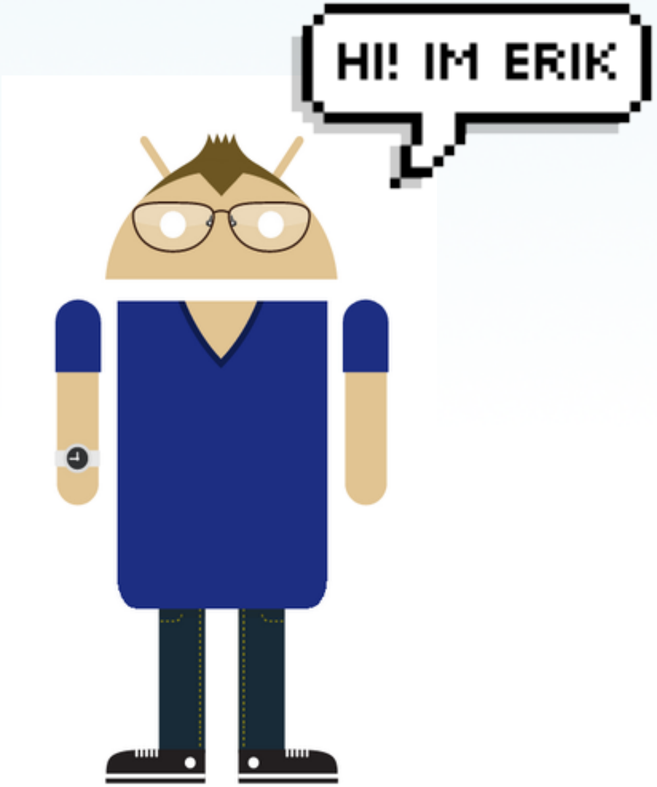
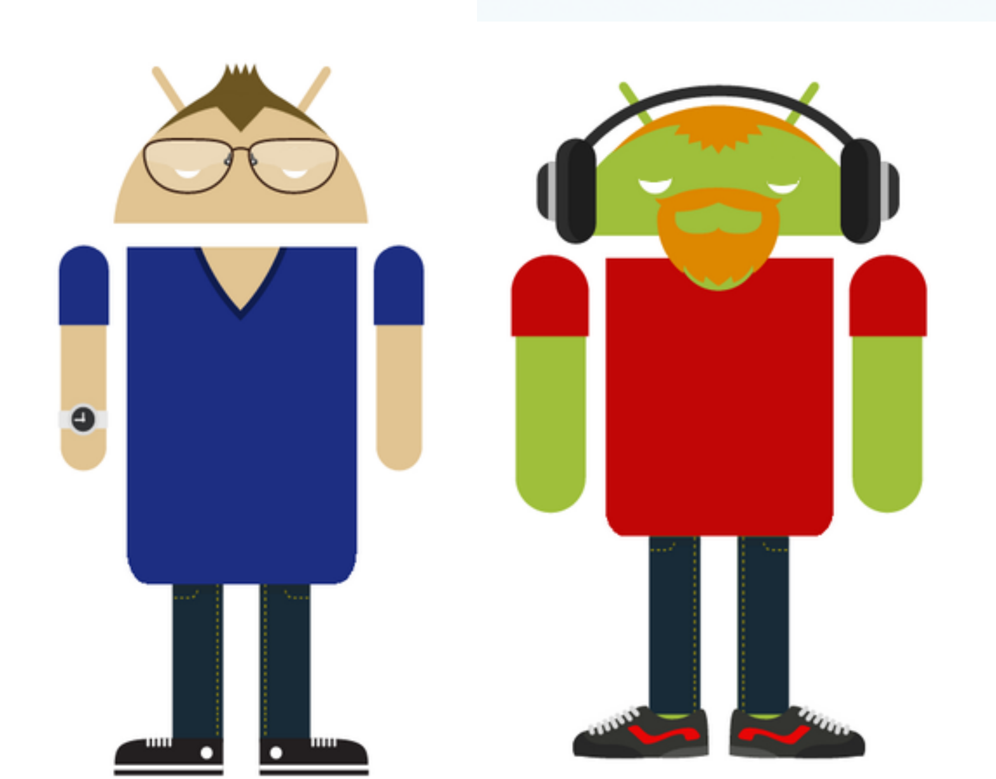
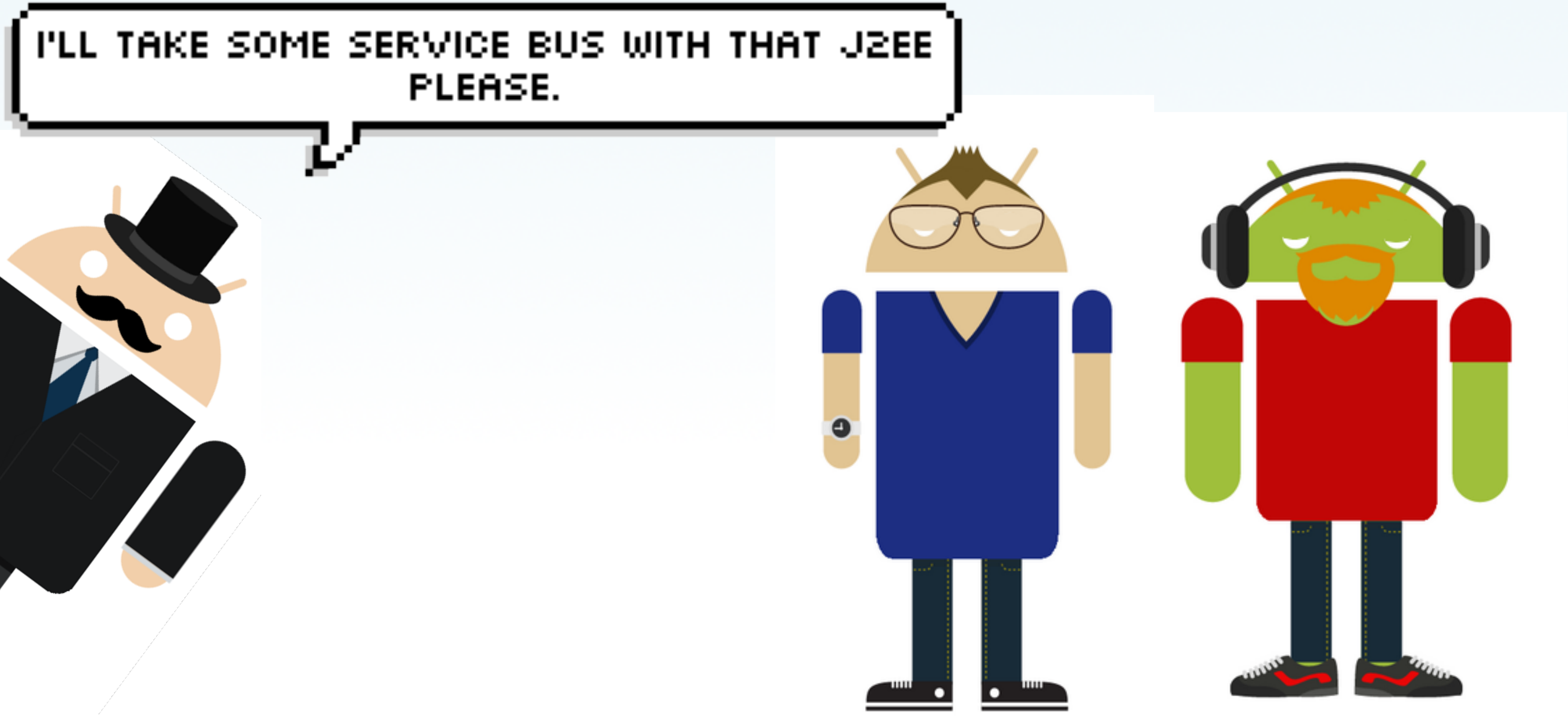
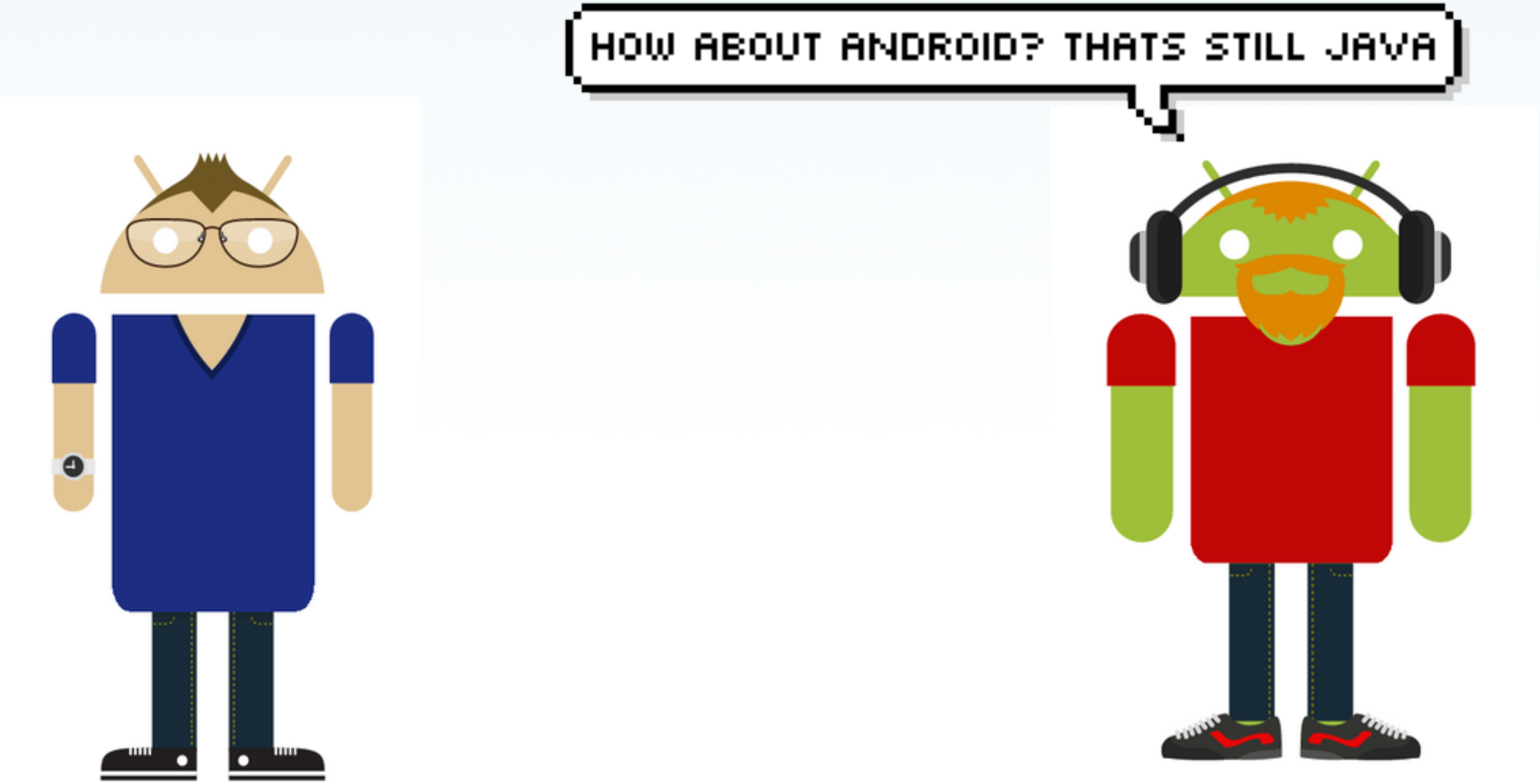


