



+ing



W

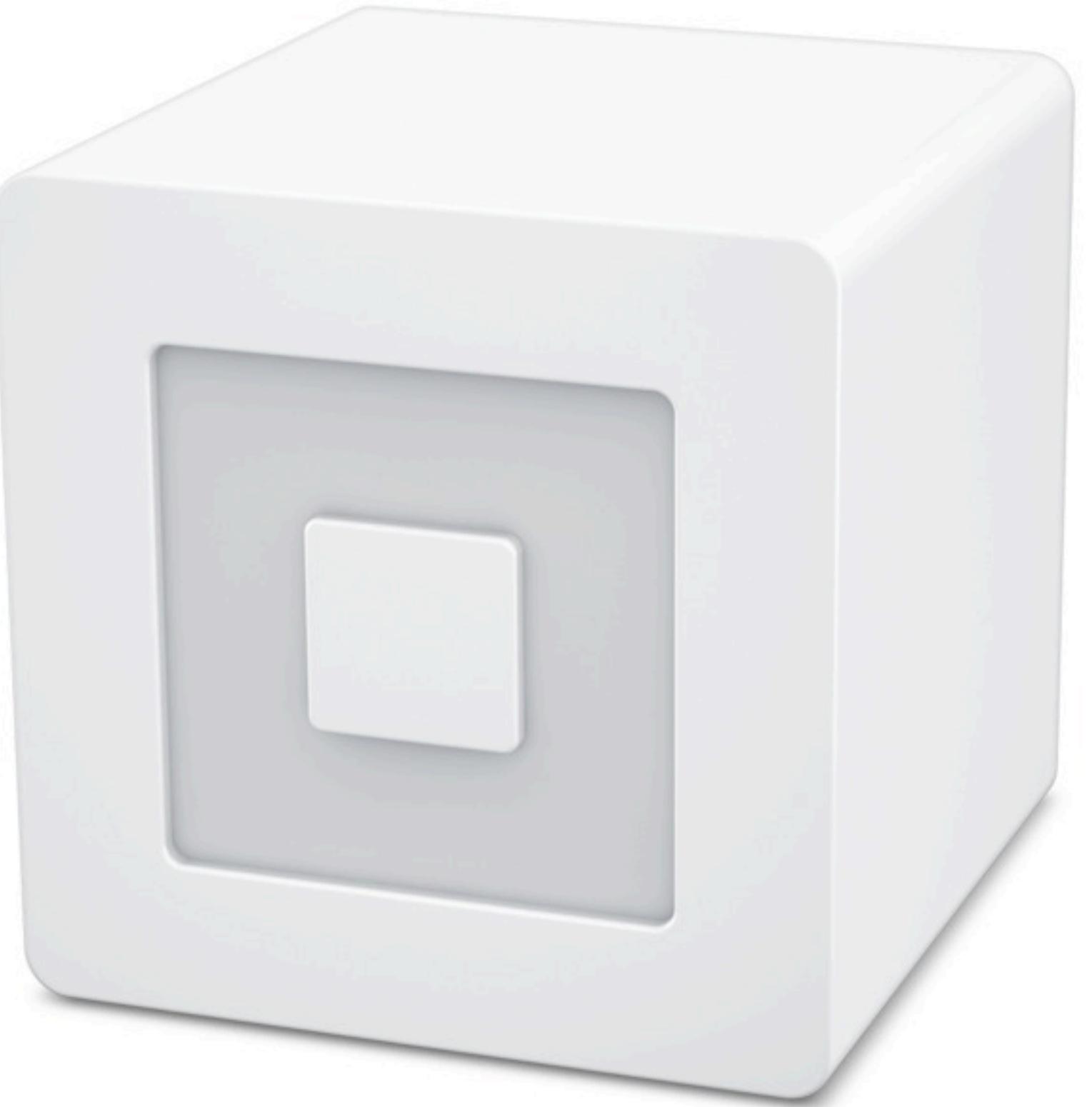




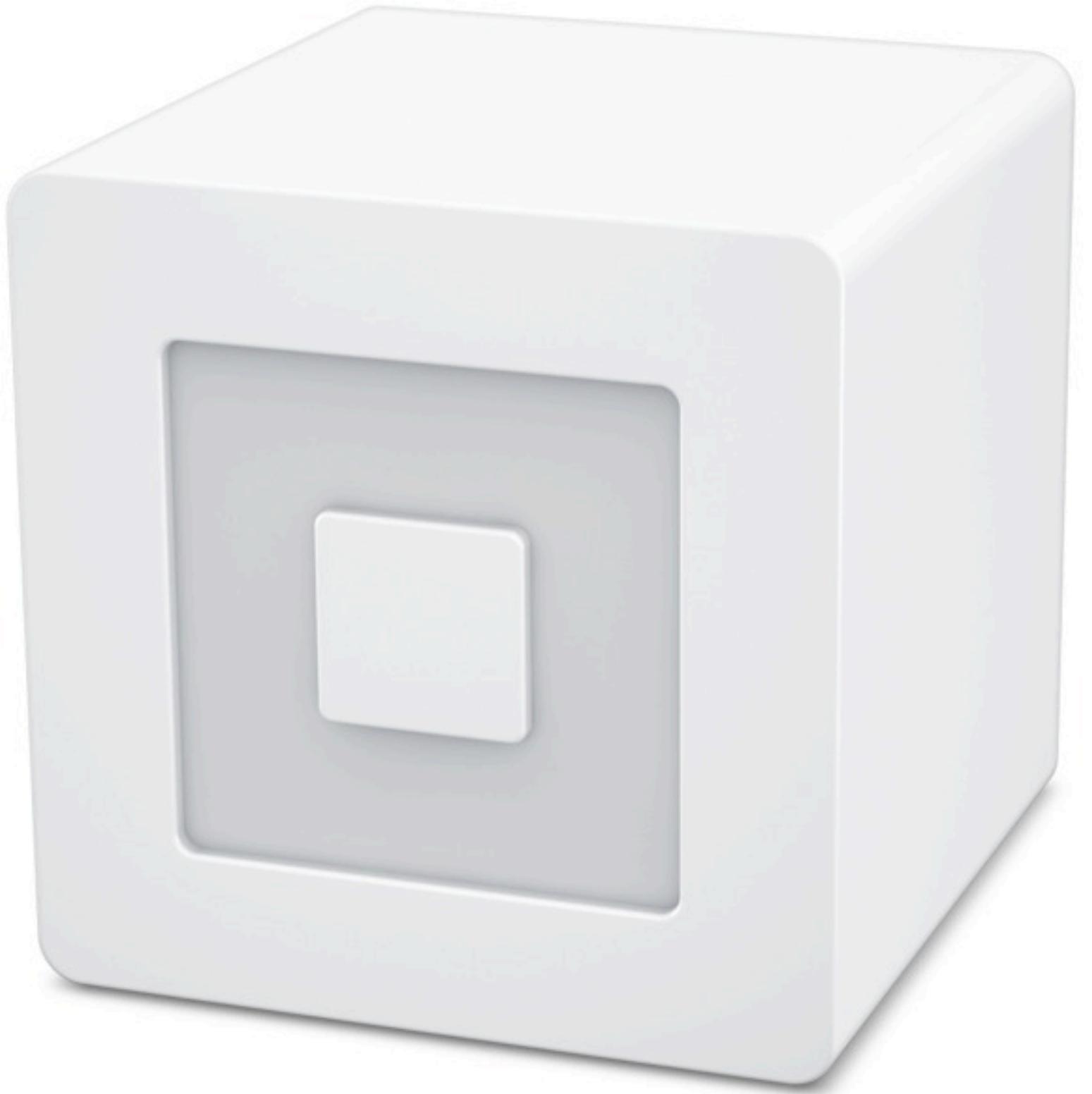
Bootstrapping



Jake Wharton



(not our logo)



Overview

Overview

- Open source speed dating

Overview

- Open source speed dating
- Use case: paying with Square Wallet

Overview

- Open source speed dating
- Use case: paying with Square Wallet
- Why you should open source

