why I Switched From VS Code to Only Using the Terminal
Disclaimer I am still a young 22 year old fresh out of school at the time I'm writing this. Many of these…
Kevin Pallikunnel in The Startup ★ 6 min read

Как протестировать?

HUMANS МТС

Создаем объект

```
let vc = UIHostingController(rootView: ViewToTest())
vc.view.frame = UIScreen.main.bounds  //Set frame
```

Устанавливаем тему

```
vc.overrideUserInterfaceStyle = .dark  //Set dark mode trait
```

Проверяем

```
assertSnapshot(matching: vc, as: .image)
```

Как протестировать?

HUMANS МТС

Меняем через консоль

```
1 xcrun simctl ui booted appearance light
2 xcrun simctl ui booted appearance dark
```

Как протестировать?

HUMANS МТС

Меняем через консоль

```
override func viewDidLoad() {
        super.viewDidLoad()
        #if DEBUG
        // This changes appearance only for debug mode
          overrideUserInterfaceStyle = .dark
        #endif
}
```

Поддержка темы

HUMANS МТС

Style vs Theme

Night ресурсы

Проблемы при поддержке

HUMANS МТС

Перезатирание стилей

Наследование от BaseActivity

Нарушение именования ресурсов

Hardcode ресурсов

Получаем список Activities

HUMANS МТС

```
fun getActivities(context: Context): Array<ActivityInfo> {
    val packageManager = context.packageManager
    val packageName = context.applicationContext.packageName
    val info = packageManager.getPackageInfo(
     packageName,
     PackageManager.GET_ACTIVITIES
 )
    return info.activities
}
```

Тестируем наследников

HUMANS

МТС

```kotlin
@Test
fun `activity inheritance`() {
    getActivities(targetContext).filter {
        it.name !in BLACKLISTED_ACTIVITIES
    }
    .map { Class.forName(it.name).kotlin }
    .forEach { activityClass ->
        assertTrue(
            activityClass.isSubclassOf(BASE_ACTIVITY)
        )
    }
}
```

Создаем Linter Issue

HUMANS MTC

```
Issue.create(
    ID,
    DESCRIPTION,
    CATEGORY,
    EXPLANATION,
    PRIORITY,
    SEVERITY,
    Implementation(
        Detector::class.java,
        Scope.RESOURCE_FILE_SCOPE
    )
)
```

Lint Report

HUMANS MTC

Overview

Usability	
1	⚠ LibraryValueNaming: Incorrect naming of resource

Incorrect naming of resource

../../src/main/res/values/dimens.xml:3: Resources can be overridden by values in by other models. Please specify unique name.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <dimen name="some_incorrect_name_padding">16dp</dimen>
4 </resources>
```

LibraryValueNaming | Usability | Warning | Priority 4/10

Пишем детектор

HUMANS МТС

```
override fun appliesTo(folderType: ResourceFolderType): Boolean {
    return folderType == ResourceFolderType.VALUES
}

override fun getApplicableAttributes(): Collection<String>? {
    return listOf(ATTRIBUTE_NAME)
}
```

Проверяем атрибуты

HUMANS МТС