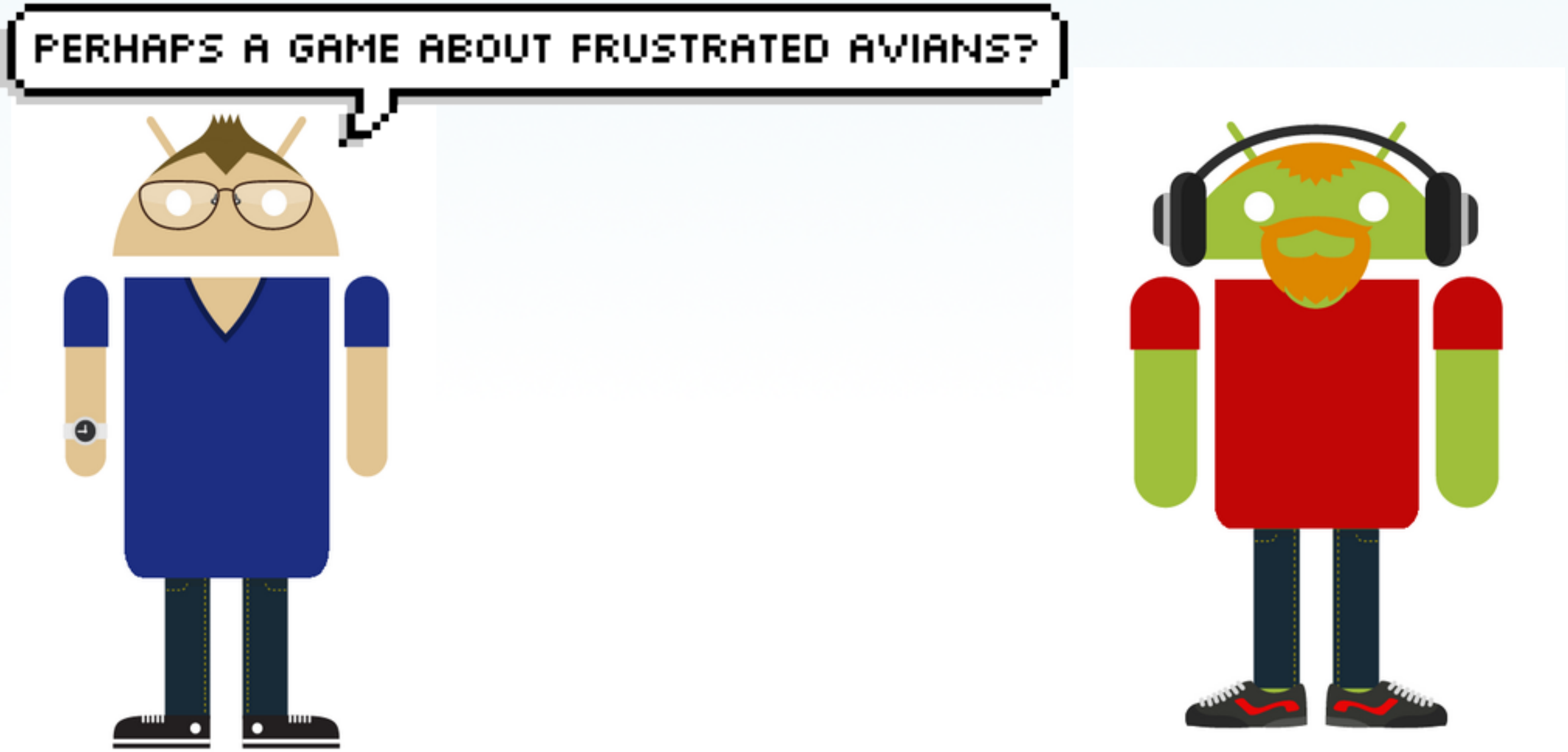
LESSPAINFUL
MOBILE APP TESTING

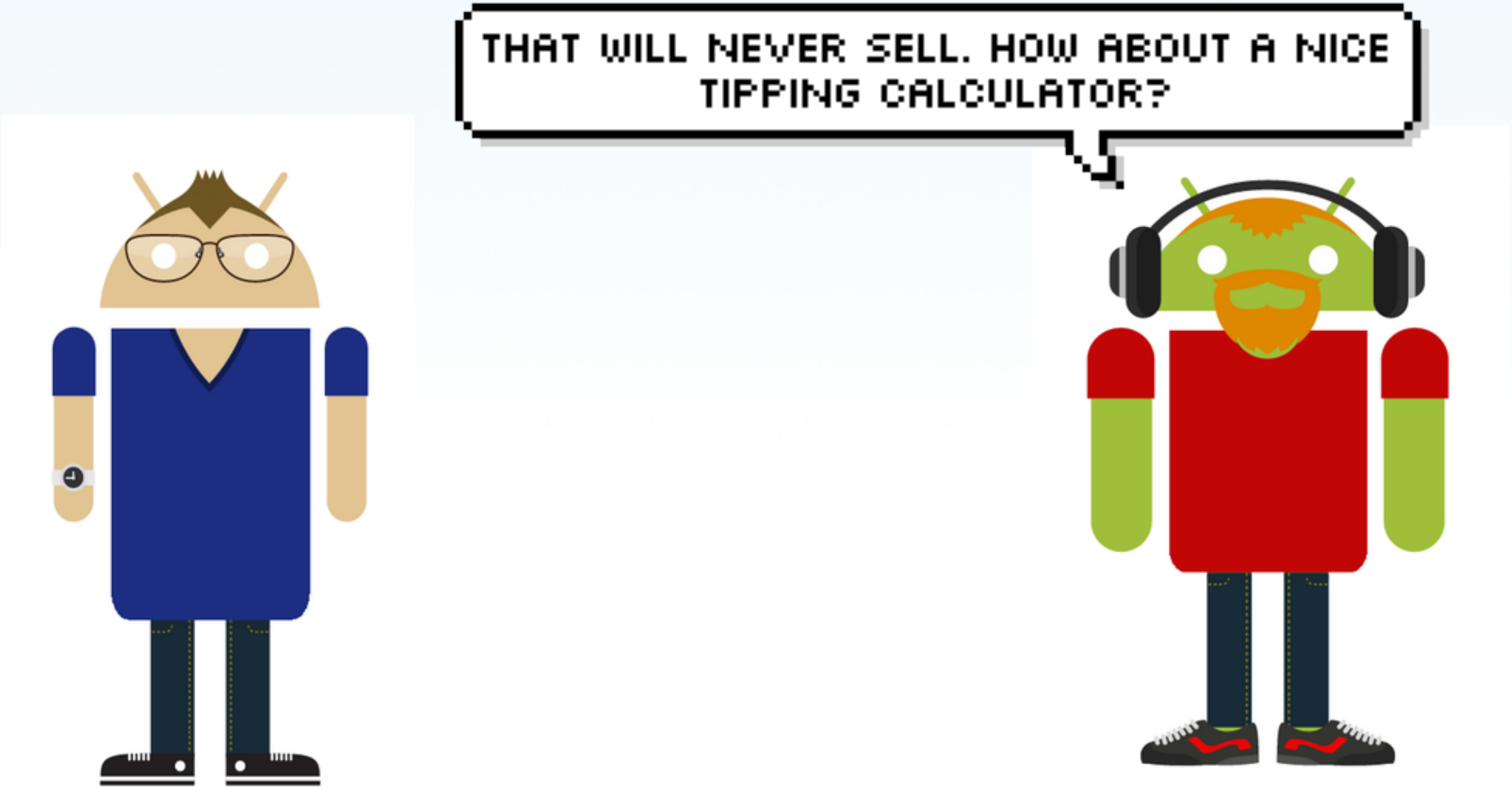


LESSPAINFUL
MOBILE APP TESTING

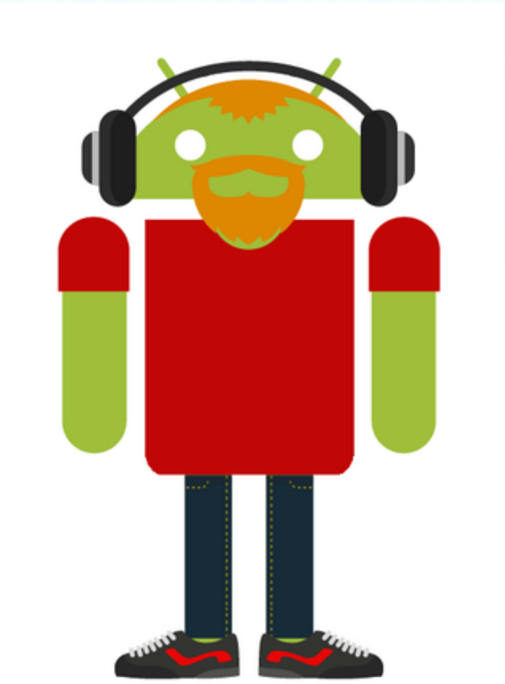

I'LL TAKE SOME SERVICE BUS WITH THAT J2EE PLEASE.


HOW ABOUT ANDROID? THATS STILL JAVA


PERHAPS A GAME ABOUT FRUSTRATED AVIANS?



LESSPAINFUL
MOBILE APP TESTING



LESSPAINFUL
MOBILE APP TESTING

Why should I automate my UI test?

Why should I automate my UI test?

- To get less bugs

Why should I automate my UI test?

- To get less bugs
- To get fast feedback - part of CI

Why should I automate my UI test?

- To get less bugs
- To get fast feedback - part of CI
- To get consistency

Why should I automate my UI test?

- To get less bugs
- To get fast feedback - part of CI
- To get consistency
- Because manual testing is mind-numbing

Introduction to the Sdk tools

Introduction to the Sdk tools

- Test projects
  - Eclipse

Introduction to the Sdk tools

Test projects

- Eclipse
- android create test-project

Introduction to the Sdk tools

Test projects

- Eclipse
- android create test-project
- Tests one specific app

Introduction to the Sdk tools

Test projects

- Eclipse
- android create test-project
- Tests one specific app

Instrumentation / InstrumentationTestRunner

Introduction to the Sdk tools

Test projects

- Eclipse
- android create test-project
- Tests one specific app

Instrumentation / InstrumentationTestRunner

TestCase classes

- Based on JUnit**3**

Your options are...

Your options are...

ActivityUnitTestCase

Your options are...

ActivityUnitTestCase

- Allows system mocks
  - setActivityContext
  - setApplication

Your options are...

ActivityUnitTestCase

- Allows system mocks
  - setActivityContext
  - setApplication

- Activity isolated from the system
  - So no:
    - startActivity
    - isTaskRoot
    - ...

Your options are...

ActivityUnitTestCase

- Allows system mocks
  - setActivityContext
  - setApplication

- Activity isolated from the system
  - So no:
    - startActivity
    - isTaskRoot

- Could be useful for testing lifecycle events

Your options are...

ActivityInstrumentationTestCase2

Your options are...

ActivityInstrumentationTestCase2

For integtration/system testing

Your options are...

ActivityInstrumentationTestCase2

- For integtration/system testing
- No system mocks

Your options are...

ActivityInstrumentationTestCase2

- For integtration/system testing
- No system mocks
- Not isolated from system

From here on out

System/Integration tests

From here on out

System/Integration tests... great!

From here on out

System/Integration tests... great!

...but slow!

From here on out

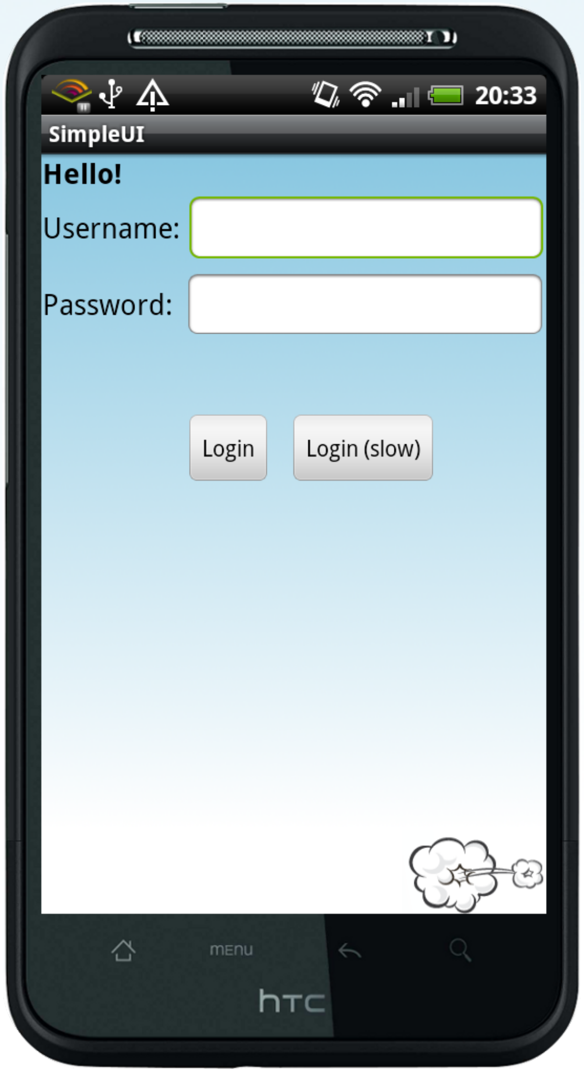
System/Integration tests... great!

...but slow!

So Unit test what you can.

A test

Enter username
Enter password
Press login
Verify greeting "Welcome!"

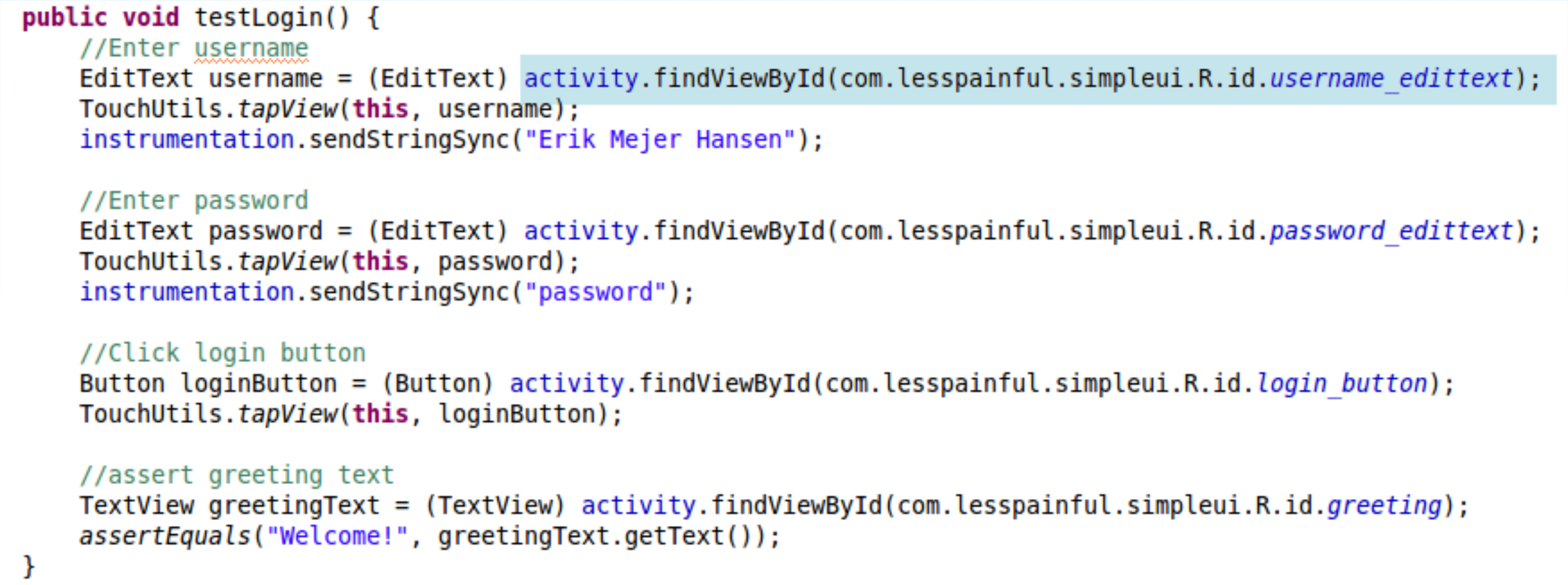

20:33

SimpleUI

Hello!