Overview

- Open source speed dating
- Use case: paying with Square Wallet
- Why you should open source
- Shameless plug for Square

ActionBarSherlock

ActionBarSherlock

- Backports action bar to Android 2.1+

ActionBarSherlock

- Backports action bar to Android 2.1+
- Mirrors API 14 API

ActionBarSherlock

- Backports action bar to Android 2.1+
- Mirrors API 14 API
- Requires duplicated theme attributes but otherwise a mirror

ActionBarSherlock

- Backports action bar to Android 2.1+
- Mirrors API 14 API
- Requires duplicated theme attributes but otherwise a mirror
- Extension of support library

ActionBarSherlock

- Backports action bar to Android 2.1+
- Mirrors API 14 API
- Requires duplicated theme attributes but otherwise a mirror
- Extension of support library
- SearchView, ShareActionProvider

ActionBarSherlock

ActionBarSherlock

```
// Native  
getActionBar().setHomeAsUpEnabled(true);
```

ActionBarSherlock

```
// Native  
getActionBar().setHomeAsUpEnabled(true);
```

```
// ActionBarSherlock  
getSupportActionBar().setHomeAsUpEnabled(true);
```

ActionBarSherlock

ActionBarSherlock

```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.bikeshed, menu);  
    return super.onCreateOptionsMenu(menu);  
}  
  
public boolean onOptionsItemSelected(MenuItem item) {  
    if (item.getItemId() == R.id.bike) {  
        // Do things  
        return true;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

ActionBarSherlock

ActionBarSherlock

```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.bikeshed, menu);  
    return super.onCreateOptionsMenu(menu);  
}  
  
public boolean onOptionsItemSelected(MenuItem item) {  
    if (item.getItemId() == R.id.bike) {  
        // Do things  
        return true;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

ActionBarSherlock

ActionBarSherlock

```
// Native  
import android.app.ActionBar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.Window;
```

ActionBarSherlock

```
// Native
import android.app.ActionBar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.Window;

// ActionBarSherlock
import com.actionbarsherlock.app.ActionBar;
import com.actionbarsherlock.view.Menu;
import com.actionbarsherlock.view.MenuItem;
import com.actionbarsherlock.view.Window;
```

ActionBarSherlock

ActionBarSherlock

```
<style name="Theme.Soundboard" extends="android:Theme.Holo">
<item name="android: actionBarStyle">
    @style/Widget.Soundboard.ActionBar
</item>
</style>
```

ActionBarSherlock

```
<style name="Theme.Soundboard" extends="android:Theme.Holo">
<item name="android: actionBarStyle">
    @style/Widget.Soundboard.ActionBar
</item>
</style>

<style name="Widget.Soundboard.ActionBar"
       extends="android:Widget.Holo.ActionBar">
<item name="android: displayOptions">showLogo | homeAsUp</item>
<item name="android: logo">@drawable/logo</item>
</style>
```

ActionBarSherlock

ActionBarSherlock

```
<style name="Theme.Soundboard" extends="Theme.Sherlock">
<item name=" actionBarStyle">
    @style/Widget.Soundboard.ActionBar
</item>
<item name="android: actionBarStyle">
    @style/Widget.Soundboard.ActionBar
</item>
</style>
```

