Автоматизация тестирования мобильных приложений в облаке

Дмитрий Сидоренко, Nordcloud/RoboQA

Содержание

- Два экстремальных подхода к тестированию
- Цели автоматизации тестирования
- Языки и фреймворки для автоматизации
- Как перейти от ручного тестирования к автоматическому?
- Пример использования AWS Device Farm
- Какие еще бывают Device Farms
- Победа сил разума над толстыми бюджетами

Мое геймдев прошлое

QUADCOM
interactive

glu™

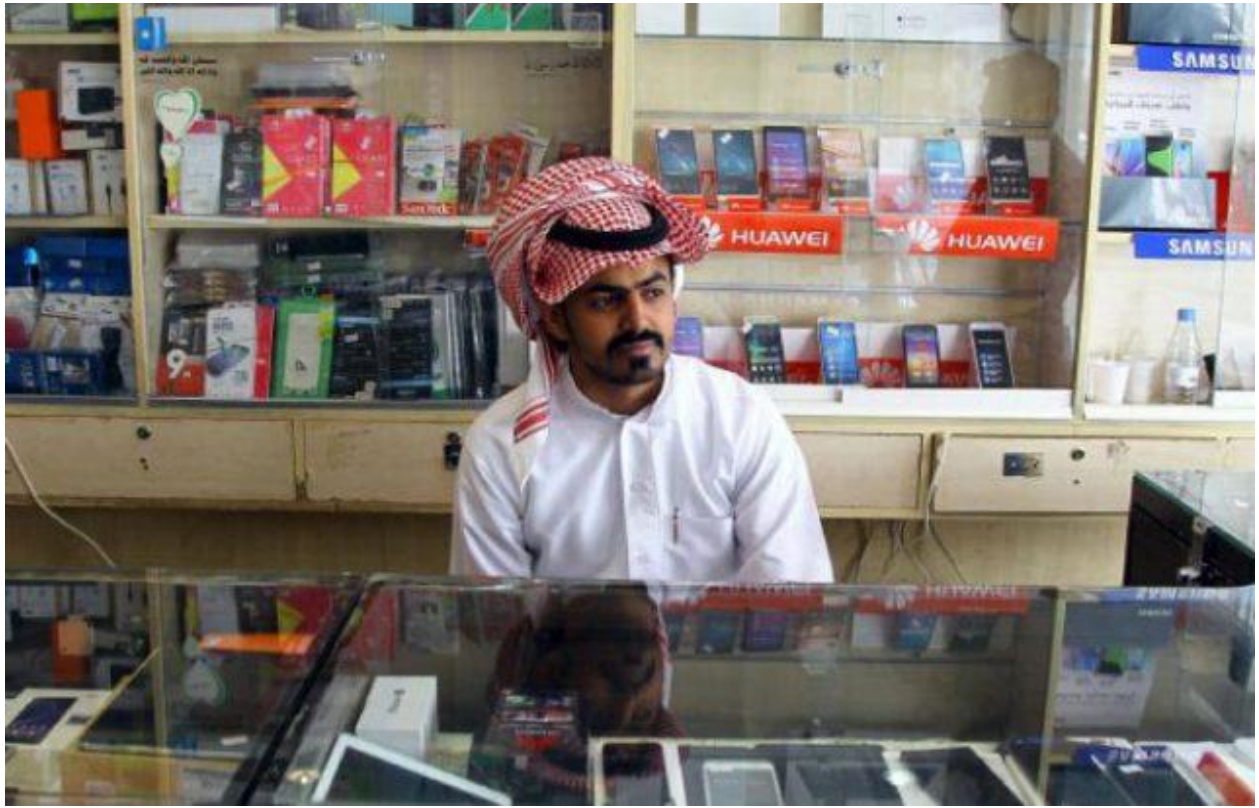
Мое облачное настоящее

RoboQA


A
Test Life

Senior QA

Девайс менеджер



Много рядовых тестировщиков

Что плохо при таком подходе

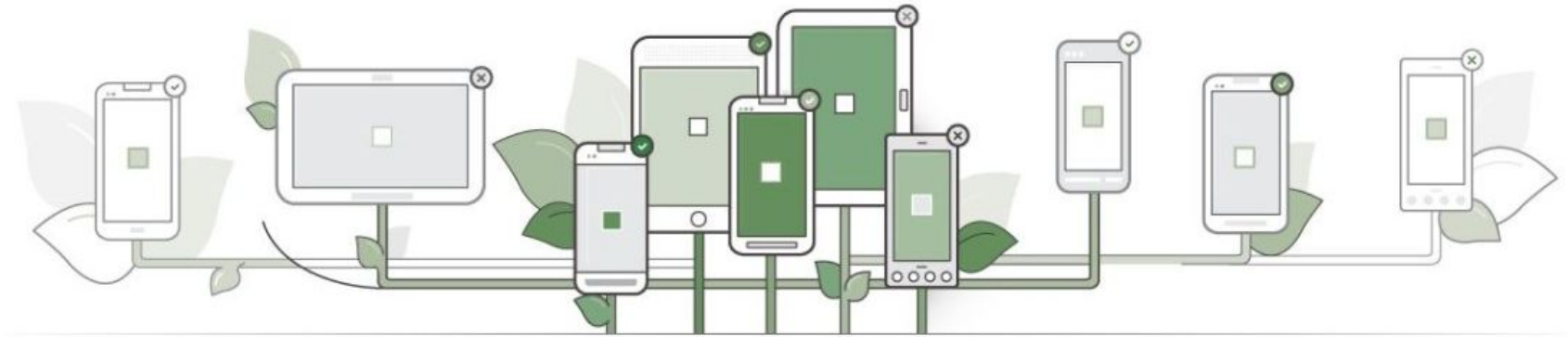
- Сложная структура команды и процессов
- Большие расходы на покупку устройств
- Расходы на содержание парка устройств
- Много людей для монотонной работы по прогону тест планов
- Ограниченная возможность параллельного выполнения тест кейсов
- Скорость итерации ограничена возможностями QA команды

Что хотелось бы видеть в идеале

- Высокий уровень качества приложения на устройствах с разными техническими характеристиками
- Запускать функциональные тесты на реальных устройствах
- Короткие итерации разработка - продакшен - фидбек
- Должно вписываться в разумный бюджет
- QA команда параллельно занимается несколькими проектами