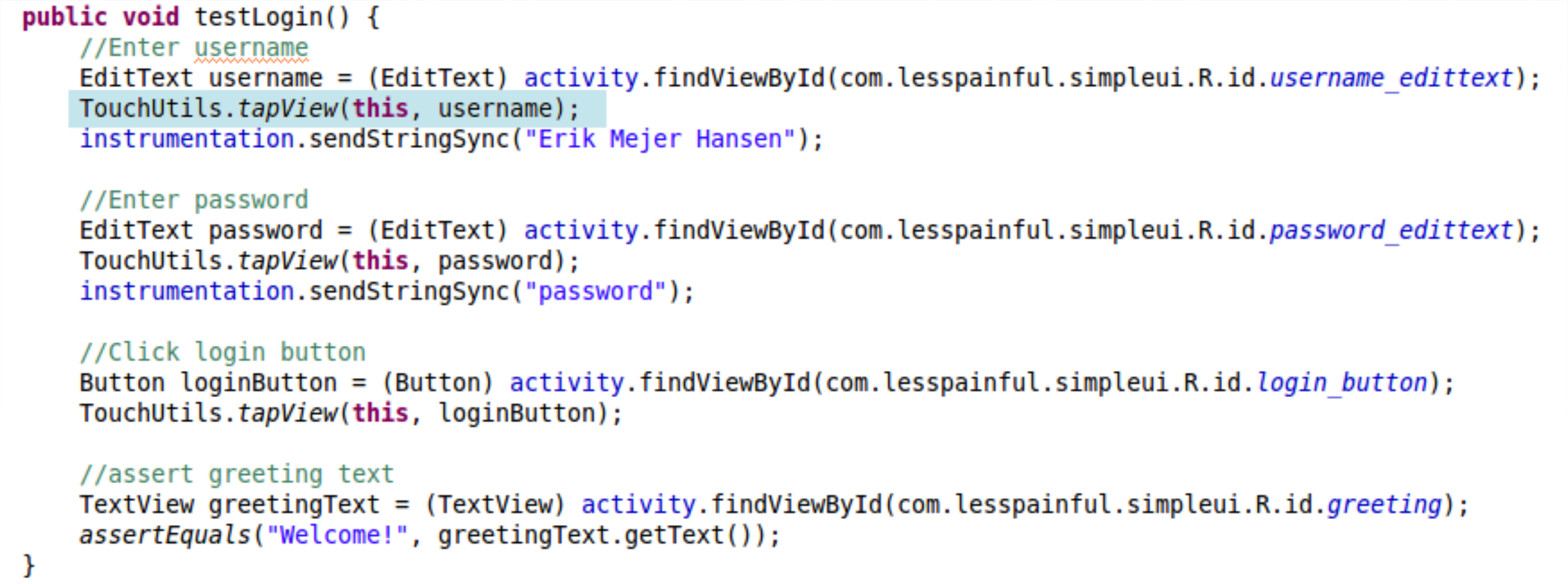
Username:

Password:



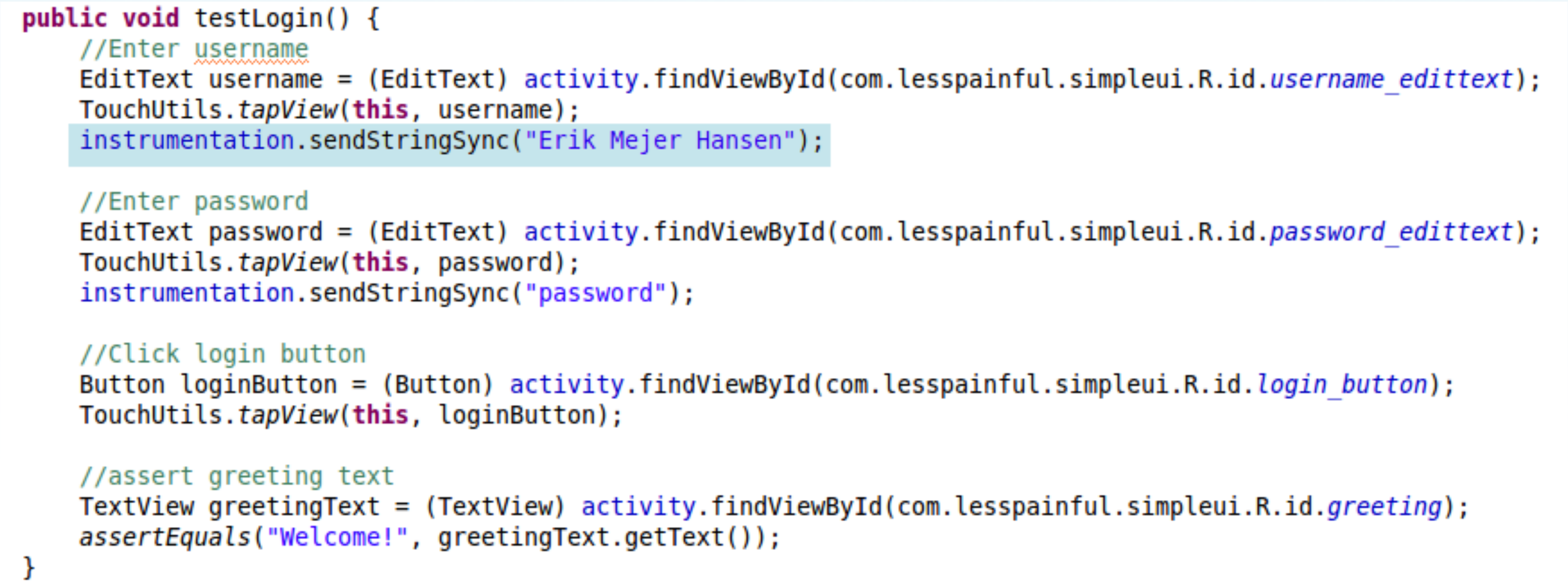
```java
public void testLogin() {
    //Enter username
    EditText username = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.username_edittext);
    TouchUtils.tapView(this, username);
    instrumentation.sendStringSync("Erik Mejer Hansen");

    //Enter password
    EditText password = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.password_edittext);
    TouchUtils.tapView(this, password);
    instrumentation.sendStringSync("password");

    //Click login button
    Button loginButton = (Button) activity.findViewById(com.lesspainful.simpleui.R.id.login_button);
    TouchUtils.tapView(this, loginButton);

    //assert greeting text
    TextView greetingText = (TextView) activity.findViewById(com.lesspainful.simpleui.R.id.greeting);
    assertEquals("Welcome!", greetingText.getText());
}
```

LESSPAINFUL
MOBILE APP TESTING

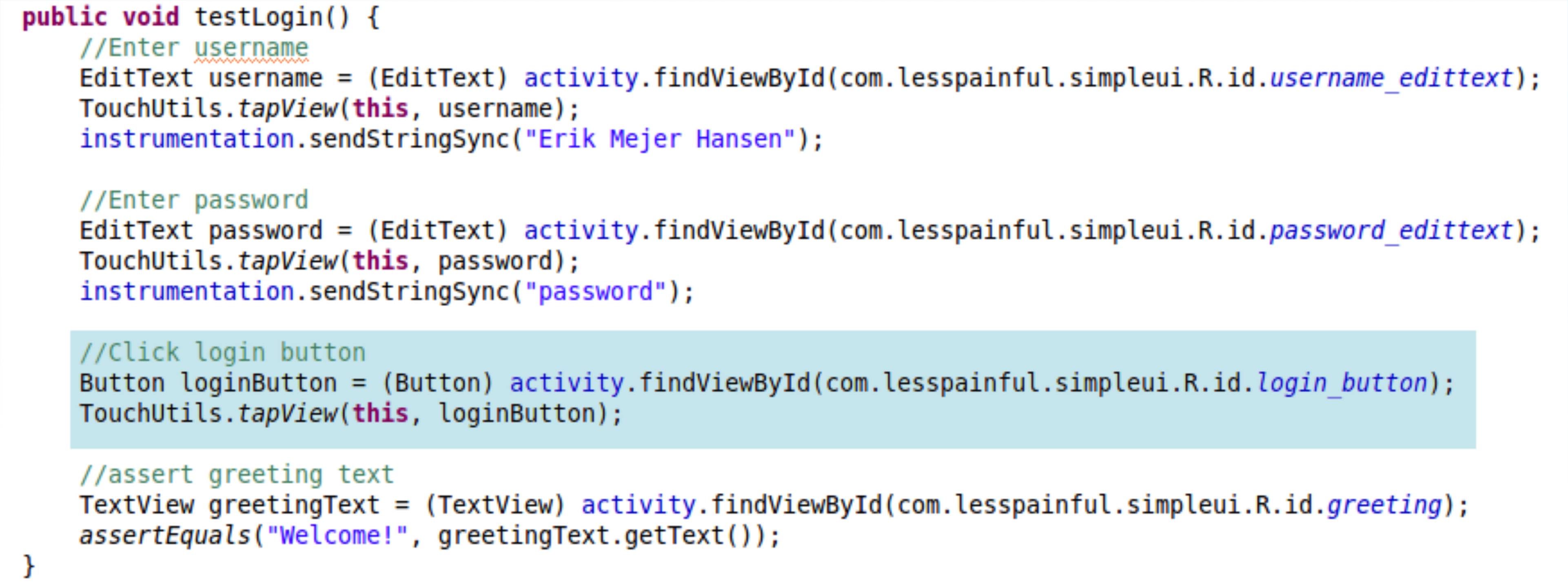


```
public void testLogin() {
    //Enter username
    EditText username = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.username_edittext);
    TouchUtils.tapView(this, username);
    instrumentation.sendStringSync("Erik Mejer Hansen");

    //Enter password
    EditText password = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.password_edittext);
    TouchUtils.tapView(this, password);
    instrumentation.sendStringSync("password");

    //Click login button
    Button loginButton = (Button) activity.findViewById(com.lesspainful.simpleui.R.id.login_button);
    TouchUtils.tapView(this, loginButton);

    //assert greeting text
    TextView greetingText = (TextView) activity.findViewById(com.lesspainful.simpleui.R.id.greeting);
    assertEquals("Welcome!", greetingText.getText());
}
```

LESSPAINFUL
MOBILE APP TESTING

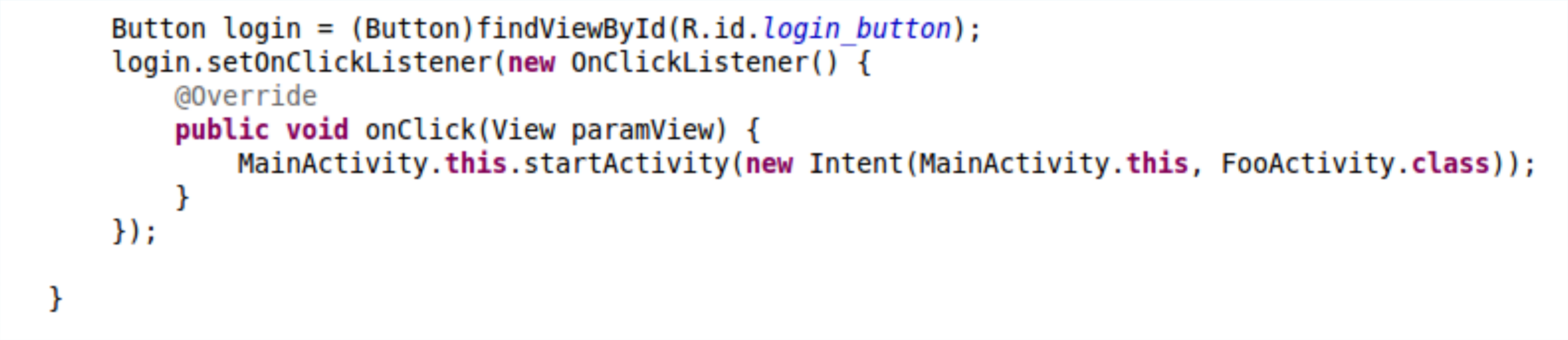


```
public void testLogin() {
    //Enter username
    EditText username = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.username_edittext);
    TouchUtils.tapView(this, username);
    instrumentation.sendStringSync("Erik Mejer Hansen");

    //Enter password
    EditText password = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.password_edittext);
    TouchUtils.tapView(this, password);
    instrumentation.sendStringSync("password");

    //Click login button
    Button loginButton = (Button) activity.findViewById(com.lesspainful.simpleui.R.id.login_button);
    TouchUtils.tapView(this, loginButton);

    //assert greeting text
    TextView greetingText = (TextView) activity.findViewById(com.lesspainful.simpleui.R.id.greeting);
    assertEquals("Welcome!", greetingText.getText());
}
```



```java
public void testLogin() {
    //Enter username
    EditText username = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.username_edittext);
    TouchUtils.tapView(this, username);
    instrumentation.sendStringSync("Erik Mejer Hansen");

    //Enter password
    EditText password = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.password_edittext);
    TouchUtils.tapView(this, password);
    instrumentation.sendStringSync("password");

    //Click login button
    Button loginButton = (Button) activity.findViewById(com.lesspainful.simpleui.R.id.login_button);
    TouchUtils.tapView(this, loginButton);

    //assert greeting text
    TextView greetingText = (TextView) activity.findViewById(com.lesspainful.simpleui.R.id.greeting);
    assertEquals("Welcome!", greetingText.getText());
}
```