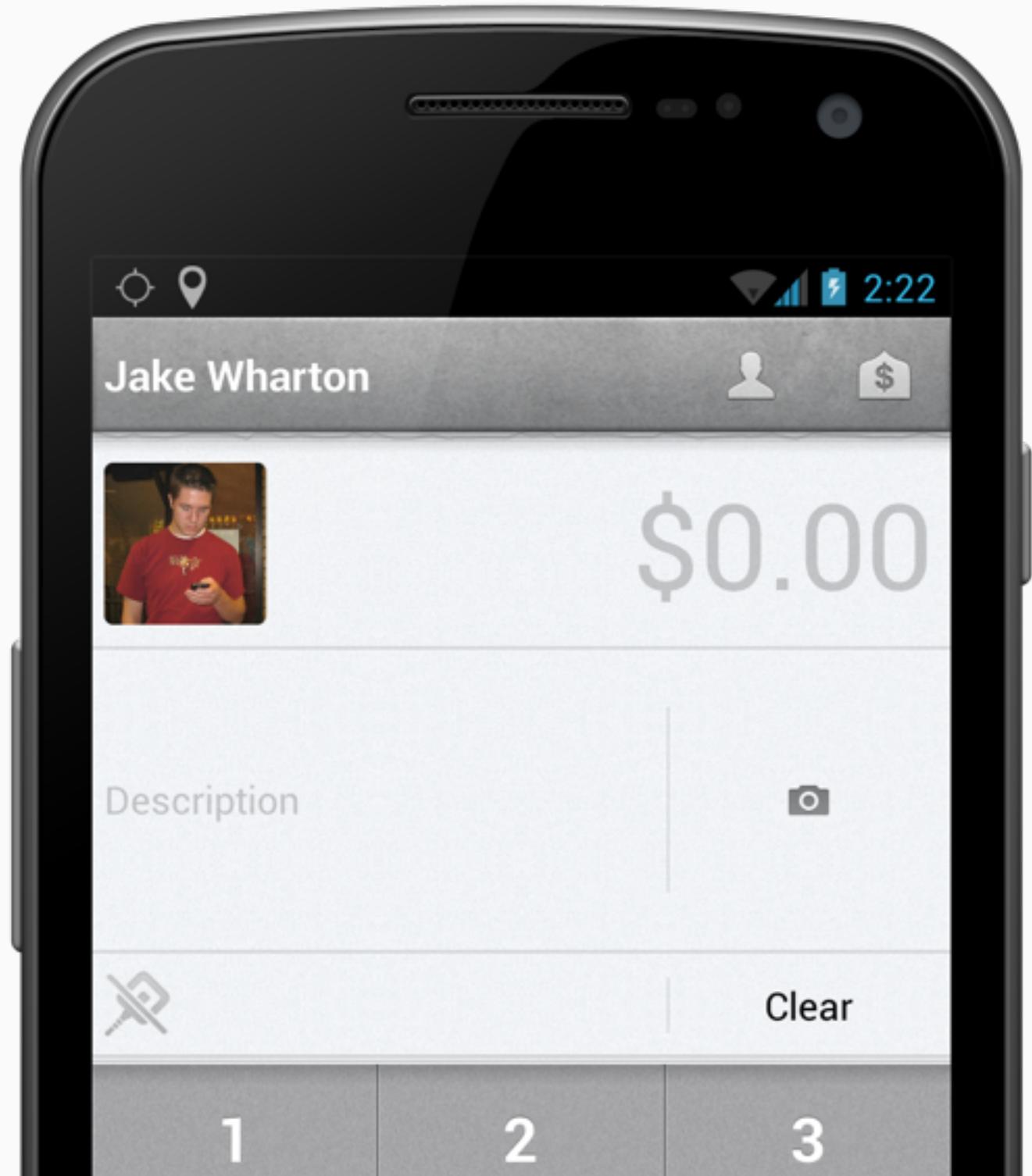
ActionBarSherlock

```
<style name="Theme.Soundboard" extends="Theme.Sherlock">
<item name=" actionBarStyle">
    @style/Widget.Soundboard.ActionBar
</item>
<item name="android: actionBarStyle">
