HI! I'M ERIK

HI! I'M JONAS
I'll take some service bus with that J2EE please.
How about Android? That's still Java.
PERHAPS A GAME ABOUT FRUSTRATED AVIANS?
That will never sell. How about a nice tipping calculator?
EXCELLENT!
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 JUnit3
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 integration/system testing
Your options are...

ActivityInstrumentationTestCase2

For integration/system testing

No system mocks
Your options are...

ActivityInstrumentationTestCase2

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

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

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

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

```java
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));
    }
});
```
What happens now?

```java
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 problem right?
What happens now?

```java
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

```java
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

```java
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

```java
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 Activity change
    currentActivity = getInstrumentation().waitForMonitorWithTimeout(monitor, 5);

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

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

And not have to worry about Activities?
Jayway and Robotium to the rescue

Created by Jayway (open source)
Jayway and Robotium to the rescue

Created by Jayway (open source)

Keeps track of the current Activity
Jayway and Robotium to the rescue

Created by Jayway (open source)

Keeps track of the current Activity

Provides a set of helper functionality
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
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
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
Benefits
Benefits

Quicker to write
Benefits

Quicker to write

More robust
Benefits

Quicker to write

More robust

Much more readable
Benefits

Quicker to write
More robust
Much more readable
Helpful community
Benefits

Quicker to write
More robust
Much more readable
Helpful community

robotium.org
Finally... some code
Full disclosure warning

We are going to try and sell you something now.
Ready for launch?

<table>
<thead>
<tr>
<th>Scenario</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Log in</td>
<td>✓</td>
</tr>
<tr>
<td>Log in slowly</td>
<td>✓</td>
</tr>
<tr>
<td>No username supplied</td>
<td>✓</td>
</tr>
<tr>
<td>No password supplied</td>
<td>✓</td>
</tr>
<tr>
<td>Wrong username</td>
<td>✓</td>
</tr>
<tr>
<td>Wrong password</td>
<td>✓</td>
</tr>
</tbody>
</table>
Ready for launch?

LOOKS GOOD. LET'S SHIP IT!
Ready for launch?

ERIK? WHERE’S THE "FARTING CLOUD(TM)"?
Ready for launch?

ERIK? WHERE'S THE "FARTING CLOUD(TM)"?
Ready for launch?

AND THE BOSS REALLY LIKED THAT "FARTING CLOUD(TM)"
Ready for launch?

AND THE BOSS REALLY LIKED THAT "FARTING CLOUD(TM)"

LESS MOBILE APP
Ready for launch?

LETS JUST TEST ON A FEW DEVICES. HOW MANY CAN THERE BE?
How bad is it?

source: developer.google.com
How bad is it?

source: developer.google.com
SO WE HAVE TO TEST ON?
ViewAsserts?

assertBaselineAligned
assertHasScreenCoordinates
assertOffScreenAbove
assertRightAligned
Sounds like fun?

<table>
<thead>
<tr>
<th>Scenario:</th>
<th>X10</th>
<th>Desire HD</th>
<th>...</th>
<th>HTC Tattoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log in slowly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No username supplied</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No password supplied</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong username</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong password</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sounds like fun?
Sounds like fun?

<table>
<thead>
<tr>
<th>Scenario:</th>
<th>X10</th>
<th>Desire HD</th>
<th>...</th>
<th>HTC Tattoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log in</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Log in slowly</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>No username supplied</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>No password supplied</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Wrong username</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Wrong password</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Scenario:</td>
<td>X10</td>
<td>Desire HD</td>
<td>...</td>
<td>HTC Tattoo</td>
</tr>
<tr>
<td>-----------</td>
<td>-----</td>
<td>-----------</td>
<td>-----</td>
<td>------------</td>
</tr>
<tr>
<td>Log in (Landscape)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log in slowly (Landscape)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No username supplied (Landscape)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No password supplied (Landscape)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong username (Landscape)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong password (Landscape)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log in (Portrait)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log in slowly (Portrait)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No username supplied (Portrait)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No password supplied (Portrait)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong username (Portrait)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong password (Portrait)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
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 don't have to run out and buy
Then

Run it across multiple devices

That you don't have to run out and buy

And see the results
Then

Run it across multiple devices

That you don't have to run out and buy

And see the results

Easily comparable

Shareable
PSSSST! SHOW THE REPORT NOW
LessPainful.com
contact@lesspainful.com
QUESTIONS?