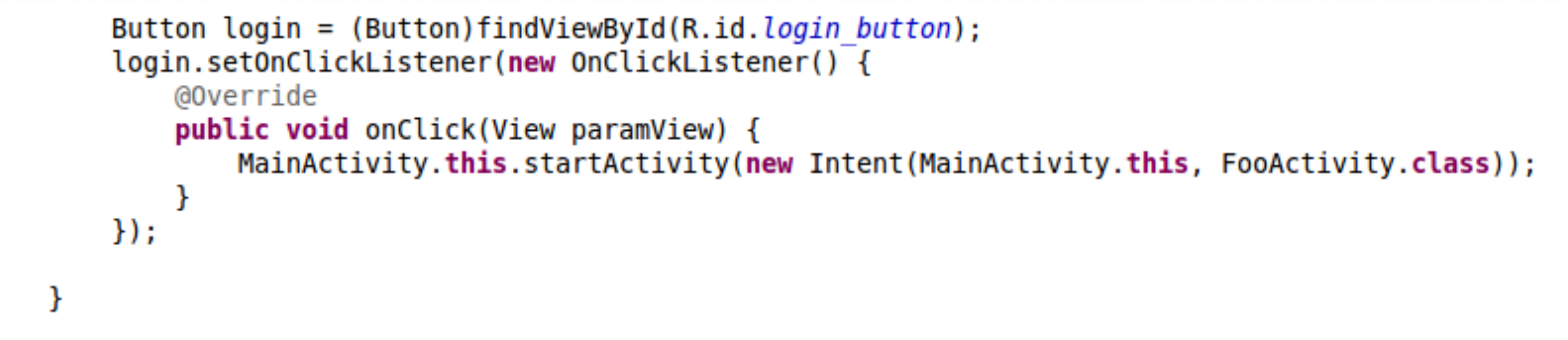
What happens now?



```
        Button login = (Button)findViewById(R.id.login_button);
        login.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View paramView) {
                MainActivity.this.startActivity(new Intent(MainActivity.this, FooActivity.class));
            }
        });

    }
```

LESSPAINFUL
MOBILE APP TESTING

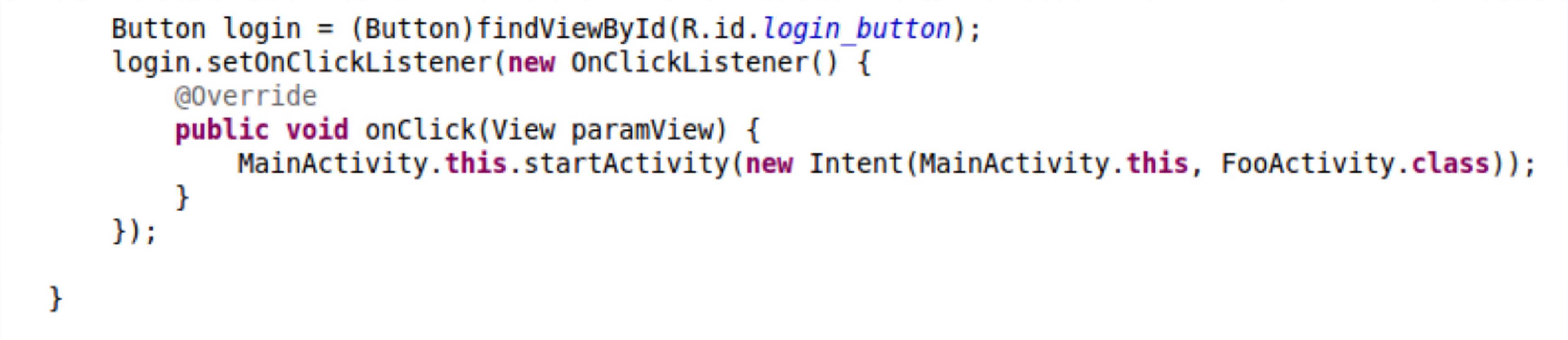
What happens now?



```
    Button login = (Button)findViewById(R.id.login_button);
    login.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View paramView) {
            MainActivity.this.startActivity(new Intent(MainActivity.this, FooActivity.class));
        }
    });

}
```

No probem right?

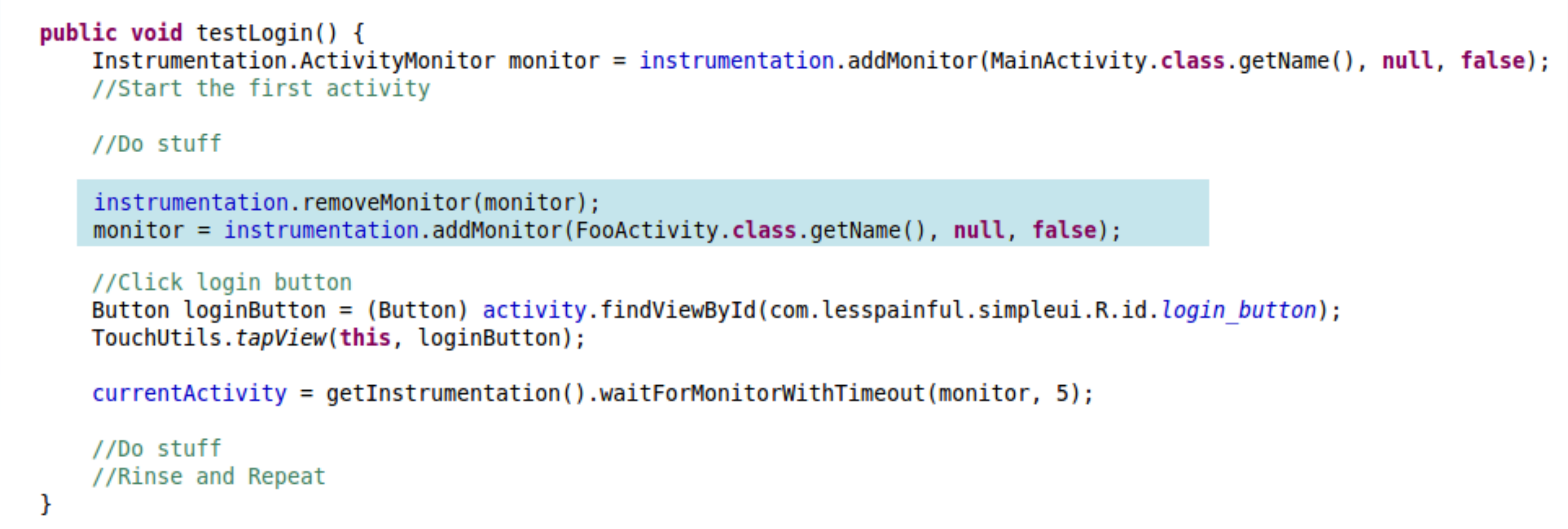
What happens now?



```
        Button login = (Button)findViewById(R.id.login_button);
        login.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View paramView) {
                MainActivity.this.startActivity(new Intent(MainActivity.this, FooActivity.class));
            }
        });

    }
```

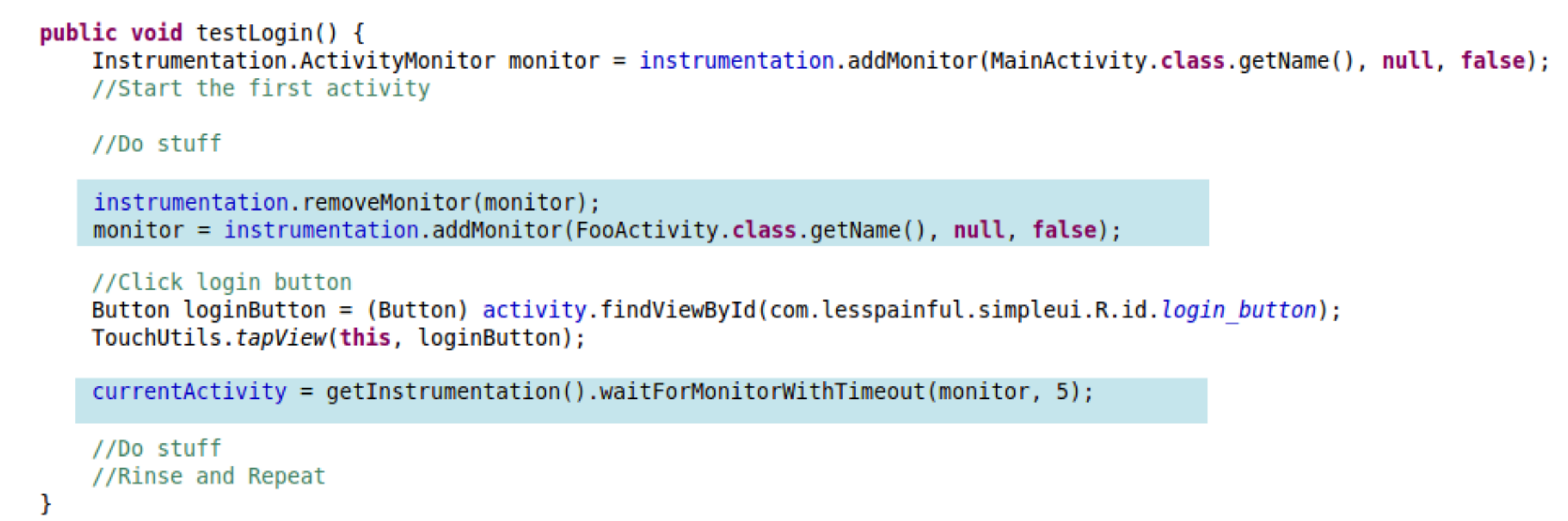
No probem right?.... well....

Testing across activities



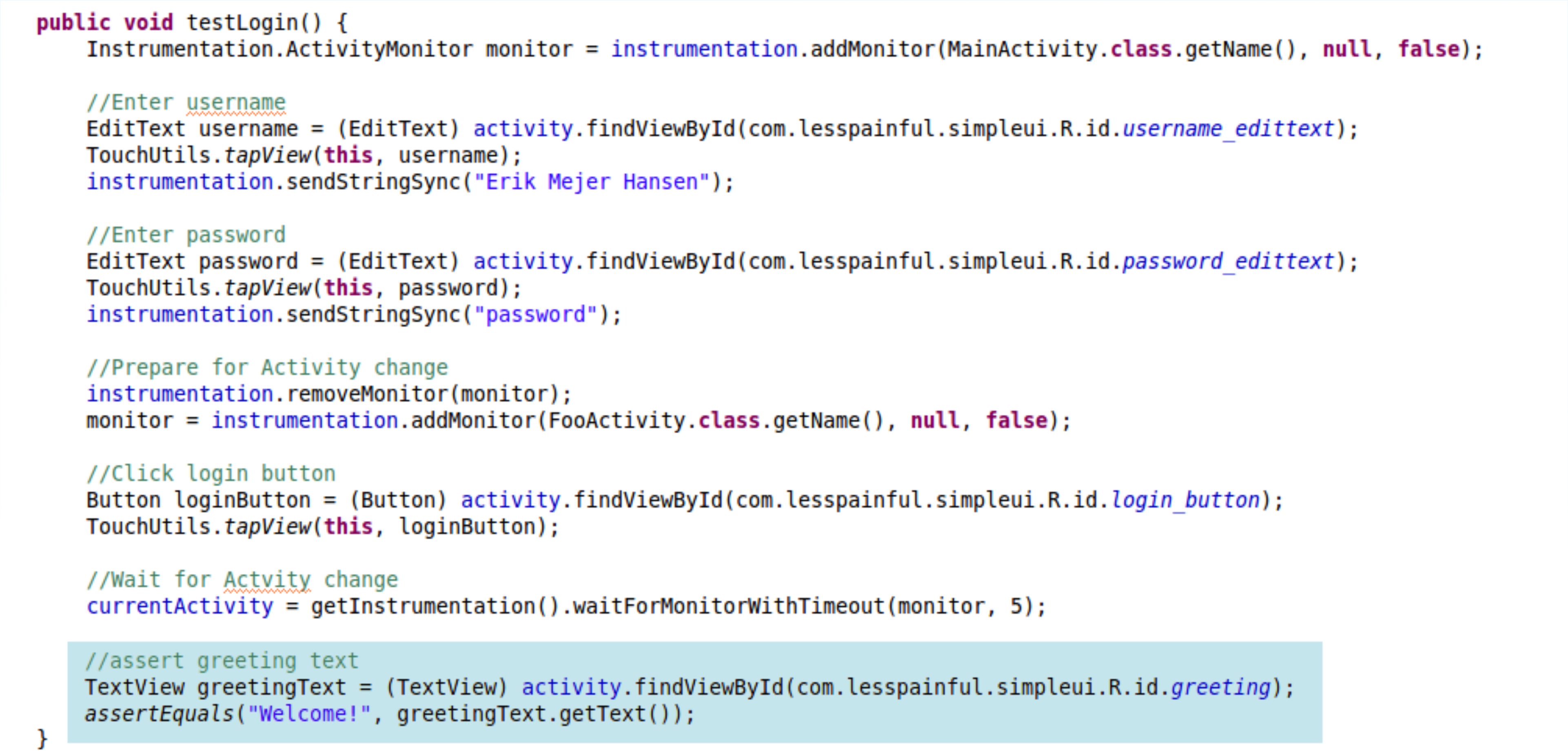
```
public void testLogin() {
    Instrumentation.ActivityMonitor monitor = instrumentation.addMonitor(MainActivity.class.getName(), null, false);
    //Start the first activity

    //Do stuff

    instrumentation.removeMonitor(monitor);
    monitor = instrumentation.addMonitor(FooActivity.class.getName(), null, false);

    //Click login button
    Button loginButton = (Button) activity.findViewById(com.lesspainful.simpleui.R.id.login_button);
    TouchUtils.tapView(this, loginButton);

    currentActivity = getInstrumentation().waitForMonitorWithTimeout(monitor, 5);

    //Do stuff
    //Rinse and Repeat
}
```

Testing across activities



```
public void testLogin() {
    Instrumentation.ActivityMonitor monitor = instrumentation.addMonitor(MainActivity.class.getName(), null, false);
    //Start the first activity

    //Do stuff

    instrumentation.removeMonitor(monitor);
    monitor = instrumentation.addMonitor(FooActivity.class.getName(), null, false);

    //Click login button
    Button loginButton = (Button) activity.findViewById(com.lesspainful.simpleui.R.id.login_button);
    TouchUtils.tapView(this, loginButton);

    currentActivity = getInstrumentation().waitForMonitorWithTimeout(monitor, 5);

    //Do stuff
    //Rinse and Repeat
}
```

Asserting the greeting



```
public void testLogin() {
    Instrumentation.ActivityMonitor monitor = instrumentation.addMonitor(MainActivity.class.getName(), null, false);

    //Enter username
    EditText username = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.username_edittext);
    TouchUtils.tapView(this, username);
    instrumentation.sendStringSync("Erik Mejer Hansen");

    //Enter password
    EditText password = (EditText) activity.findViewById(com.lesspainful.simpleui.R.id.password_edittext);
    TouchUtils.tapView(this, password);
    instrumentation.sendStringSync("password");

    //Prepare for Activity change
    instrumentation.removeMonitor(monitor);
    monitor = instrumentation.addMonitor(FooActivity.class.getName(), null, false);

    //Click login button
    Button loginButton = (Button) activity.findViewById(com.lesspainful.simpleui.R.id.login_button);
    TouchUtils.tapView(this, loginButton);

    //Wait for Actvity change
    currentActivity = getInstrumentation().waitForMonitorWithTimeout(monitor, 5);

    //assert greeting text
    TextView greetingText = (TextView) activity.findViewById(com.lesspainful.simpleui.R.id.greeting);
    assertEquals("Welcome!", greetingText.getText());
}
```

LESSPAINFUL
MOBILE APP TESTING

So...