```
override fun visitAttribute(context: XmlContext, attribute: Attr) {
    if (!REGEXP.matches(attribute.value)) {
        context.report(
            Issue.ISSUE,
            context.getLocation(attribute),
            Issue.EXPLANATION
        )
    }
}
```

Решения

HUMANS

МТС

Корректность наследования

Уникальность стилей и ресурсов

Статическая верификация атрибутов

Управление устройством

Управляем темой

HUMANS

МТС

Провайдеры темы

Тестовый манифест

setDefault NightMode

Robolectric qualifiers

Как протестировать?

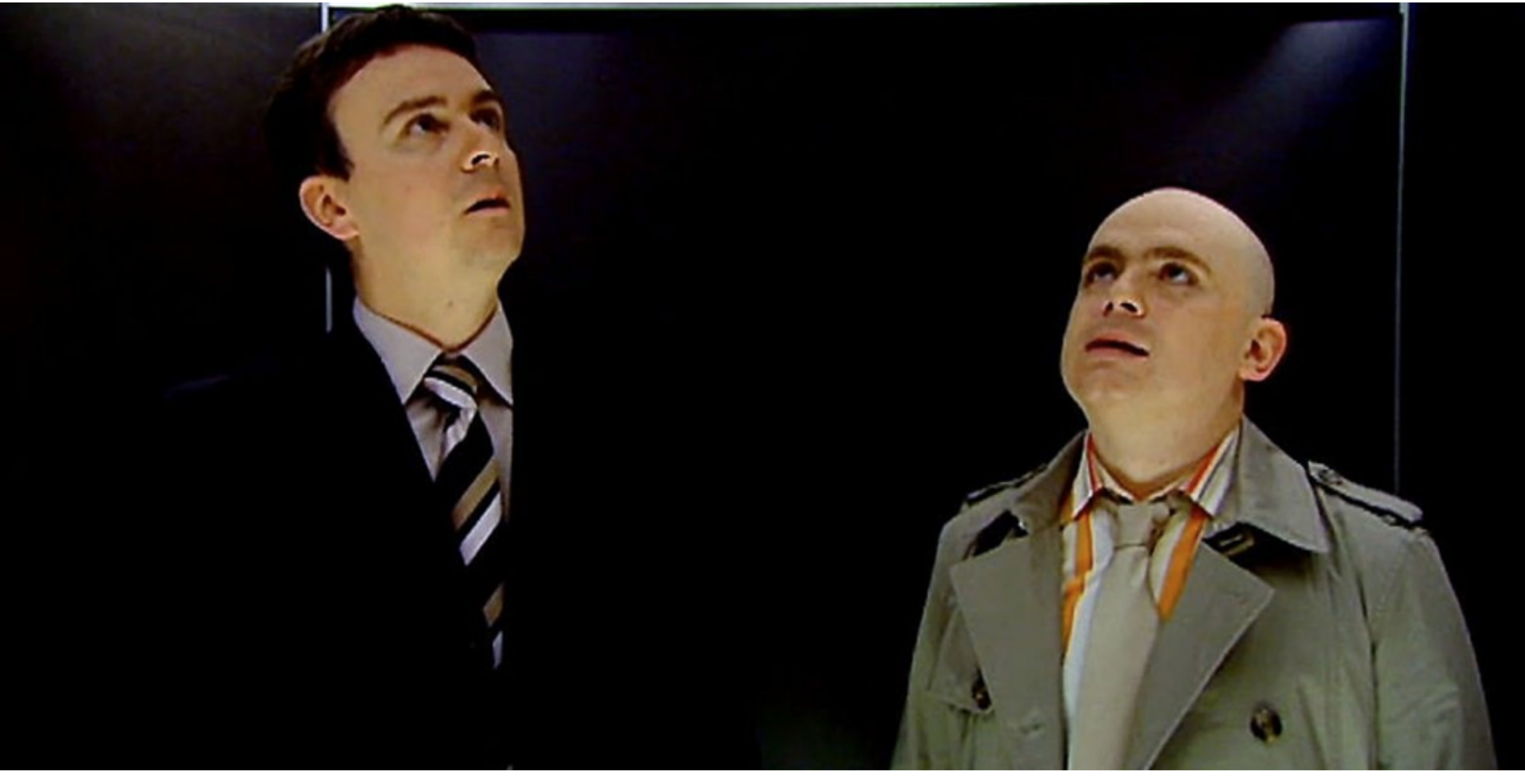
HUMANS

МТС

Почему не Settings?

HUMANS МТС

- Мало опций у симулятора
- Нужно прокликивать тестом
- Нужно проверять состояние настроек



HUMANS МТС

Установи темную тему

Измени оформление на светлое

Измени оформление на светлое

Установи светлое оформление

Поменяй тему

Измени тему

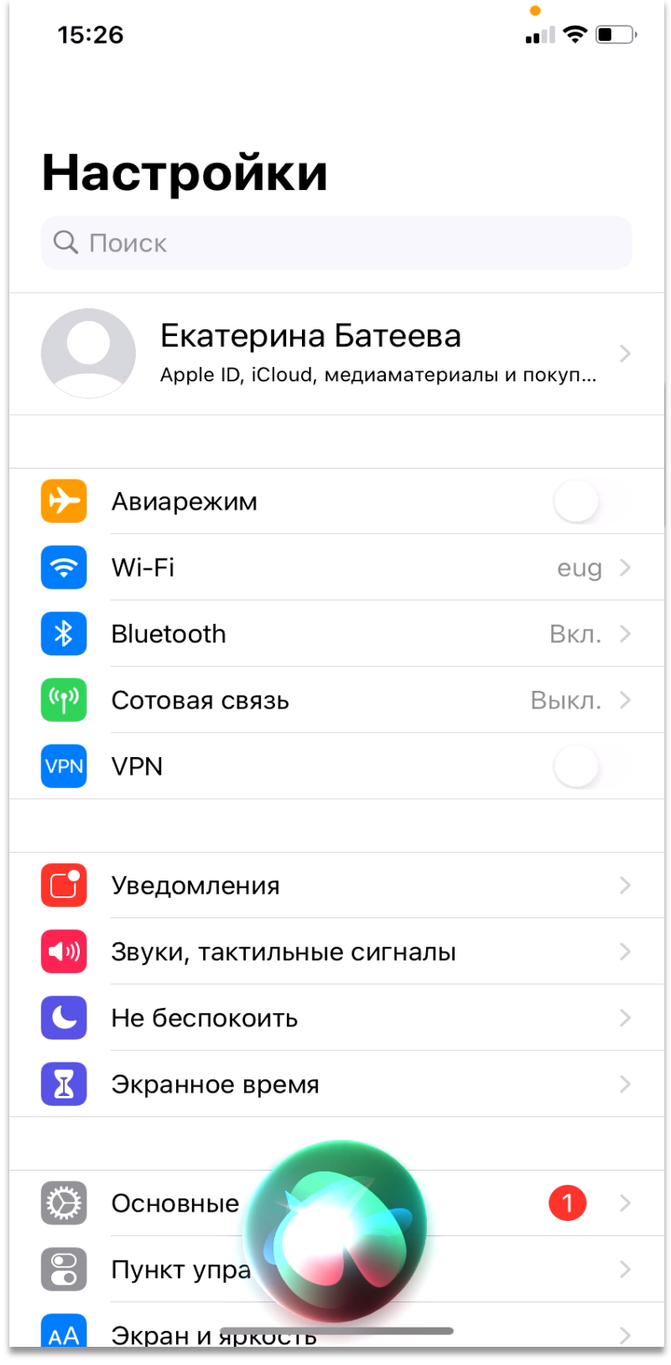
HUMANS MTC

Change Appearance

Измени вид

Измени оформление

Well no... but actually yes!



HUMANS

МТС

А что насчет Siri?

HUMANS МТС