    @style/Widget.Soundboard.ActionBar
</item>
</style>

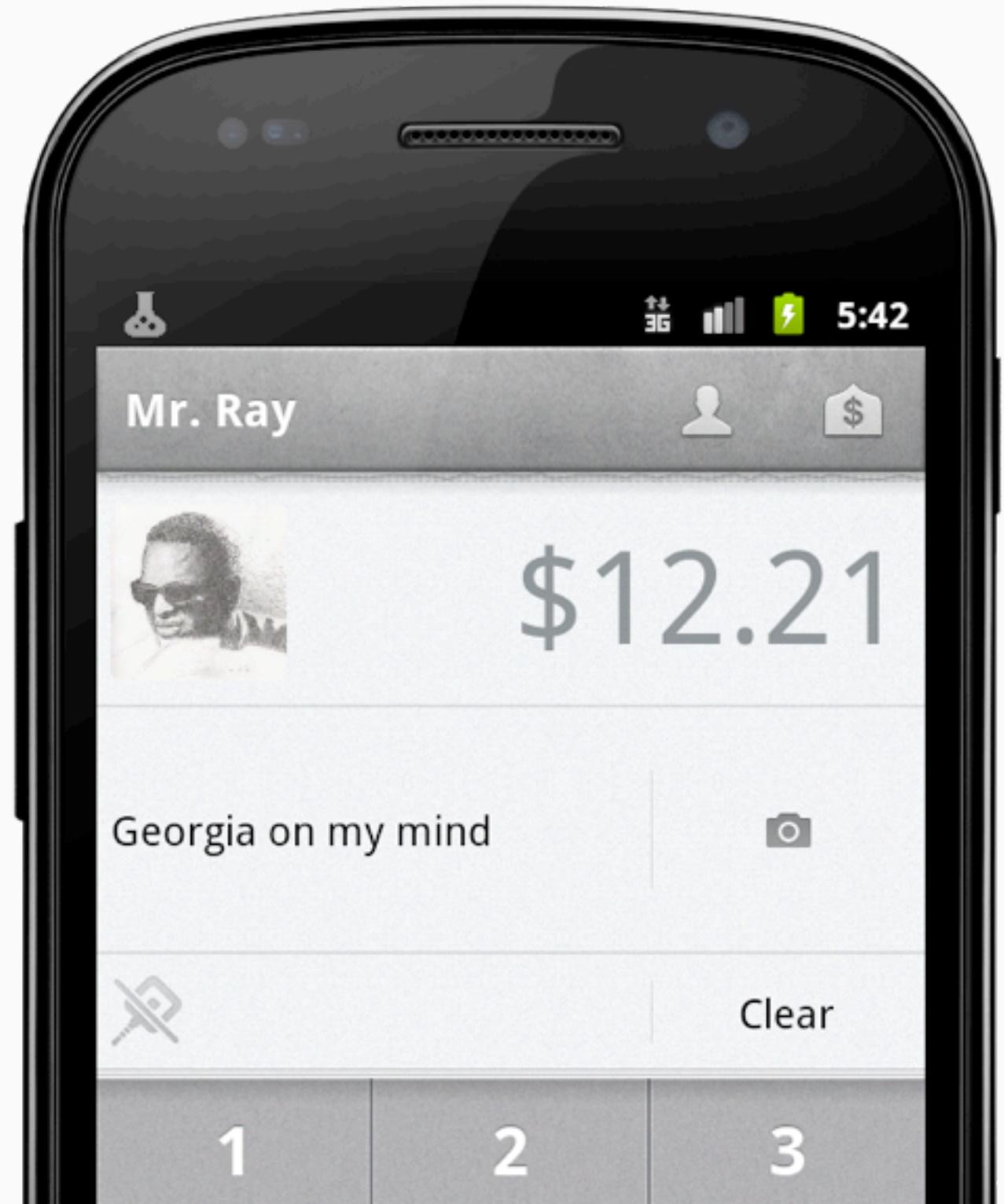
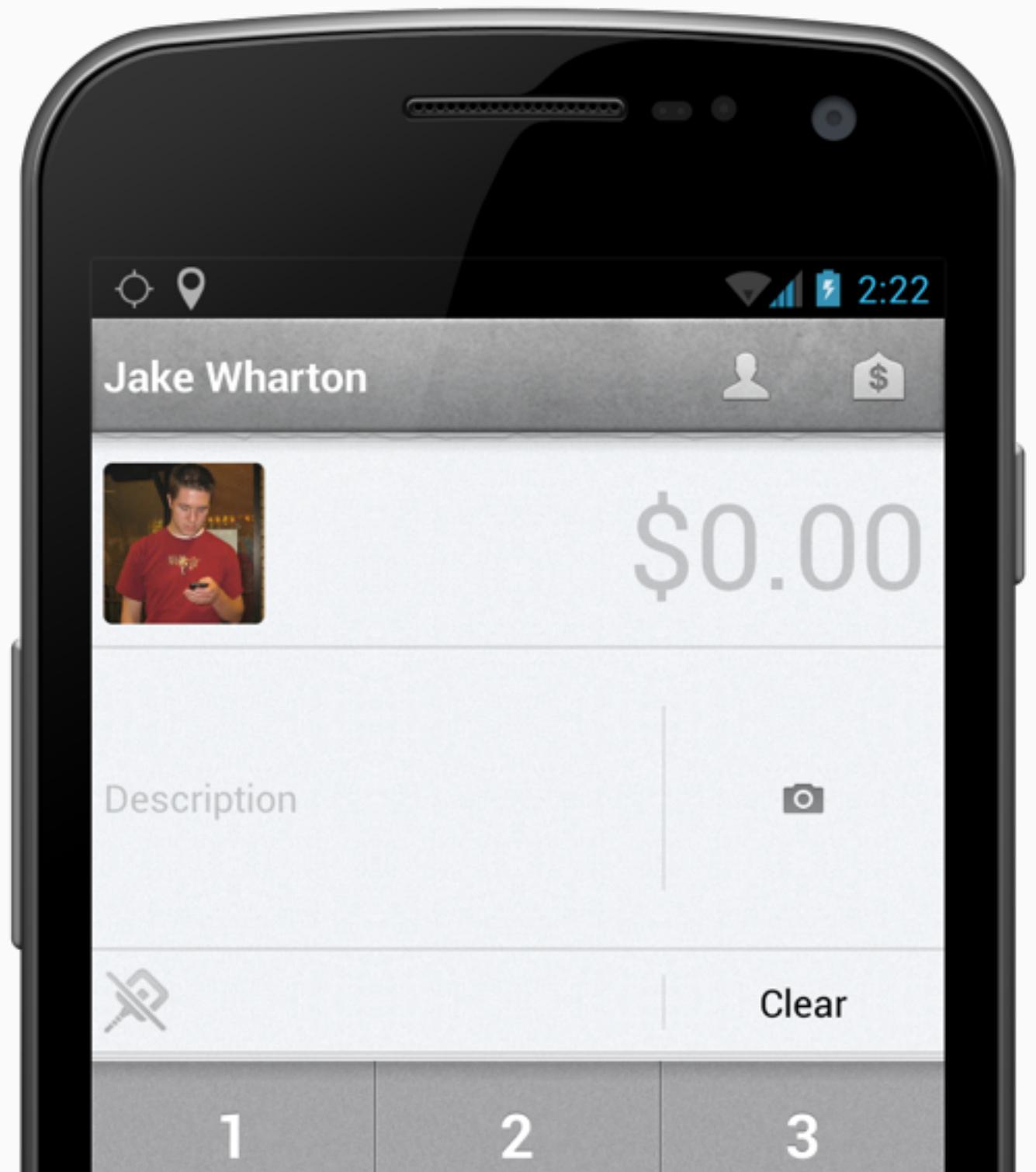
<style name="Widget.Soundboard.ActionBar"
       extends="Widget.Sherlock.ActionBar">
<item name=" displayOptions">showLogo | homeAsUp</item>
<item name=" android: displayOptions">showLogo | homeAsUp</item>
<item name=" logo">@drawable/logo</item>
<item name=" android: logo">@drawable/logo</item>
</style>
```