So...

The test has to know when activity changes occur

So...

The test has to know when activity changes occur

Test become very sensitive to changes in activity flow

So...

The test has to know when activity changes occur

Test become very sensitive to changes in activity flow

Wouldn't it be nice if you could just?

```
clickOnButton("Login")
AssertTrue(searchText("Welcome!"))
```

So...

The test has to know when activity changes occur

Test become very sensitive to changes in activity flow

Wouldn't it be nice if you could just?

```
clickOnButton("Login")
AssertTrue(searchText("Welcome"))
```

And not have to worry about Activities?

Jayway and Robotium to the rescue

Created by Jayway (open source)

robotium

It's like Selenium, but for Android™

Jayway and Robotium to the rescue

Created by Jayway (open source)

Keeps track of the current Activity

robotium

It's like Selenium, but for Android™

Jayway and Robotium to the rescue

Created by Jayway (open source)

Keeps track of the current Activity

Provides a set of helper functionality

robotium

It's like Selenium, but for Android™

Jayway and Robotium to the rescue

Created by Jayway (open source)

Keeps track of the current Activity

Provides a set of helper functionality

- searchText
- searchButton

robotium

It's like Selenium, but for Android™

Jayway and Robotium to the rescue

Created by Jayway (open source)

Keeps track of the current Activity

Provides a set of helper functionality

- searchText
- searchButton
- clickOnButton
- pressMenuItem

robotium

It's like Selenium, but for Android™

Jayway and Robotium to the rescue

Created by Jayway (open source)

Keeps track of the current Activity

Provides a set of helper functionality
- searchText
- searchButton
- clickOnButton
- pressMenuItem
- waitForText

robotium

It's like Selenium, but for Android™

Benefits

Benefits

Quicker to write

Benefits

Quicker to write

More robust

robotium

It's like Selenium, but for Android™

Benefits

Quicker to write

More robust

Much more readable

robotium

It's like Selenium, but for Android™

Benefits

Quicker to write

More robust

Much more readable

Helpful community

robotium

It's like Selenium, but for Android™

Benefits

Quicker to write

More robust

Much more readable

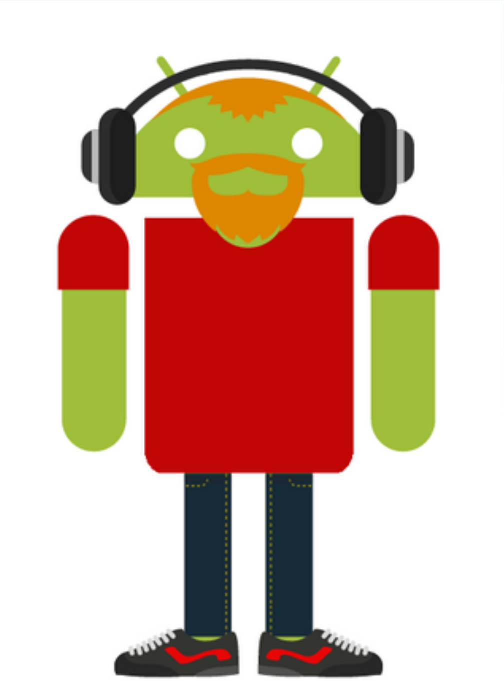
Helpful community

robotium.org

robotium

It's like Selenium, but for Android™

Finally... some code



robotium

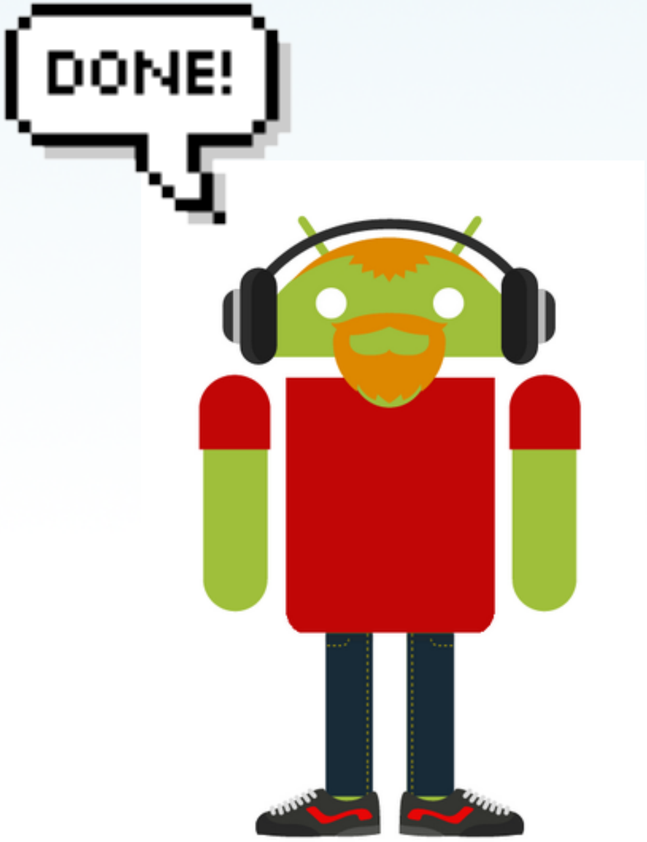
It's like Selenium, but for Android™

Full disclosure warning

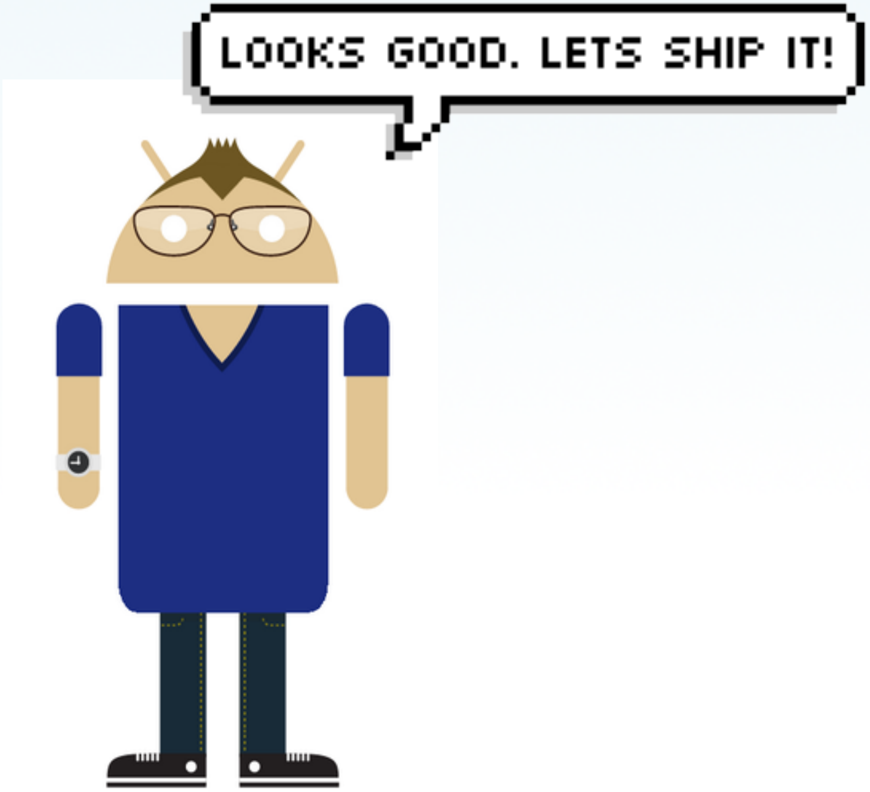
We are going to try and sell you something now.

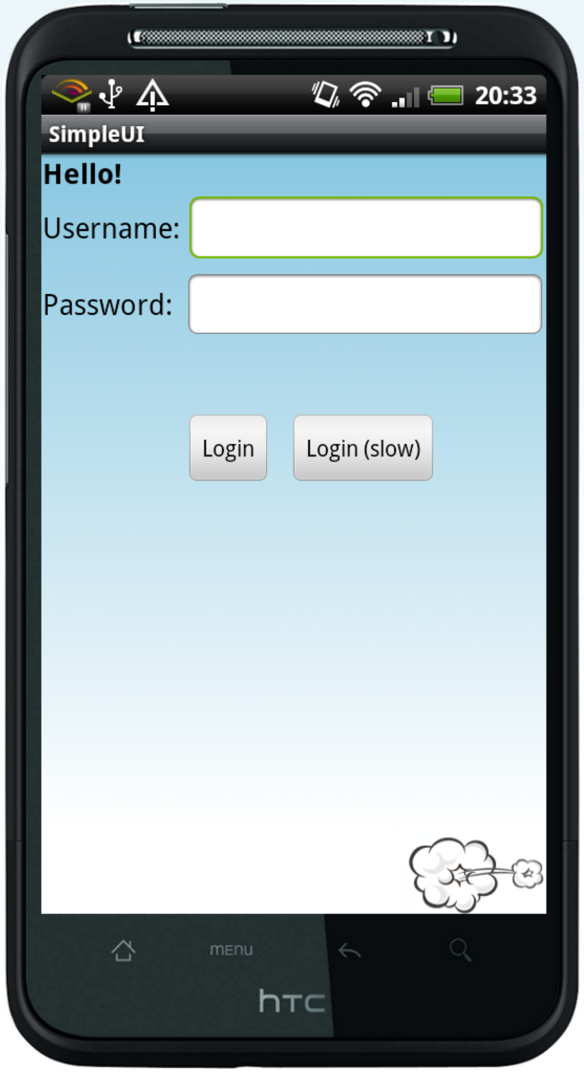
Ready for launch?

Scenario:	
Log in	✓
Log in slowly	✓
No username supplied	✓
No password supplied	✓
Wrong username	✓
Wrong password	✓



Ready for launch?





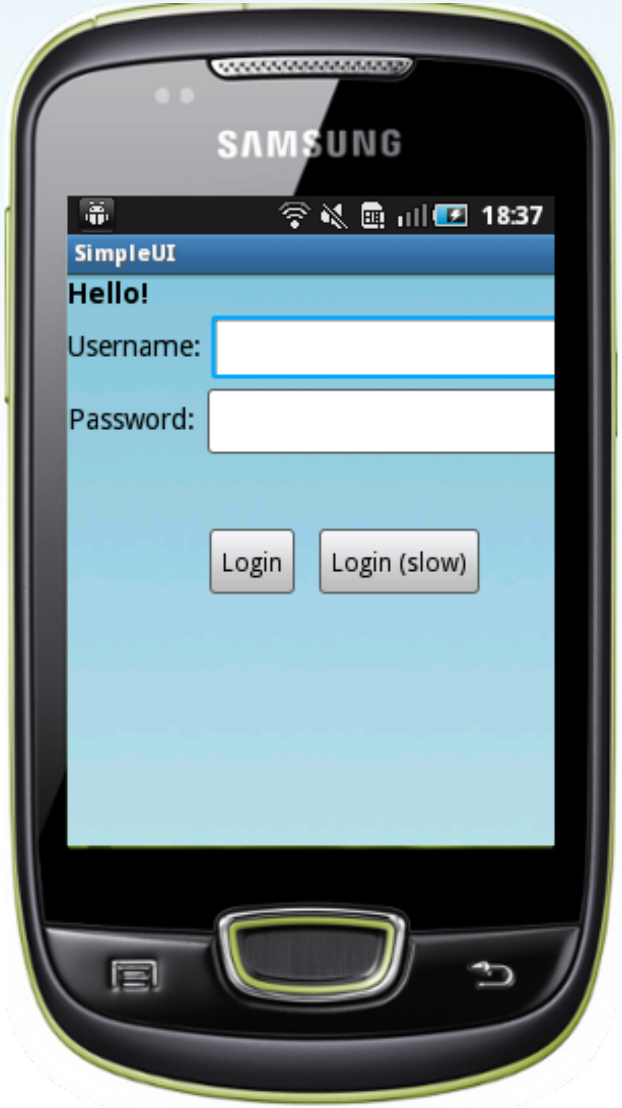
Hello!