```
Выключить Wi-Fi
  XCUIDevice.shared()
  .siriService.activate(
    voiceRecognitionText: “Switch off Wi-fi”
  )
```

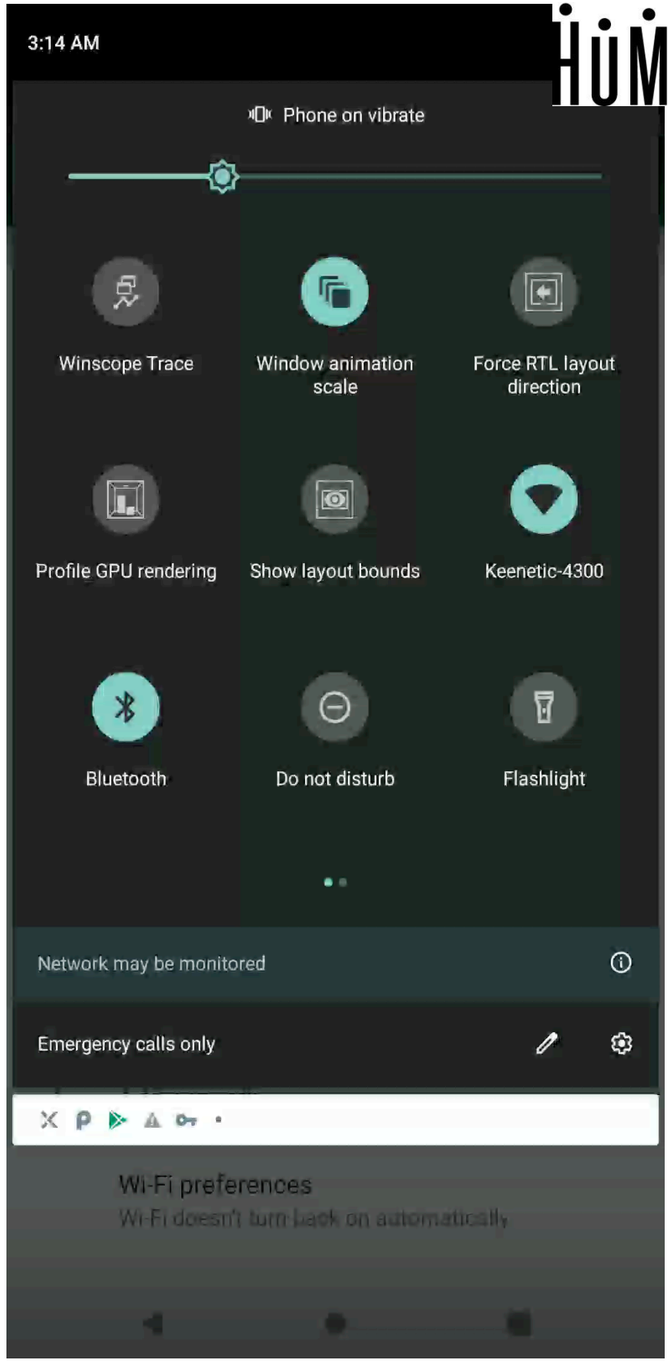
Переключение темы

HUMANS МТС