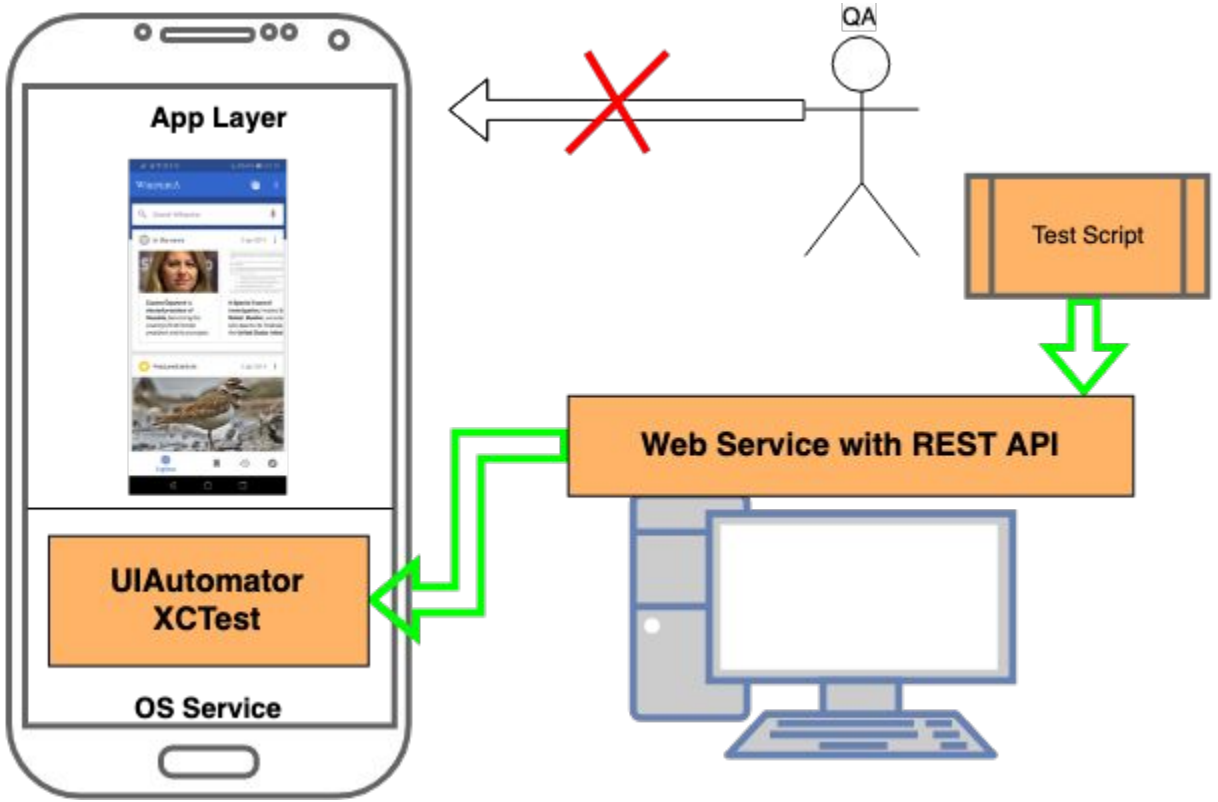
Скилы + современные технологии

Тестовый скрипт + Device Farm

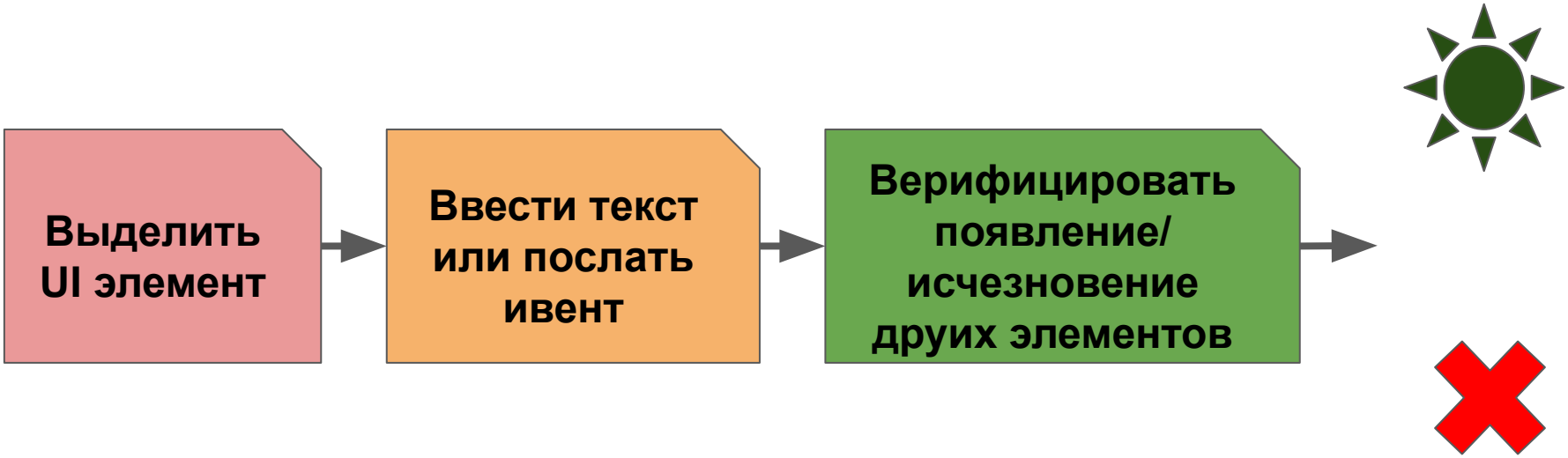


Web Service with REST API

Архитектура управляющего приложения



Элементы типового тест кейса



Популярные языки и библиотеки

- JavaScript
- Java
- C#
- Ruby
- PHP
- Python
- Perl

- JUnit
- NUnit

Когда все работает но медленно

- У нас уже есть тест планы и ручное тестирование
- QA перегружены рутинными задачами исполнения тест планов и верификацией пофикшеных багов
- Наши QA пишут тест планы но они не умеют программировать!
- Наши программисты не могут (не любят) тратить много времени на написание функциональных тестов!