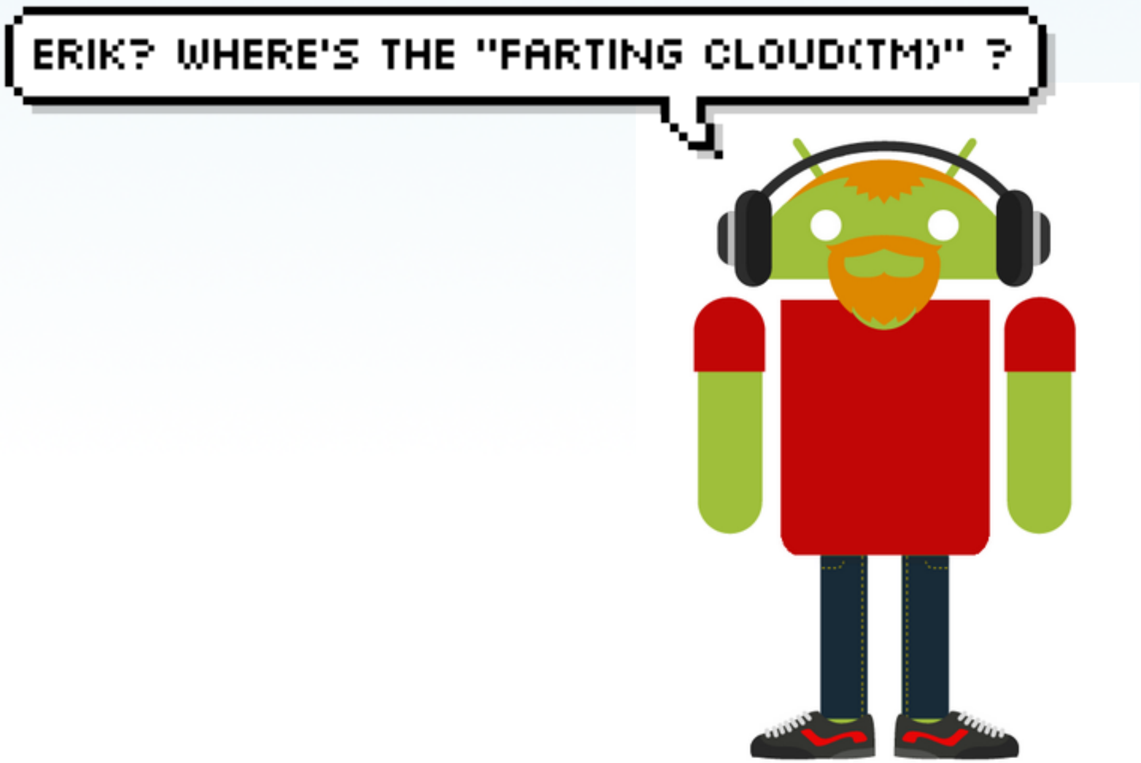
Username:

Password:

LESSPAINFUL
MOBILE APP TESTING

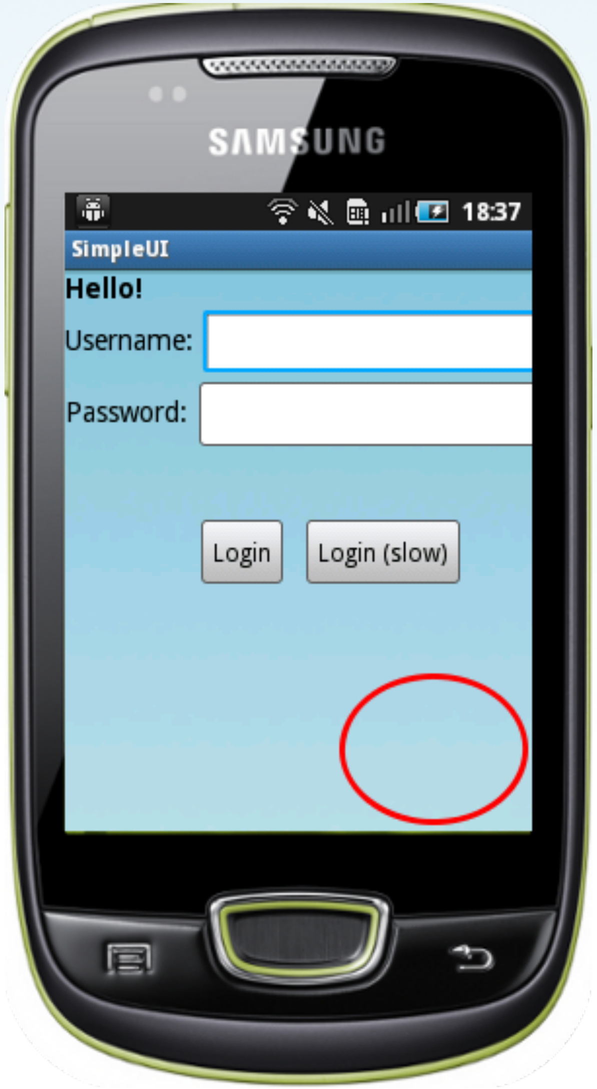
Ready for launch?

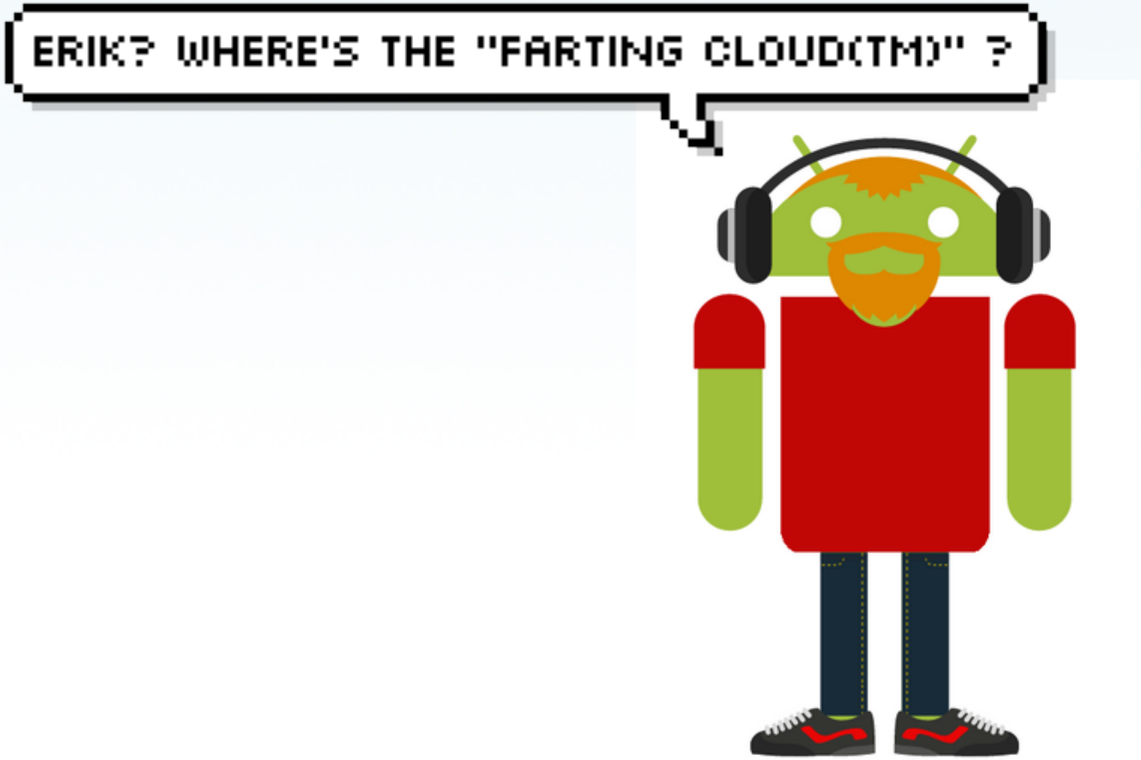




LESSPAINFUL
MOBILE APP TESTING

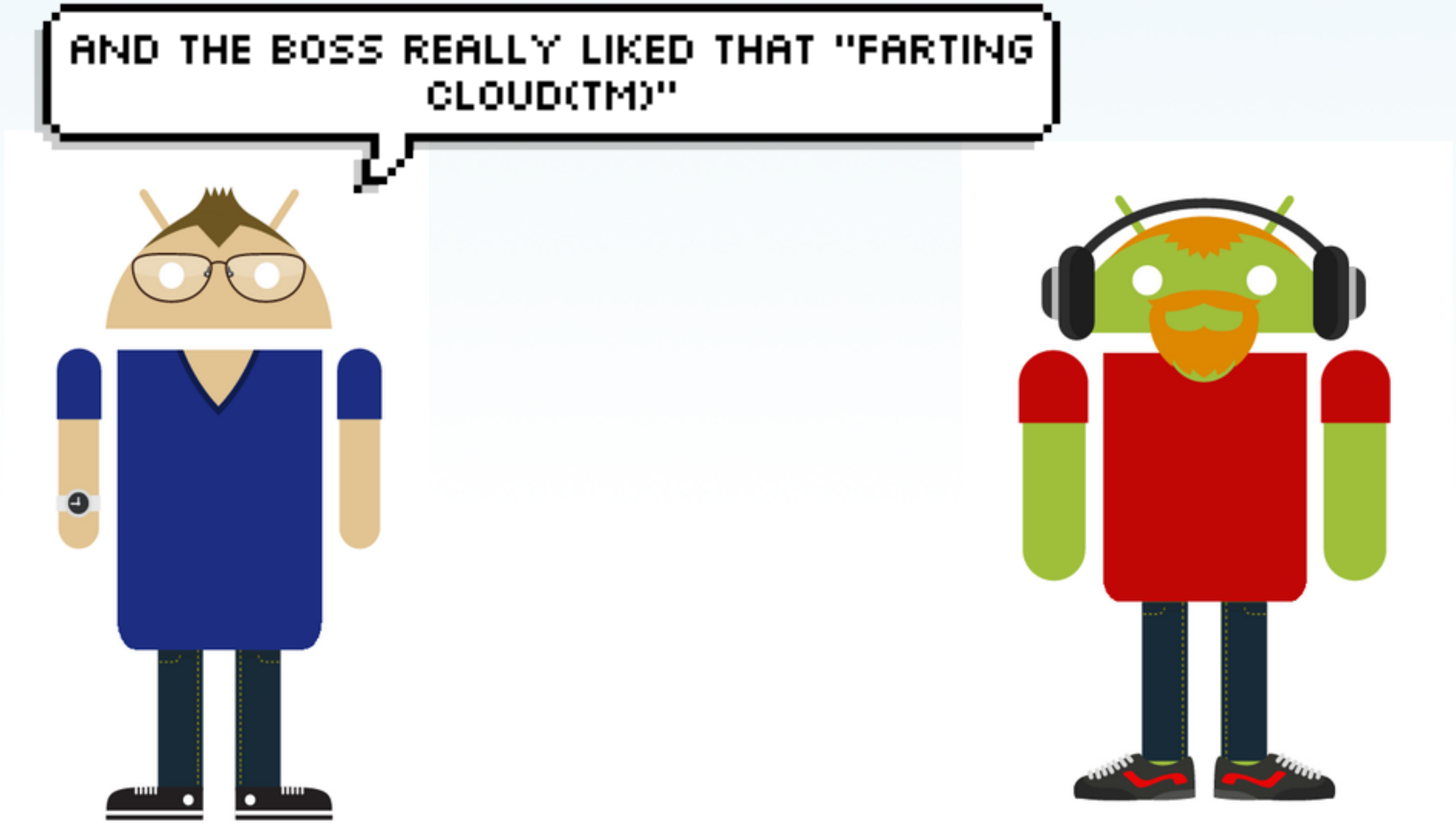
Ready for launch?





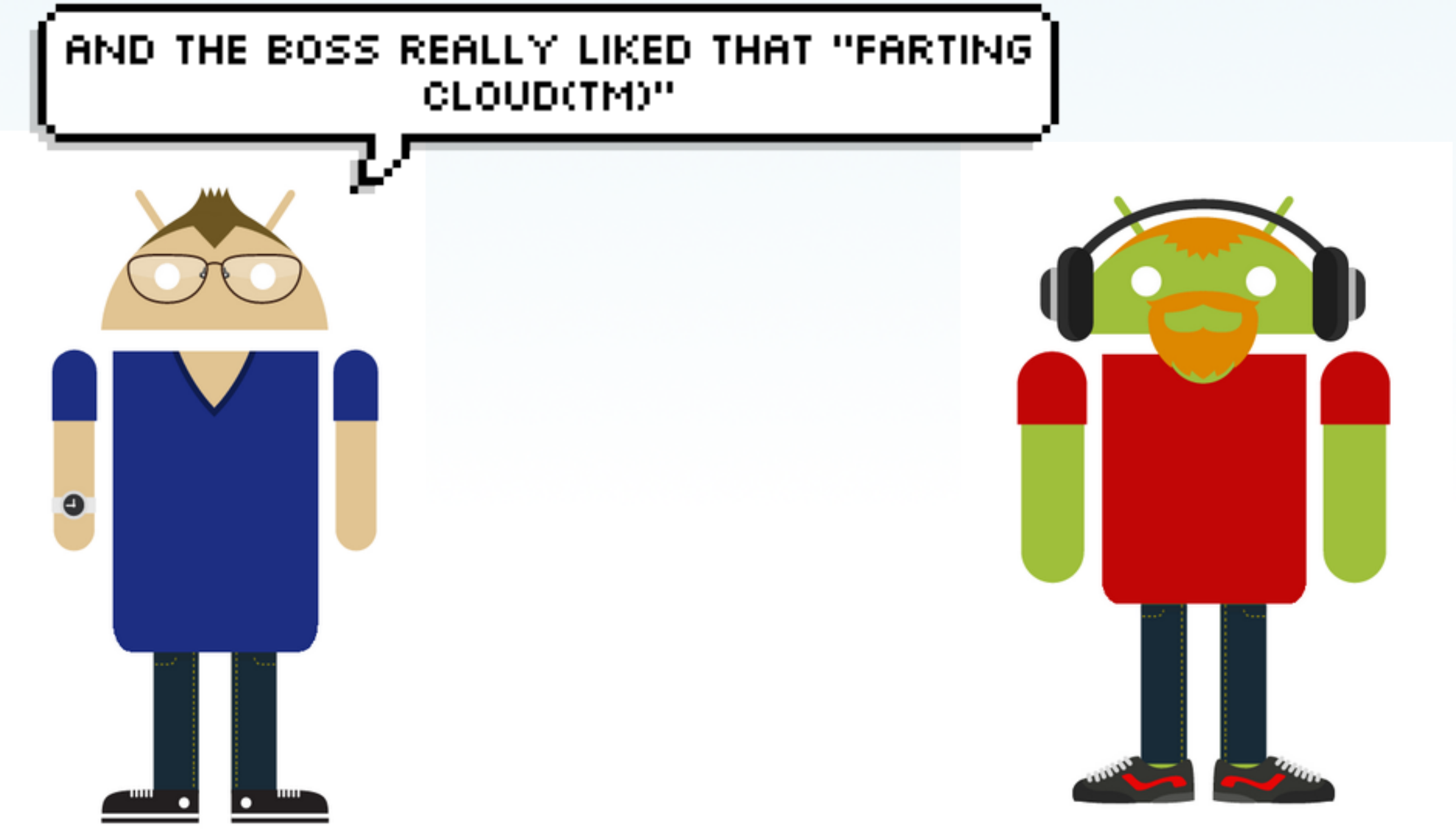
LESSPAINFUL
MOBILE APP TESTING

Ready for launch?