```
@Test
fun `black theme - background is black`() {
    context.setTheme(R.style.Theme_Dark)
    val colorResId = context.getAttributeValue(R.attr.hms_background)
    val color = ContextCompat.getColor(context, colorResId)
    expectThat(color).isEqualTo(Color.BLACK)
}
```

Управляем wi-fi

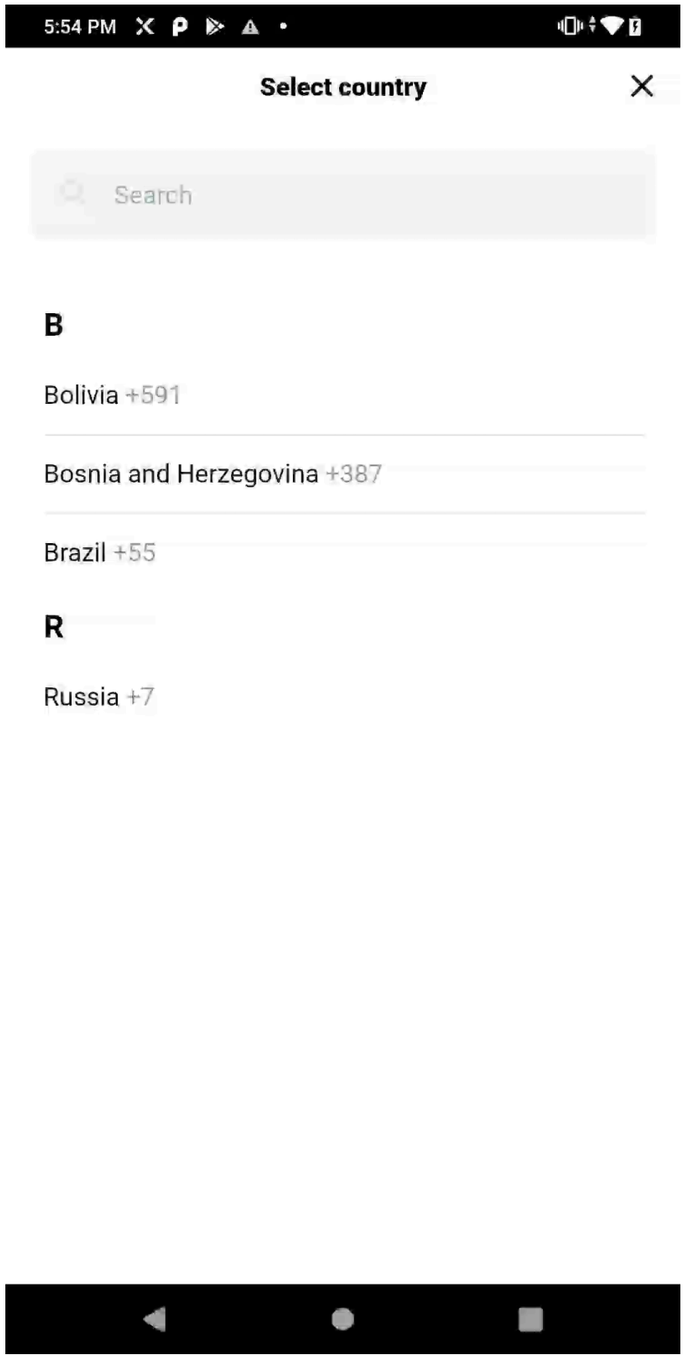
```
UiDevice.executeShellCommand(
    "settings put global
     airplane_mode_on 1"
)
device.network.enable()
device.network.disable()
```



Пользовательский ввод

Вводим данные через эспрессо