Добавим Огурец

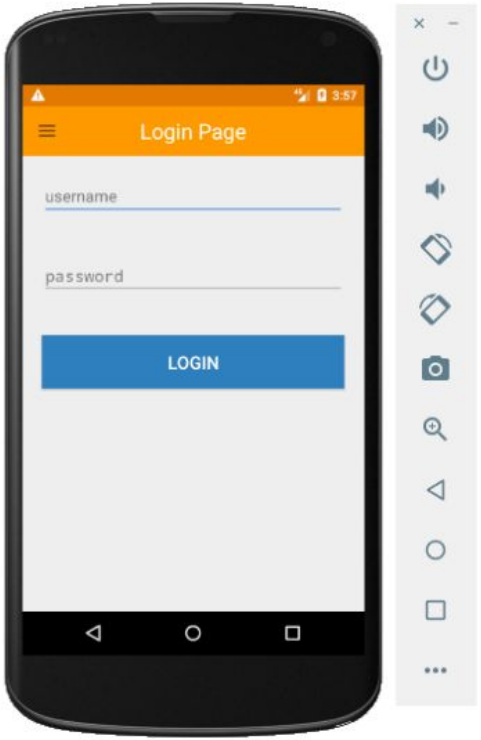
https://cucumber.io/ - Behaviour Driven Testing Framework

Разделяет скрипт на две части:

1. .feature файл с тест кейсом на естественном языке
2. программный код на языке программирования Java

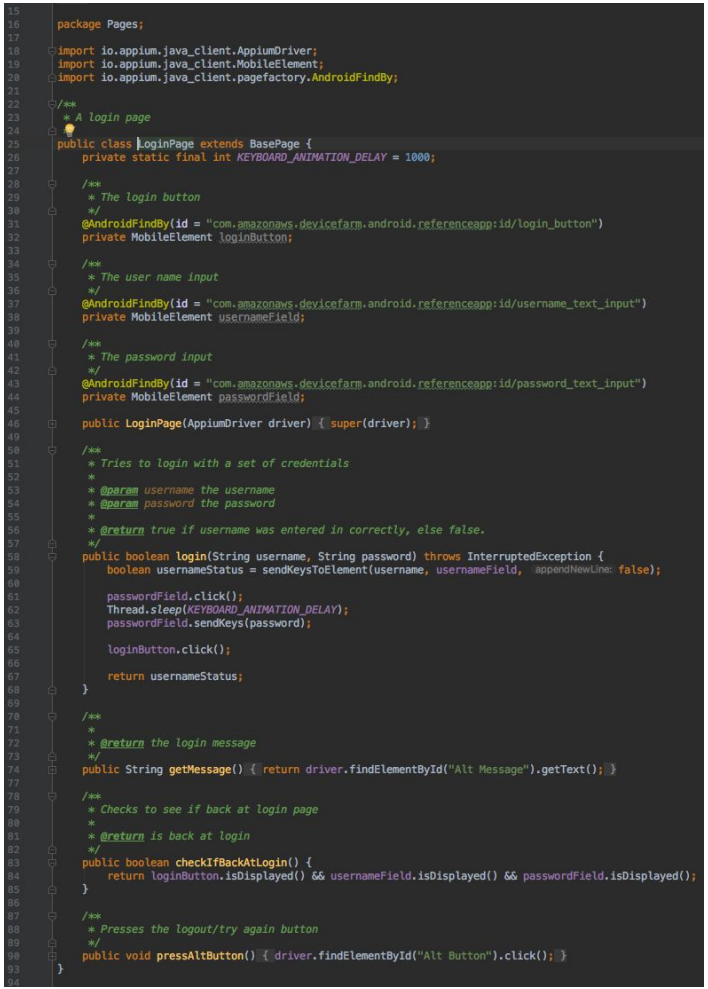
Занимаемся тестированием всей командой!