LESSPAINFUL
MOBILE APP TESTING

Ready for launch?

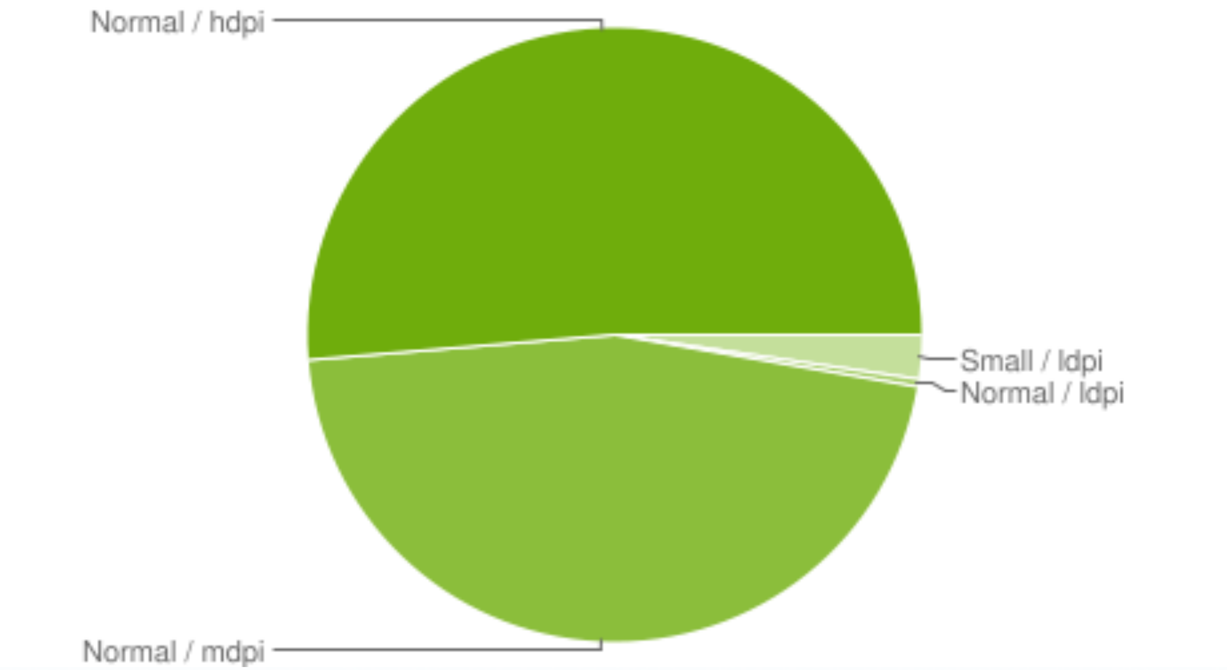


LESS

MOBILE AP

Ready for launch?

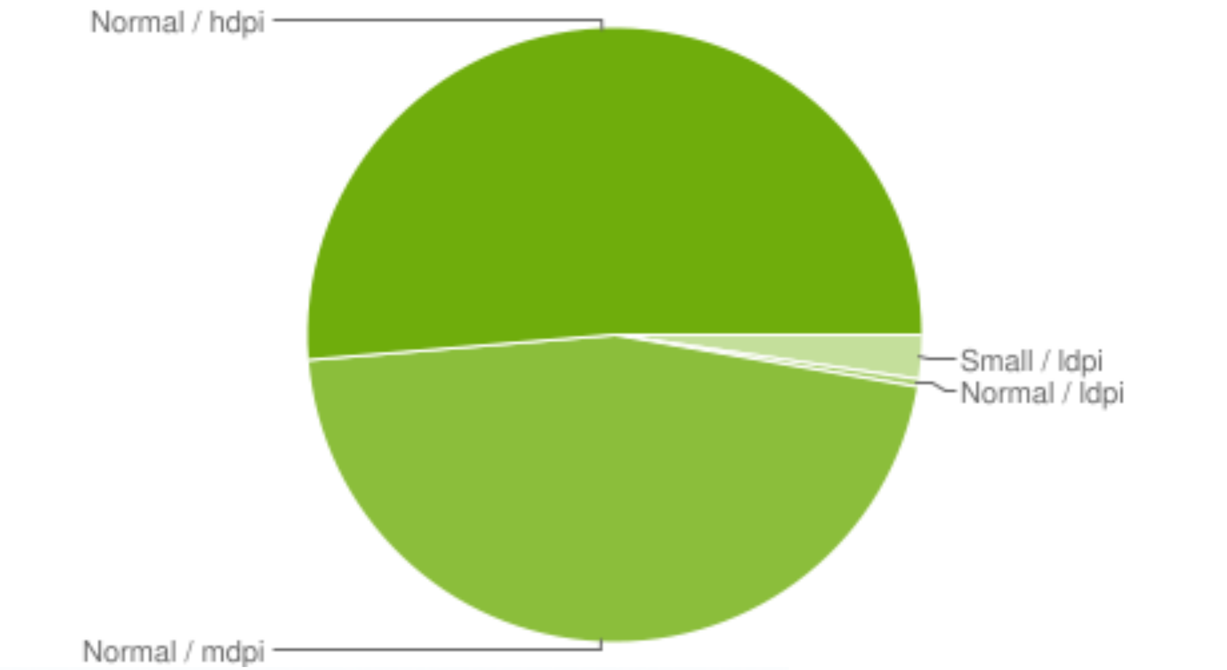
How bad is it?

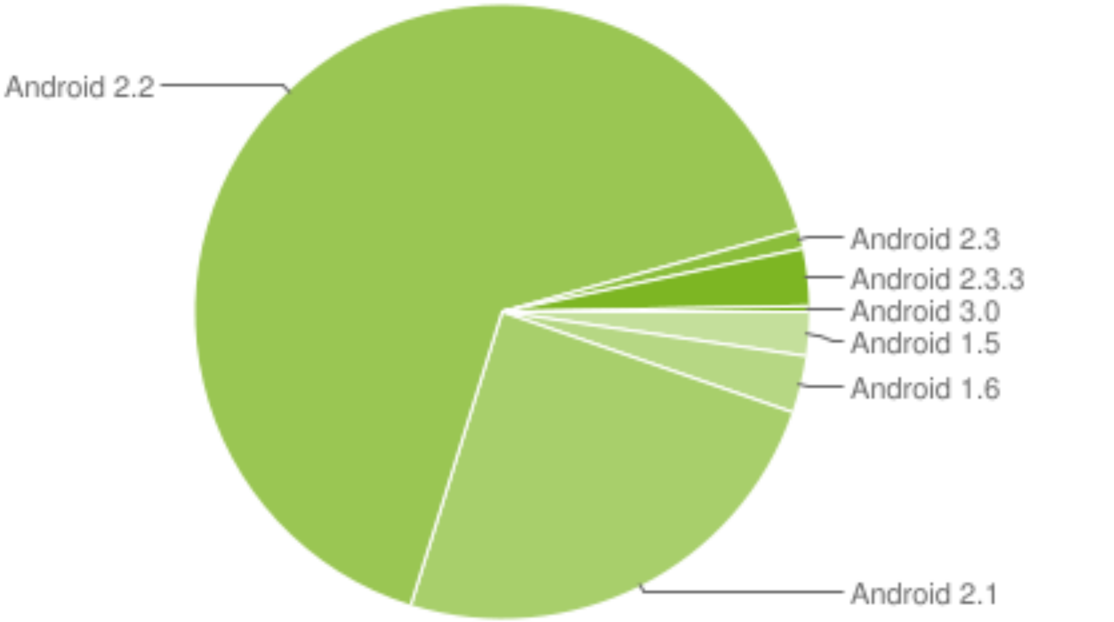


source: developer.google.com

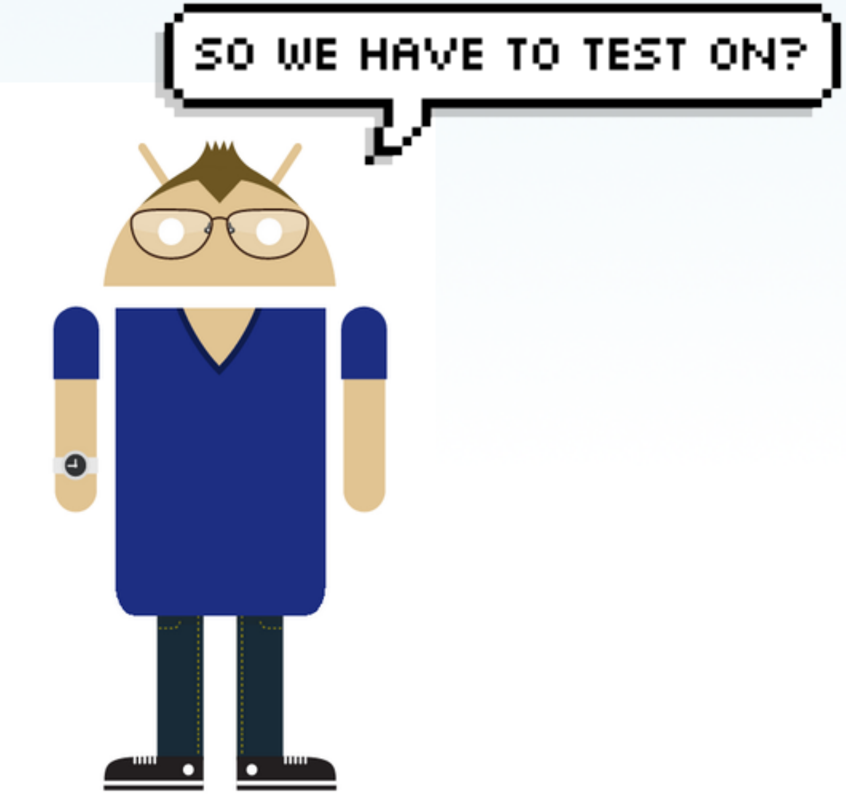
LESSPAINFUL
MOBILE APP TESTING

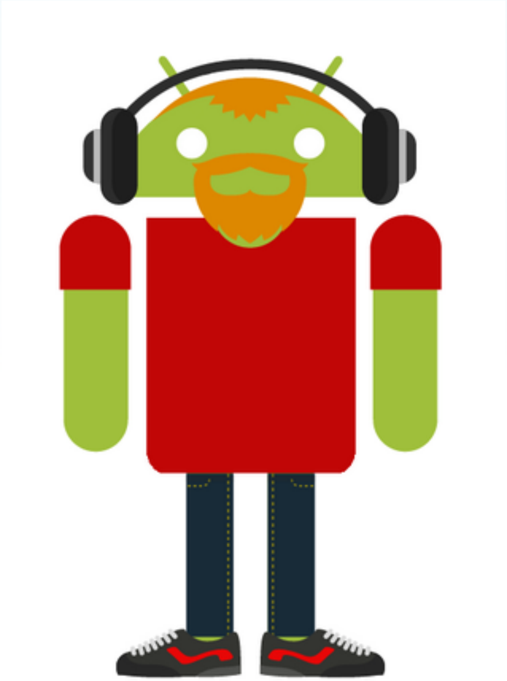
How bad is it?





source: developer.google.com


SO WE HAVE TO TEST ON?



28 DEVICES?!

ViewAsserts?

assertBaselineAligned

assertHasScreenCoordinates

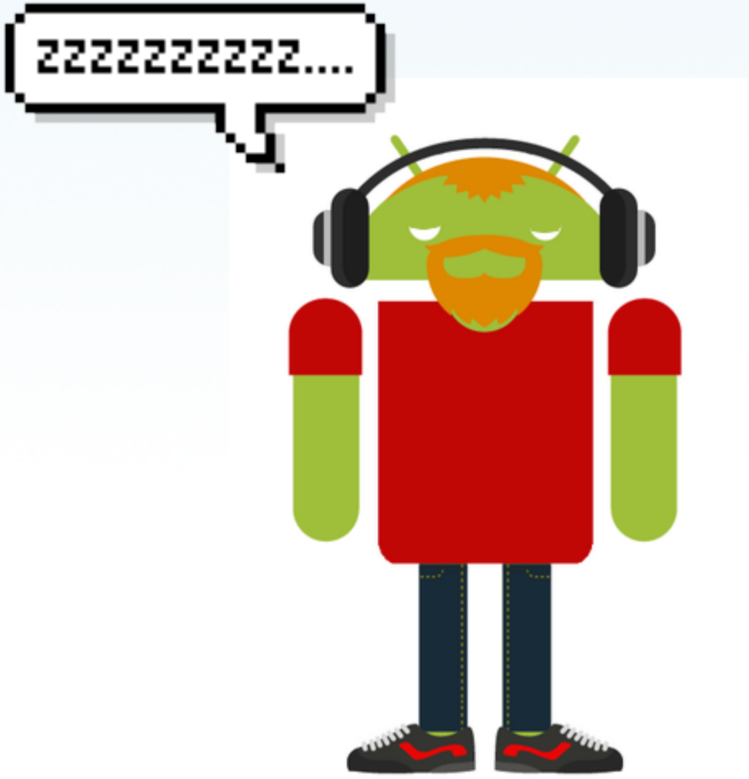
assertOffScreenAbove

assertRightAligned

Sounds like fun?