```
step("search 'Bra'") {
    search { typeText("Bra") }
    clearSearch { isVisible() }
}
step("Verify 'Brazil' in recycler") {
    countryCodes {
        childAt<CountryCodeItem>(0) {
            text.hasText("Brazil +55")
        }
    }
}
```



МТС

HUMANS

Проверяем inputType

HUMANS МТС

```
UiDevice.executeShellCommand(
    "dumpsys input_method | grep mInputShown"
).contains(
    "mInputShown=true "
)
```

Проверяем inputType

HUMANS МТС

Открытие и закрытие клавиатуры

HUMANS МТС

```
external_kb_connected = false
osascript -e 'quit app "Simulator”’
```

Открытие и закрытие клавиатуры

HUMANS МТС

```
SIMUS_KEYBOARD=$(/usr/libexec/PlistBuddy -c "Print :DevicePreferences"
~/Library/Preferences/com.apple.iphonesimulator.plist | perl -lne 'print
$1 if /^ (\S*) =/')
```

Открытие и закрытие клавиатуры

HUMANS МТС

```
echo "$SIMUS_KEYBOARD" | while read -r a;
  do /usr/libexec/PlistBuddy -c
  "Set :DevicePreferences:$a:ConnectHardwareKeyboard
$external_kb_connected"
~/Library/Preferences/com.apple.iphonesimulator.plist
  || /usr/libexec/PlistBuddy -c
"Add :DevicePreferences:$a:ConnectHardwareKeyboard bool
$external_kb_connected"
~/Library/Preferences/com.apple.iphonesimulator.plist;
done
```

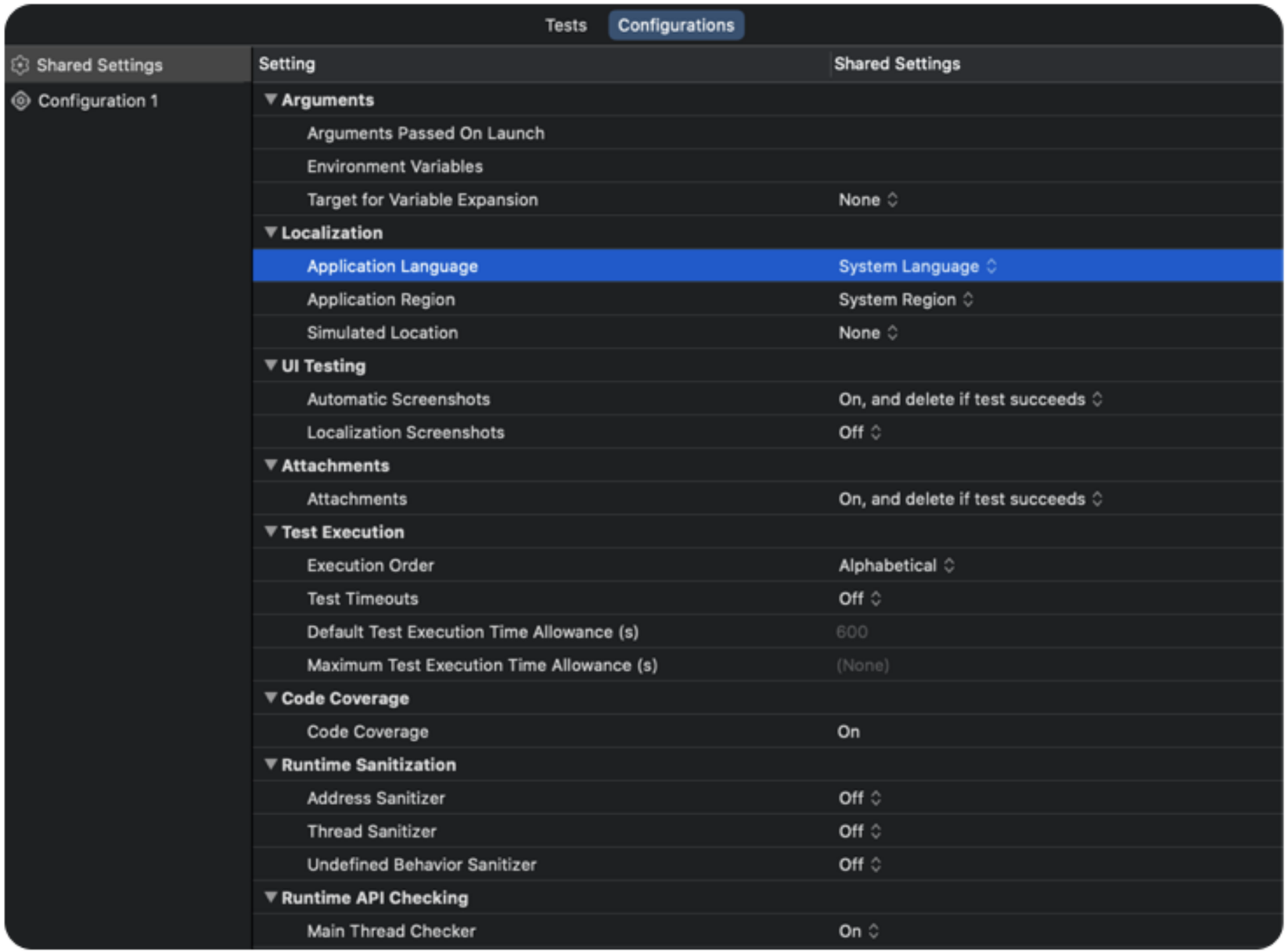
МТС

Локализация

Локализация

HUMANS МТС

- Устанавливаем через XCTestPlan
- С помощью параметров внутри тестов


Tests
Configurations
Shared Settings
Configuration 1
Setting
Shared Settings
Arguments
Arguments Passed On Launch
Environment Variables
Target for Variable Expansion
None
Localization
Application Language
System Language
Application Region
System Region
Simulated Location
None
UI Testing
Automatic Screenshots
On, and delete if test succeeds
Localization Screenshots
Off
Attachments
Attachments
On, and delete if test succeeds
Test Execution
Execution Order
Alphabetical
Test Timeouts
Off
Default Test Execution Time Allowance (s)
600
Maximum Test Execution Time Allowance (s)
(None)
Code Coverage
Code Coverage
On
Runtime Sanitization
Address Sanitizer
Off
Thread Sanitizer
Off
Undefined Behavior Sanitizer
Off
Runtime API Checking
Main Thread Checker
On

Локализация

HUMANS МТС