Тестируемое приложение

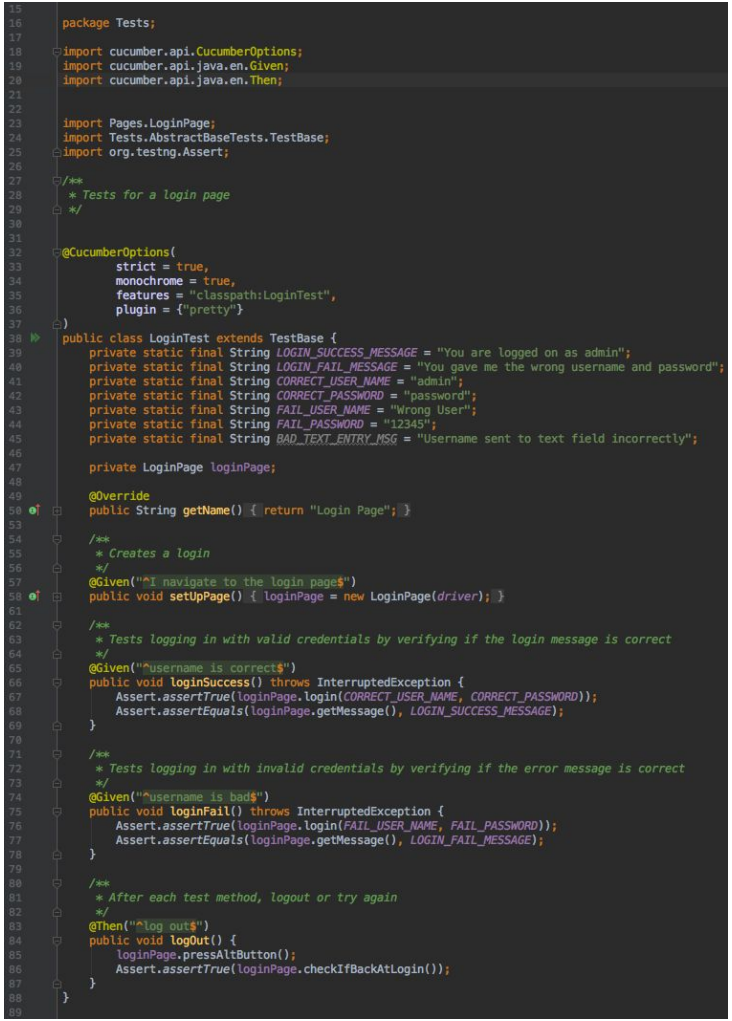


Описание тест кейса на естественном языке

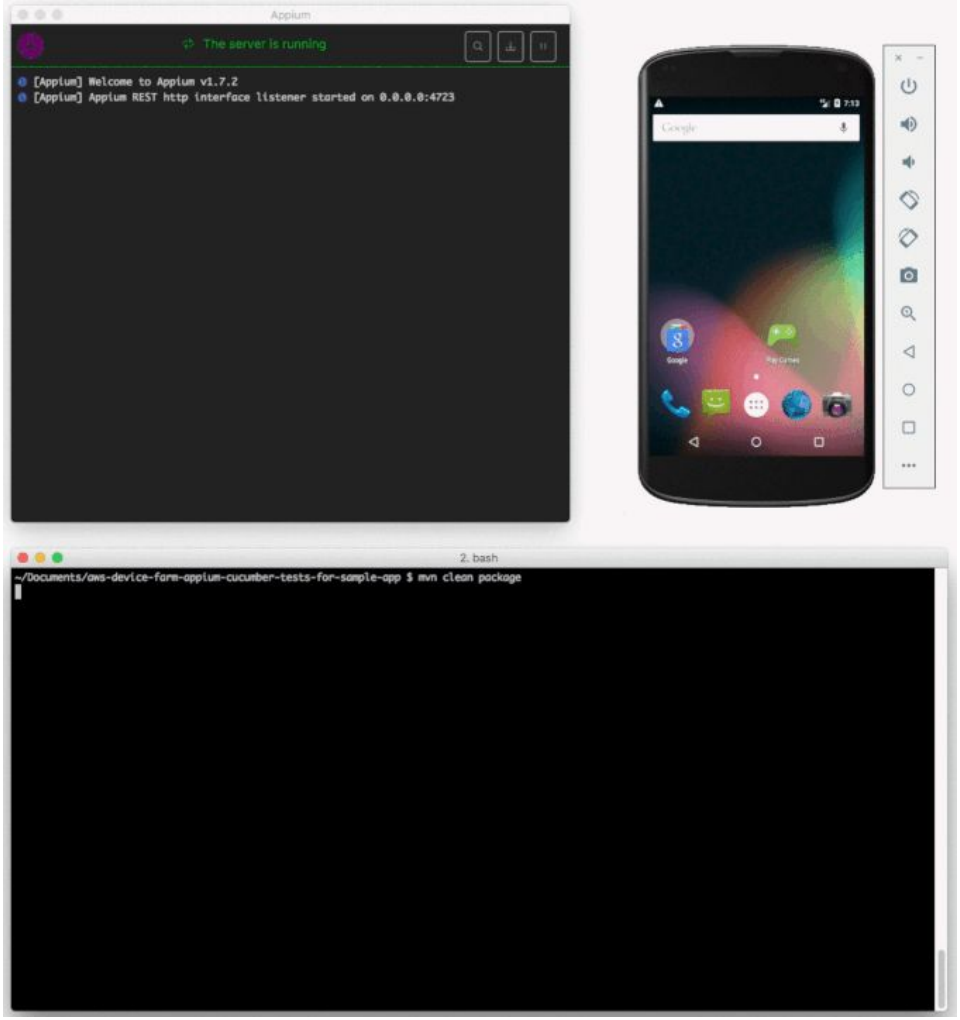
```
Feature: Does the login page work?
    We want the login page to work when the login credentials are valid, and fail otherwise

    Background: A Login Page
        Given I navigate to the login page

    Scenario: Login fails
        Given username is bad
        Then log out


    Scenario: Login succeeds
        Given username is correct
        Then log out
```



```java
package Pages;

import io.appium.java_client.AppiumDriver;
import io.appium.java_client.MobileElement;
import io.appium.java_client.pagefactory.AndroidFindBy;

/**
 * A login page
 */
public class LoginPage extends BasePage {
    private static final int KEYBOARD_ANIMATION_DELAY = 1000;

    /**
     * The login button
     */
    @AndroidFindBy(id = "com.amazonaws.devicefarm.android.referenceapp:id/login_button")
    private MobileElement loginButton;

    /**
     * The user name input
     */
    @AndroidFindBy(id = "com.amazonaws.devicefarm.android.referenceapp:id/username_text_input")
    private MobileElement usernameField;

    /**
     * The password input
     */
    @AndroidFindBy(id = "com.amazonaws.devicefarm.android.referenceapp:id/password_text_input")
    private MobileElement passwordField;

    public LoginPage(AppiumDriver driver) { super(driver); }

    /**
     * Tries to login with a set of credentials
     *
     * @param username the username
     * @param password the password
     *
     * @return true if username was entered in correctly, else false.
     */
    public boolean login(String username, String password) throws InterruptedException {
        boolean usernameStatus = sendKeysToElement(username, usernameField,  appendNewLine: false);

        passwordField.click();
        Thread.sleep(KEYBOARD_ANIMATION_DELAY);
        passwordField.sendKeys(password);

        loginButton.click();

        return usernameStatus;
    }

    /**
     *
     * @return the login message
     */
    public String getMessage() { return driver.findElementById("Alt Message").getText(); }

    /**
     * Checks to see if back at login page
     *
     * @return is back at login
     */
    public boolean checkIfBackAtLogin() {
        return loginButton.isDisplayed() && usernameField.isDisplayed() && passwordField.isDisplayed();
    }

    /**
     * Presses the logout/try again button
     */
    public void pressAltButton() { driver.findElementById("Alt Button").click(); }
}
```



```java
package Tests;

import cucumber.api.CucumberOptions;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;


import Pages.LoginPage;
import Tests.AbstractBaseTests.TestBase;
import org.testng.Assert;

/**
 * Tests for a login page
 */


@CucumberOptions(
        strict = true,
        monochrome = true,
        features = "classpath:LoginTest",
        plugin = {"pretty"}
)
public class LoginTest extends TestBase {
    private static final String LOGIN_SUCCESS_MESSAGE = "You are logged on as admin";
    private static final String LOGIN_FAIL_MESSAGE = "You gave me the wrong username and password";
    private static final String CORRECT_USER_NAME = "admin";
    private static final String CORRECT_PASSWORD = "password";
    private static final String FAIL_USER_NAME = "Wrong User";
    private static final String FAIL_PASSWORD = "12345";
    private static final String BAD_TEXT_ENTRY_MSG = "Username sent to text field incorrectly";

    private LoginPage loginPage;

    @Override
    public String getName() { return "Login Page"; }

    /**
     * Creates a login
     */
    @Given("^I navigate to the login page$")
    public void setUpPage() { loginPage = new LoginPage(driver); }

    /**
     * Tests logging in with valid credentials by verifying if the login message is correct
     */
    @Given("^username is correct$")
    public void loginSuccess() throws InterruptedException {
        Assert.assertTrue(loginPage.login(CORRECT_USER_NAME, CORRECT_PASSWORD));
        Assert.assertEquals(loginPage.getMessage(), LOGIN_SUCCESS_MESSAGE);
    }

    /**
     * Tests logging in with invalid credentials by verifying if the error message is correct
     */
    @Given("^username is bad$")
    public void loginFail() throws InterruptedException {
        Assert.assertTrue(loginPage.login(FAIL_USER_NAME, FAIL_PASSWORD));
        Assert.assertEquals(loginPage.getMessage(), LOGIN_FAIL_MESSAGE);
    }

    /**
     * After each test method, logout or try again
     */
    @Then("^log out$")
    public void logOut() {
        loginPage.pressAltButton();
        Assert.assertTrue(loginPage.checkIfBackAtLogin());
    }
}
```