Scenario:	**X10**	**Desire HD**	**...**	**HTC Tattoo**
Log in				
Log in slowly				
No username supplied				
No password supplied				
Wrong username				
Wrong password				

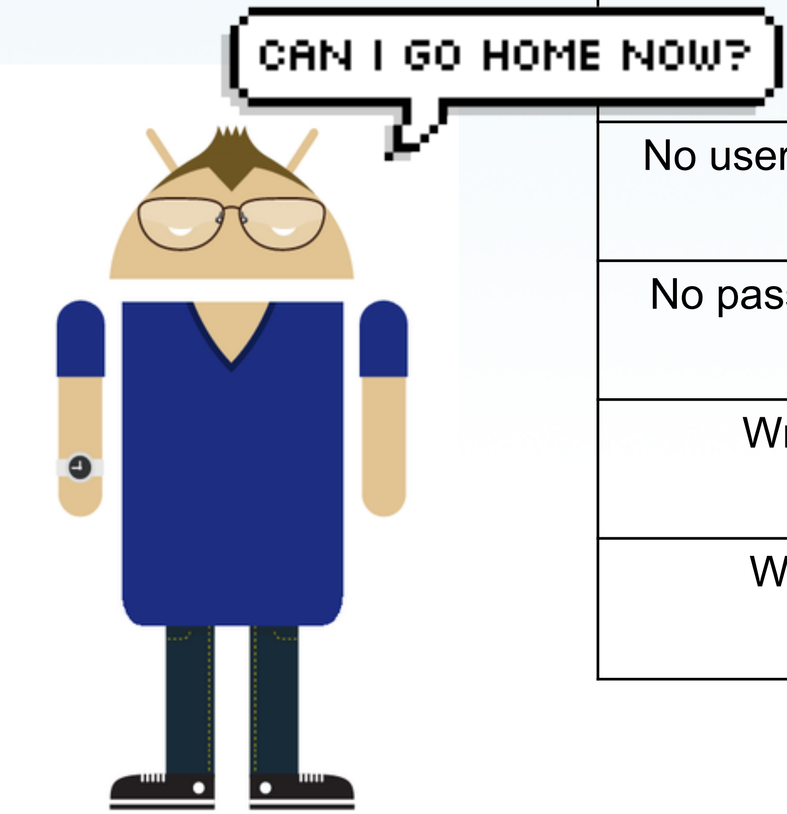
Sounds like fun?



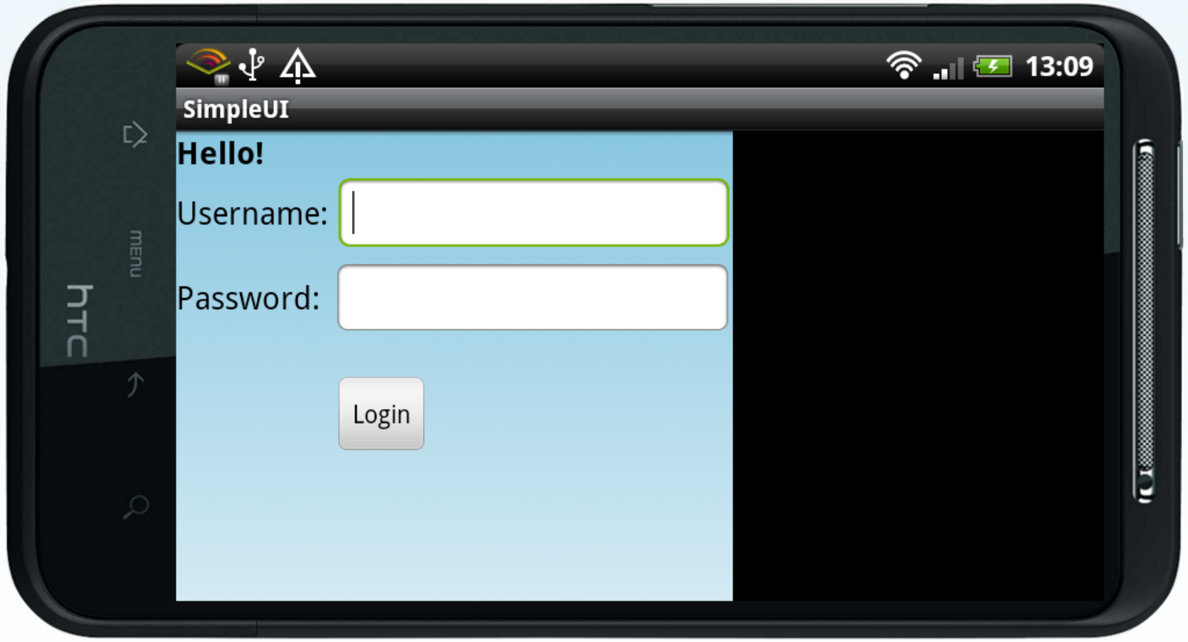
LESSPAINFUL
MOBILE APP TESTING

Sounds like fun?

Scenario:	X10	Desire HD	...	HTC Tattoo
Log in	✓	✓		✓
Log in slowly	✓	✓		✓
No username supplied	✓	✓		✓
No password supplied	✓	✓		✓
Wrong username	✓	✓		✓
Wrong password	✓	✓		✓



LESSPAINFUL
MOBILE APP TESTING

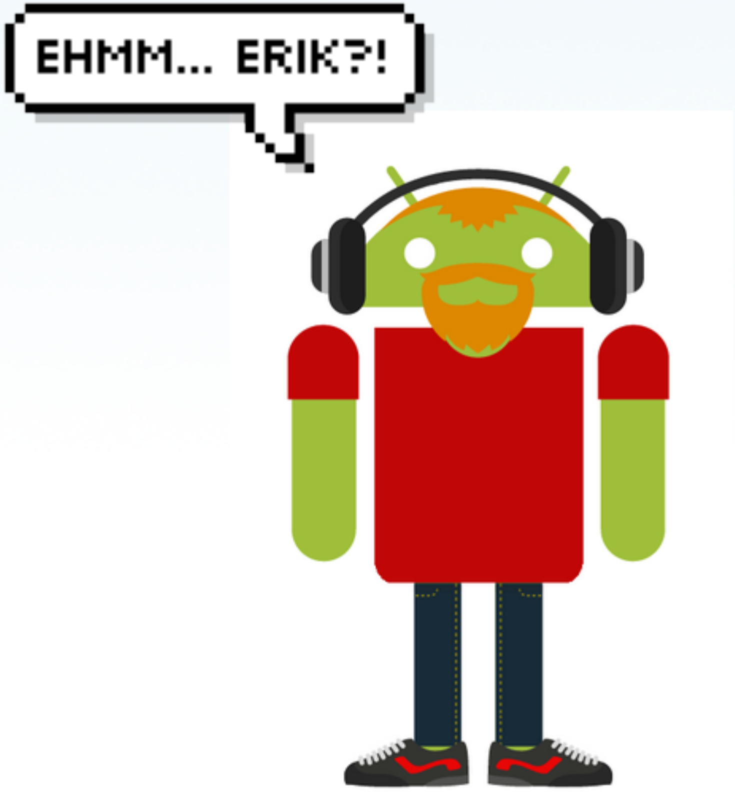


SimpleUI

Hello!

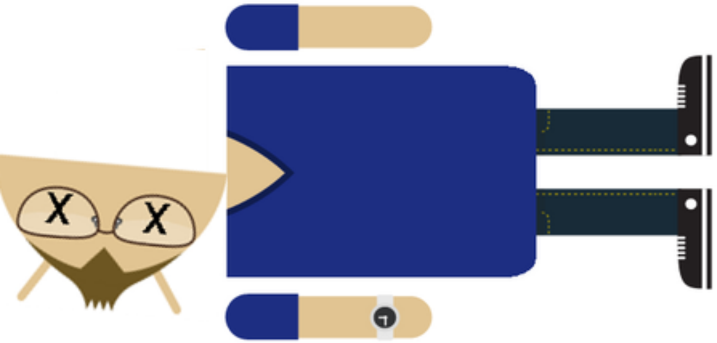
Username:

Password:



LESSPAINFUL
MOBILE APP TESTING

Scenario:	X10	Desire HD	...	HTC Tattoo
Log in (Landscape)				
Log in slowly (Landscape)				
No username supplied (Landscape)				
No password supplied (Landscape)				
Wrong username (Landscape)				
Wrong password (Landscape)				
Log in (Portrait)				
Log in slowly (Portrait)				
No username supplied (Portrait)				
No password supplied (Portrait)				
Wrong username (Portrait)				
Wrong password (Portrait)				



How about?

Feature: Login
Scenario: Login to the app

Given My "SimpleUIApp" app is running
And I wait for "Hello!" to appear

Then I enter "jonas" as "username"
And I enter "password" as "password"

Then I press the "Login" button
And I wait for "Welcome" to appear

How about?

```gherkin
Feature: Login
Scenario: Login to the app

Given My "SimpleUIApp" app is running
And I wait for "Hello!" to appear
Then take screenshot

Then I enter "jonas" as "username"
And I enter "password" as "password"
Then take screenshot

Then I press the "Login" button
And I wait for "Welcome" to appear
Then take screenshot
```

Then

Run it across multiple devices

Then

Run it across multiple devices

That you dont have to run out and buy

Then

Run it across multiple devices

That you dont have to run out and buy

And *see* the results

Then

Run it across multiple devices

- That you dont have to run out and buy

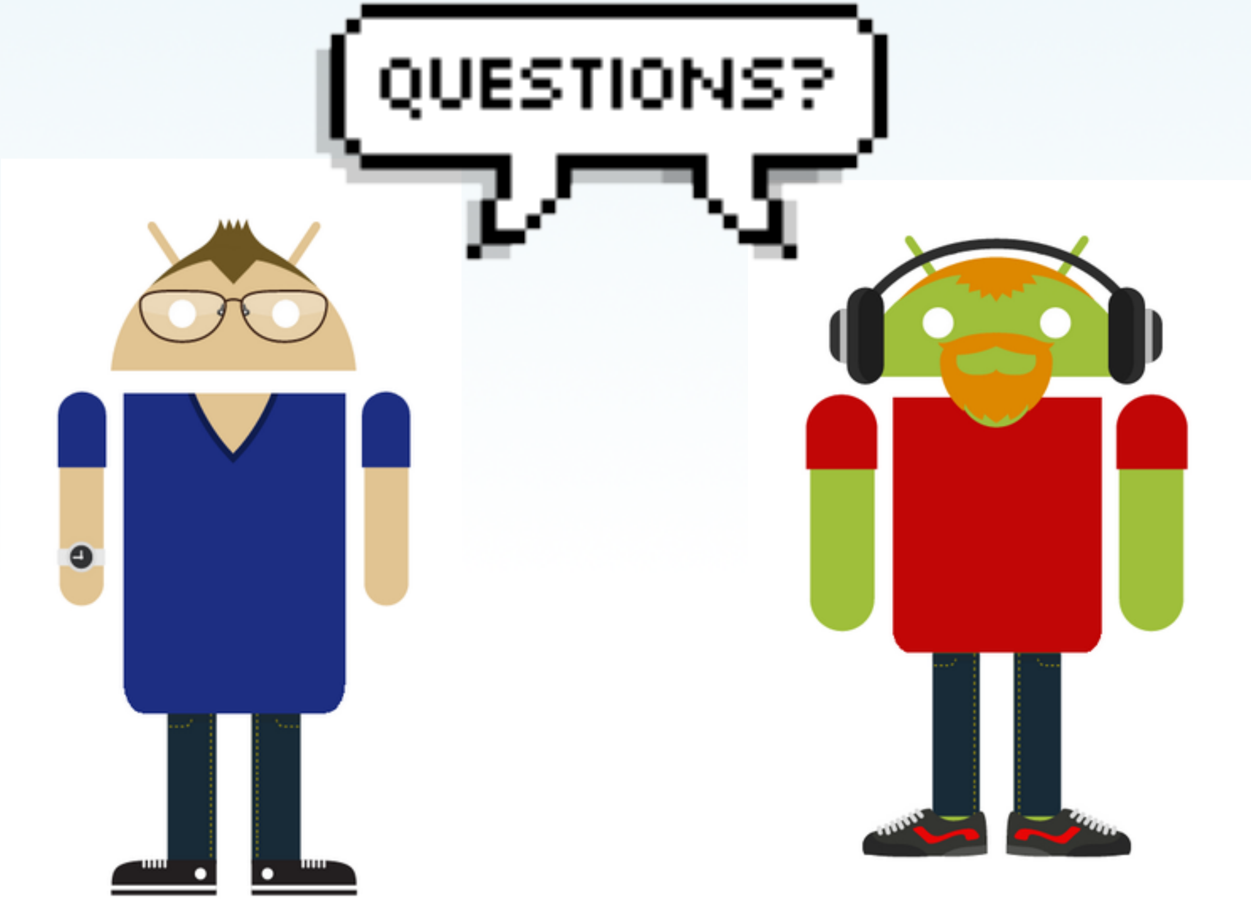
And *see* the results

- Easily comparable
- Shareable

PSSSST! SHOW THE REPORT NOW

LessPainful.com

contact@lesspainful.com



LESSPAINFUL
MOBILE APP TESTING

LessPainful.com

contact@lesspainful.com

LESSPAINFUL
MOBILE APP TESTING