```
override func setUp() {
 super.setUp()
 continueAfterFailure = false
 XCUIApplication().launchArguments += [“-AppleLanguages”, “(fr)”]
 XCUIApplication().launchArguments += [“-AppleLocale”, “fr_FR”]
 XCUIApplication().launch()
}
```

Меняем локаль в приложении

HUMANS МТС

```
fun Context.applySelectedAppLanguage(): Context {
    val locale =
LanguageProvider.create(this).getSelectedLanguage().locale
    val newConfig = Configuration(resources.configuration)
    Locale.setDefault(locale)
    newConfig.setLocale(locale)
    return createConfigurationContext(newConfig)
}
```

Проблематика локализации

HUMANS

МТС

Огромное количество вариаций

Незнание языка

Переключение локализации требует перезагрузку Activity

Нет гарантии, что не произойдет перезатирание ресурсов

Тестируем с robolectric

HUMANS MTC

```
@RunWith(ParameterizedRobolectricTestRunner::class)
@Config(qualifiers = "ru")
class SampleFormatterRuLocaleTest(
    private val durationMs: Double,
    private val expectedResult: String
) {

    @Before
    fun setUp() {
        // initialize di graph with required component
    }
```

Тест форматера

HUMANS

МТС

```
@Test
fun `format - elapsed time`() {
    val formatterImpl = get<DurationFormatter>()
    val actualMessage = formatterImplementation.format(durationMs)

    expectThat(actualMessage).isEqualTo(expectedResult)
}
```

Добавляем параметры

HUMANS МТС

```
companion object {
    @JvmStatic
    @ParameterizedRobolectricTestRunner.Parameters(
        name = "duration: {0} - expected result: {1}"
    )
    fun data() = listOf(
        arrayOf(1.hours.inMilliseconds, "1 час"),
        arrayOf(1.days.inMilliseconds, "1 день"),
    )
}
```

Решения

HUMANS МТС

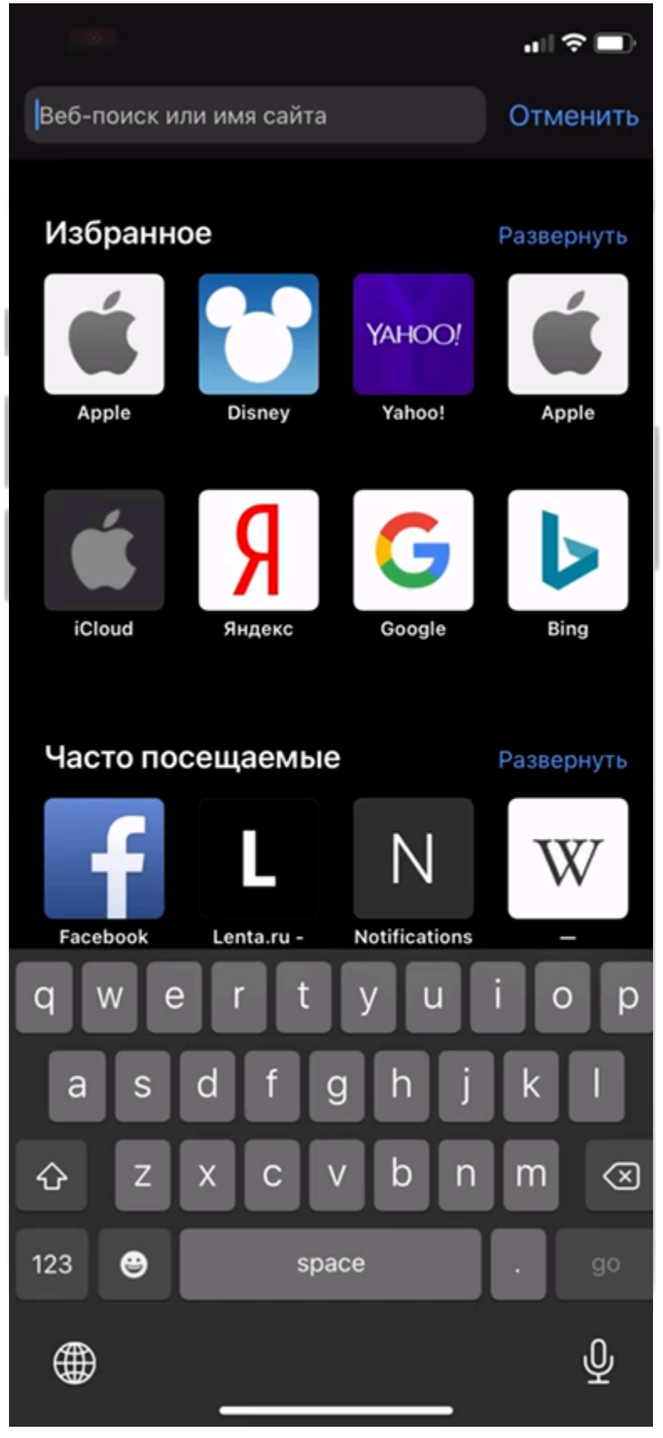
Linter уникальности ключей и пропущенных переводов

Тестируем критические части локализации

Храним переводы вместе с UI

Стартуем с экрана

Deeplink


Веб-поиск или имя сайта
Отменить
Избранное
Развернуть
Apple
Disney
Yahoo!
Apple
iCloud
Яндекс
Google
Bing
Часто посещаемые
Развернуть
Facebook
Lenta.ru -
Notifications
—
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
space
.
go

Передаем аргумент