Appium
The server is running
[Appium] Welcome to Appium v1.7.2
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
2. bash
~/Documents/aws-device-farm-appium-cucumber-tests-for-sample-app $ mvn clean package

Каждый занят любимым делом

Масштабируем с помощью AWS Device Farm

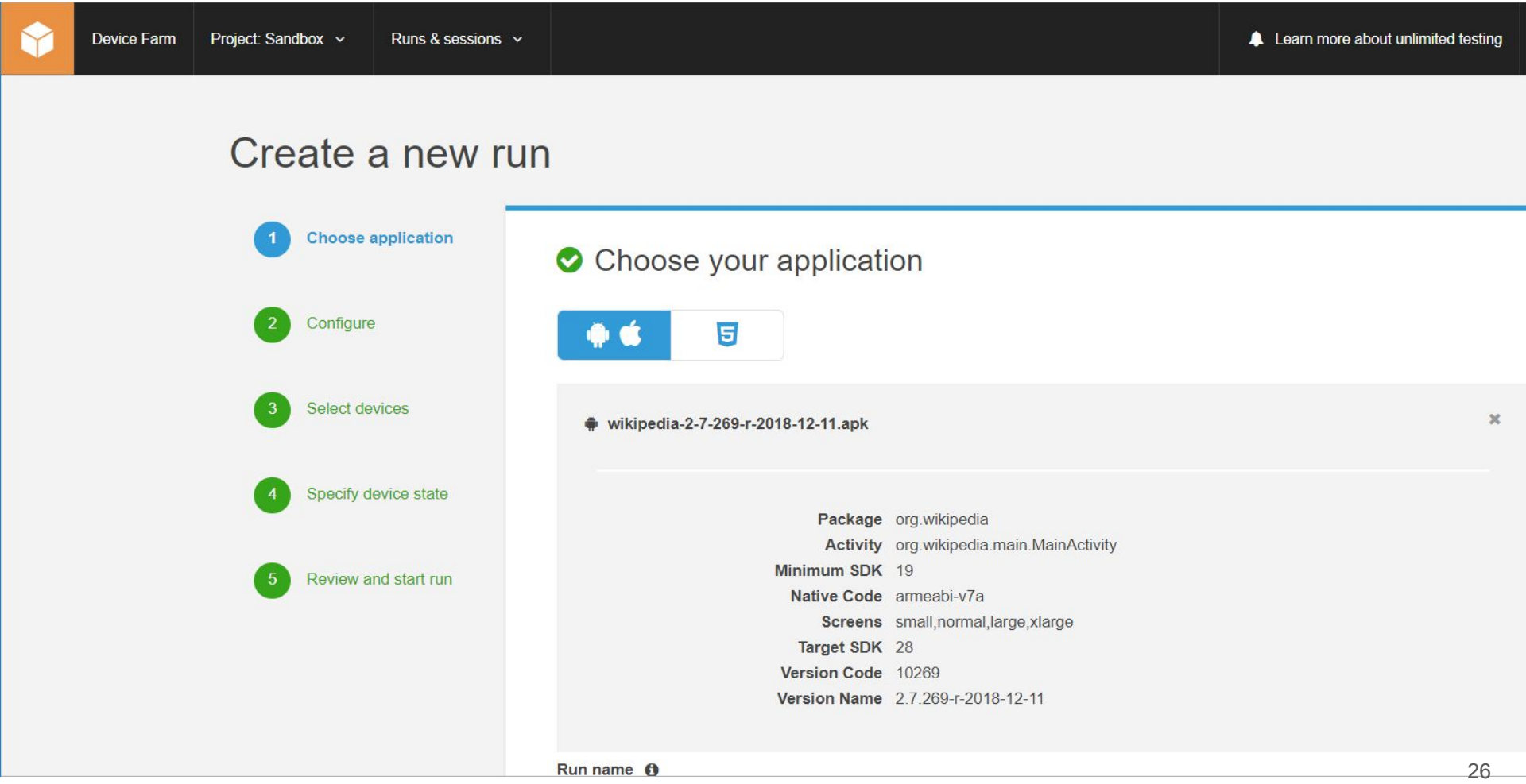
Создаем проект в Device Farm

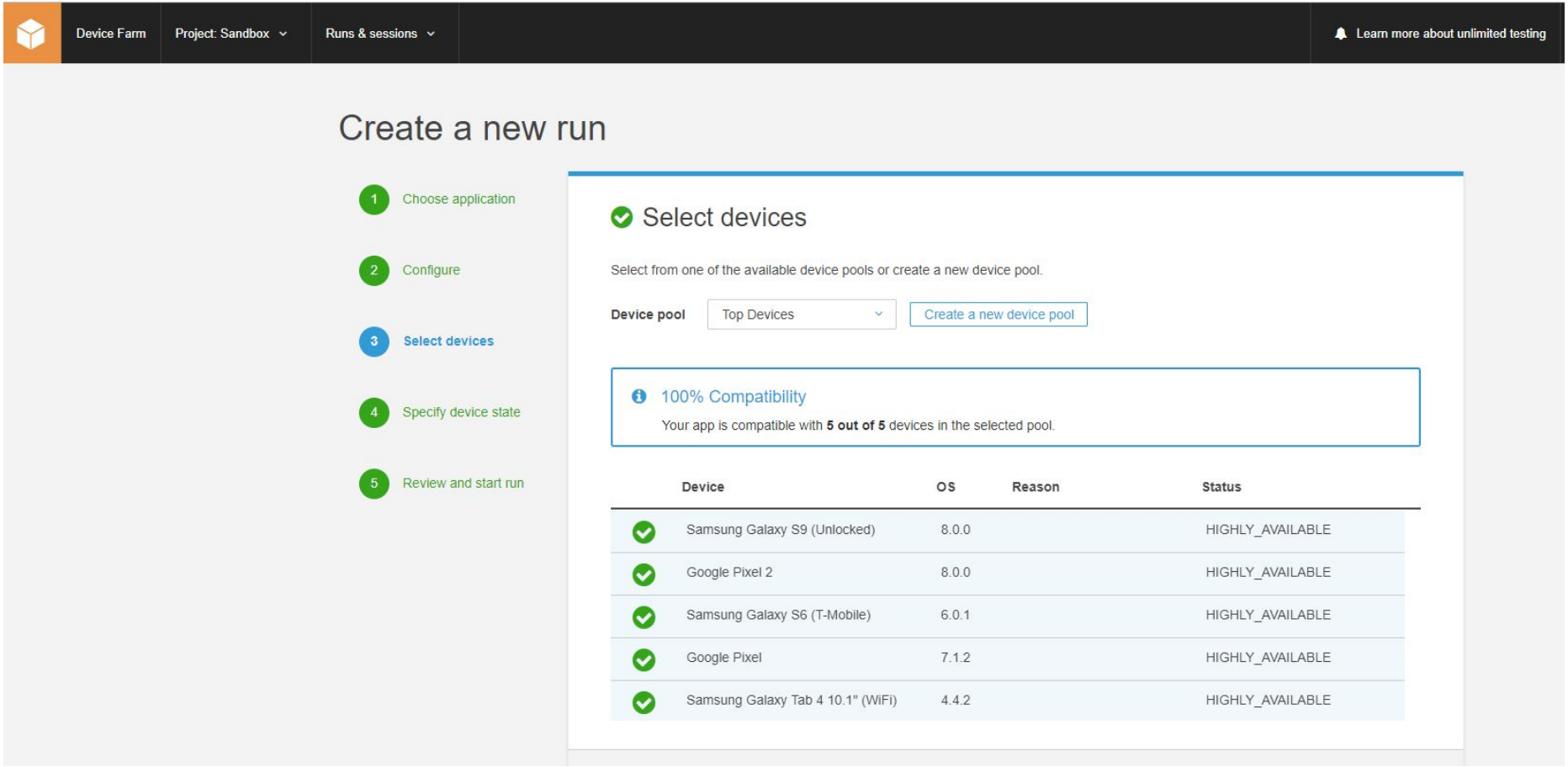
Собираем приложение -> Загружаем на веб сервис Device Farm

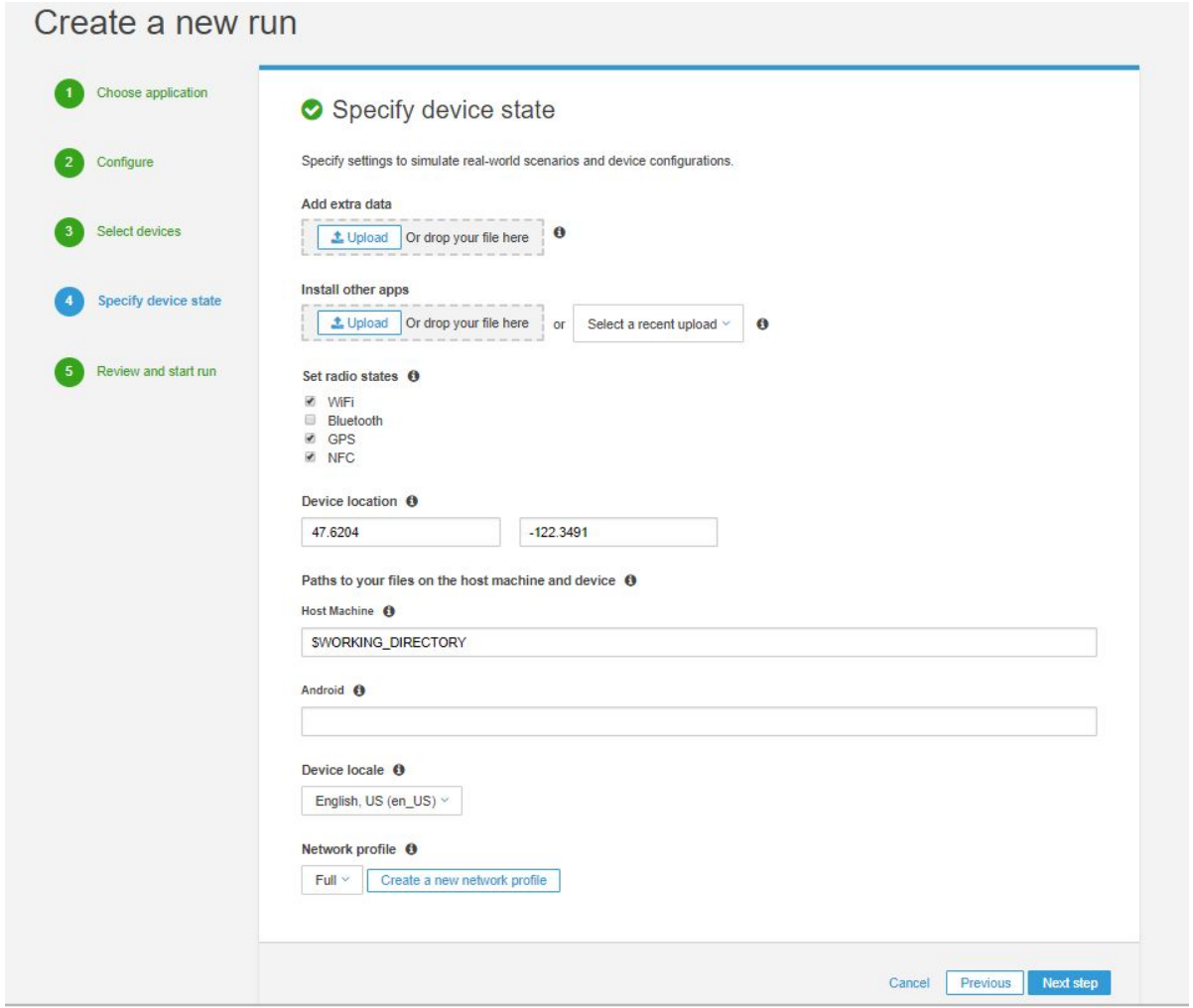
Собираем приложение с тестами -> Upload to Device Farm service