```
1 XCUIApplication(). launchEnvironment += [“startScreen”,
2 “CustomViewController”]
```

Получаем ViewController в приложении

```
1 ProcessInfo.processInfo.environment[“startScreen”]
```

Связываем с `ViewController` и подменяем `rootViewController` в `AppDelegate`

Стартуем Activity

HUMANS

МТС

ActivityScenario

VS

ActivityTestRule

Выделяем PageObject с помощью Kakao

МТС

HUMANS

```
1  object BasicScreen : Screen<BasicScreen>() {
2      val title = KTextView { withId(R.id.title) }
3      val message = KTextView { withId(R.id.message) }
4      val action = KTextView { withId(R.id.action) }
5      val icon = KImageView { withId(R.id.icon) }
6  }
```

Подключаем в тест

HUMANS МТС

- feature module
- implementation в instrumentation module
- testImplementation в feature module

androidTestImplementation в более высокоуровневые модули

Тестируем с robolectric

HUMANS

МТС

```
@Test
fun `sing in screen - elements displayed`() {
    SignInScreen {
        exit { hasDrawable(R.drawable.ic_close_black_24dp) }
        title { hasText(R.string.sign_in__entry_title) }
        loginInput { hasHint(getTargetContext().getString(R.string.sign_in_
        // rest of verifications
    }
}
```

Тестируем сценарии

HUMANS МТС

```
@Test
fun enterValidPhoneNumber_clickContinue_verificationScreenOpened() {
    before { }.after { }.run {
        scenario(EnterContactAndClickNext(correctPhone.getContact()))
        step("Verification screen displayed") {
            VerificationCodeScreen {
                // screen verification
            }
        }
    }
}
```

Тестируем сценарии

HUMANS МТС

\+

Возможность переиспользования
в robolectic и androidTest

Тесты разносятся по уровням

Дешёвые unit для проверки того
что находится на каждом экране

\−

Дорогие instrumentation
для проверки общего поведения

Необходимость сложного
подключения

Robolectric не поддерживает
kaspresso

HUMANS МТС

Выводы