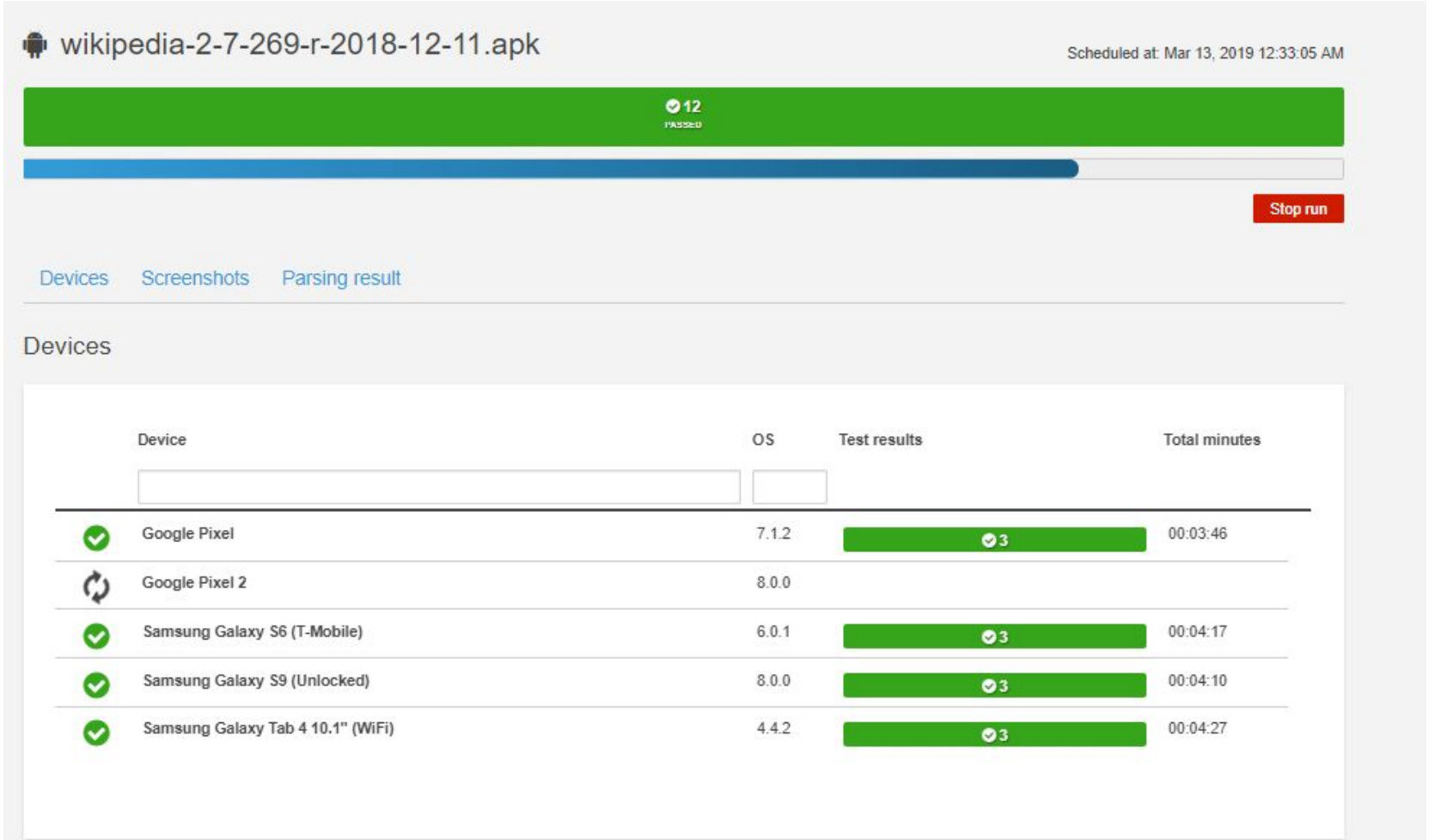
Выбираем целевые устройства

Запускаем выполнение -> Получаем результаты валидации (логи, скриншоты, видео)


Device Farm
Project: Sandbox
Runs & sessions
Learn more about unlimited testing
Create a new run
1 Choose application
2 Configure
3 Select devices
4 Specify device state
5 Review and start run
Choose your application
wikipedia-2-7-269-r-2018-12-11.apk
Package org.wikipedia
Activity org.wikipedia.main.MainActivity
Minimum SDK 19
Native Code armeabi-v7a
Screens small,normal,large,xlarge
Target SDK 28
Version Code 10269
Version Name 2.7.269-r-2018-12-11
Run name


Device Farm
Project: Sandbox
Runs & sessions
Learn more about unlimited testing
Create a new run
1 Choose application
2 Configure
3 Select devices
4 Specify device state
5 Review and start run
Select devices
Select from one of the available device pools or create a new device pool.
Device pool
Top Devices
Create a new device pool
100% Compatibility
Your app is compatible with 5 out of 5 devices in the selected pool.
Device
OS
Reason
Status
Samsung Galaxy S9 (Unlocked) 8.0.0 HIGHLY_AVAILABLE
Google Pixel 2 8.0.0 HIGHLY_AVAILABLE
Samsung Galaxy S6 (T-Mobile) 6.0.1 HIGHLY_AVAILABLE
Google Pixel 7.1.2 HIGHLY_AVAILABLE
Samsung Galaxy Tab 4 10.1" (WiFi) 4.4.2 HIGHLY_AVAILABLE

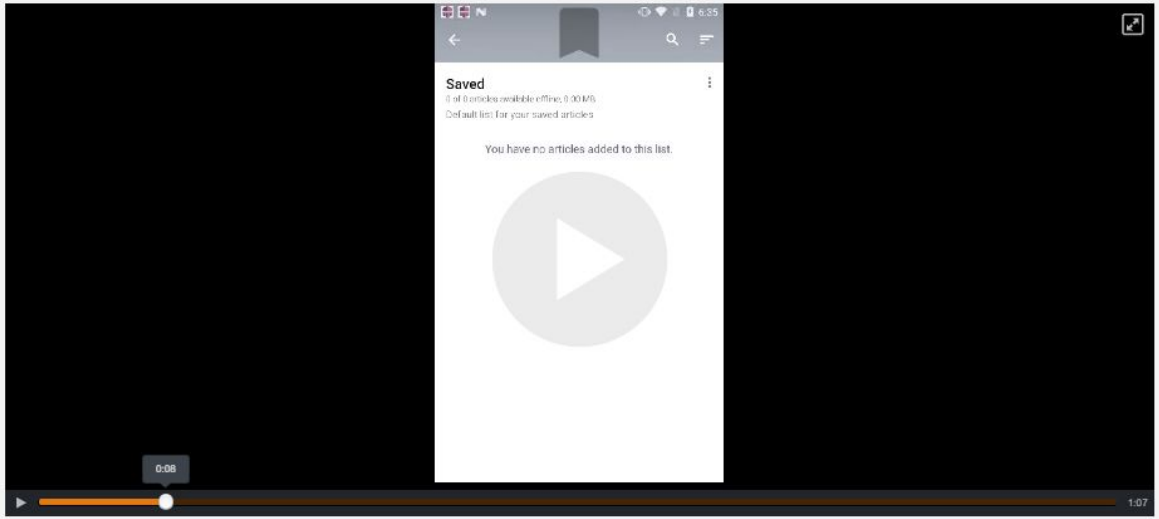

Create a new run
1 Choose application
2 Configure
3 Select devices
4 Specify device state
5 Review and start run
Specify device state
Specify settings to simulate real-world scenarios and device configurations.
Add extra data
Upload
Or drop your file here
Install other apps
Upload
Or drop your file here
or
Select a recent upload
Set radio states
WiFi
Bluetooth
GPS
NFC
Device location
47.6204
-122.3491
Paths to your files on the host machine and device
Host Machine
$WORKING_DIRECTORY
Android
Device locale
English, US (en_US)
Network profile
Full
Create a new network profile
Cancel
Previous
Next step


wikipedia-2-7-269-r-2018-12-11.apk
Scheduled at: Mar 13, 2019 12:33:05 AM
12
PASSED
Stop run
Devices
Screenshots
Parsing result
Devices
Device
OS
Test results
Total minutes
Google Pixel
7.1.2
3
00:03:46
Google Pixel 2
8.0.0
Samsung Galaxy S6 (T-Mobile)
6.0.1
3
00:04:17
Samsung Galaxy S9 (Unlocked)
8.0.0
3
00:04:10
Samsung Galaxy Tab 4 10.1" (WiFi)
4.4.2
3
00:04:27

Video Suites Logs Performance Files Screenshots

Video

Download video



Suites

Suite	OS	Test results	Total minutes
Setup Suite		1	00:01:13
Built-in Fuzz Suite		1	00:01:41
Teardown Suite		1	00:00:51

Начальный уровень

Firebase Test Lab
for Android

★ Бесплатные или очень дешевая подписка
★ Можно найти специфические устройства (например часы)
❑ Ограниченные возможности API
❑ Мало типов устройств или устройства одного производителя

Продвинутый уровень

- ★ Много устройств (несколько тысяч)
- ★ Очень развитое АПИ
  - Позволяет манипулировать пулами устройств
- ★ Много настроек
  - Подключение к различным сотовым операторам или WiFi
  - Контроль геолокации устройства
- ★ Интеграции с популярными таск/баг трекерами и CI/CD платформами
- ❑ Цена подписки более высокая чем в начальном уровне

Сколько стоит качество в вашей компании?

Unnamed BigCompany

ЗП Senior QA: 20 EUR/ч * 160 ч/мес

ЗП 3 Junior QA: 10 EUR/ч * 160 ч/мес

Device Manager: 10 EUR/ч * 160 ч/мес

Бюджет на устройства: 700EUR * 20 шт = 14000 EUR

Начальные инвестиции: 14000 EUR

Ежемесячные платежи: 9600 EUR

Unnamed SmartCompany

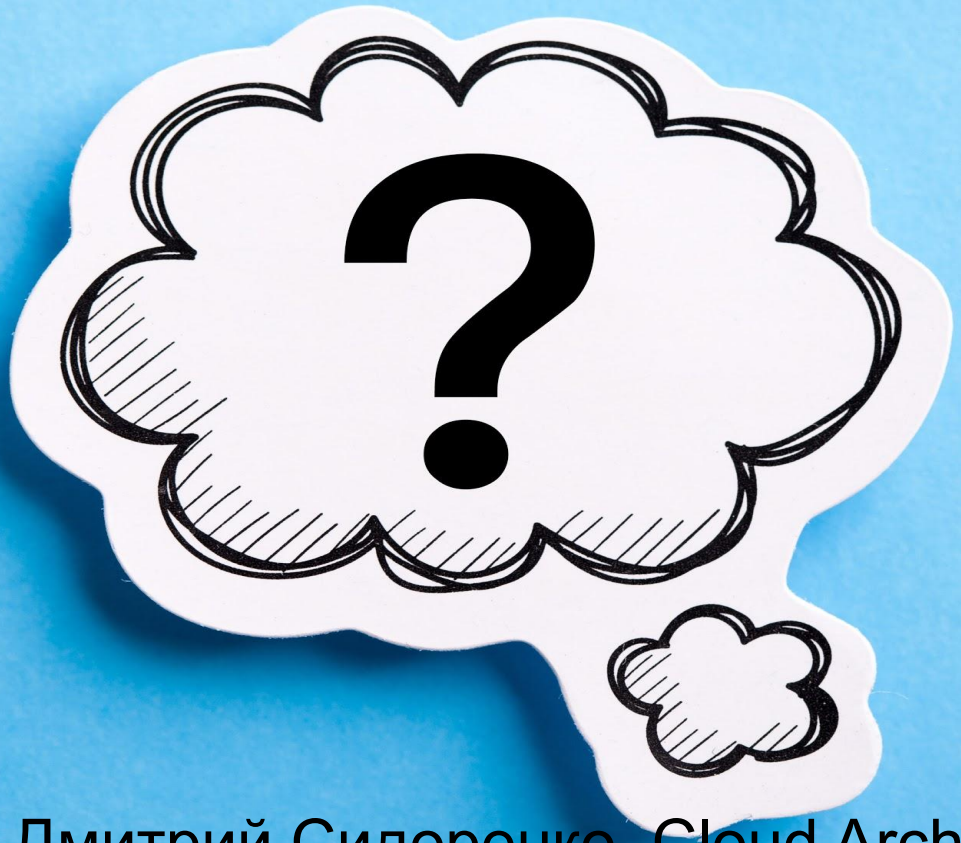
Senior QA: 20 EUR/ч * 160ч

Programmer: 35 EUR/ч * 40ч

Подписка на Device Farm: 200 EUR/мес

Начальные инвестиции: 0 EUR

Ежемесячные платежи: 4600 EUR



Дмитрий Сидоренко, Cloud Architect
Nordcloud/RoboQA, ds@robo.qa