ActionBarSherlock

ActionBarSherlock



ActionBarSherlock



ActionBarSherlock

ActionBarSherlock

abs.io/

ViewPagerIndicator

ViewPagerIndicator

- Usually the only indication of swiping

ViewPagerIndicator

- Usually the only indication of swiping
- Provides current-page context

ViewPagerIndicator

- Usually the only indication of swiping
- Provides current-page context
- It really Just Looks Nice™

ViewPagerIndicator

ViewPagerIndicator

```
ViewPager pager = findViewById(R.id.pager);  
pager.setAdapter(new MyPagerAdapter(..));
```

ViewPagerIndicator

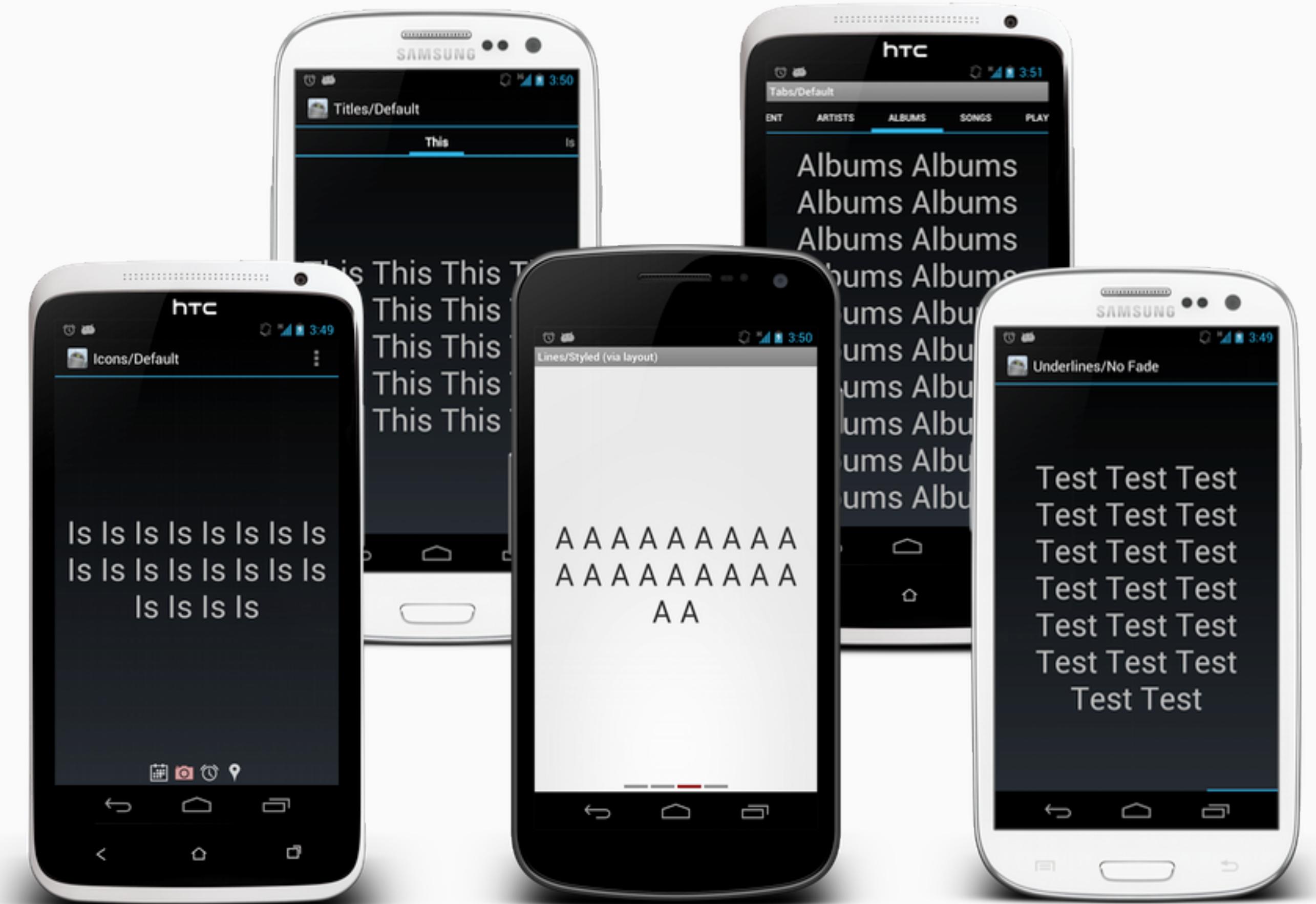
```
ViewPager pager = findViewById(R.id.pager);  
pager.setAdapter(new MyPagerAdapter(..));
```

```
PageIndicator indicator = findViewById(R.id.titles);  
indicator.setPager(pager);
```

ViewPagerIndicator

ViewPagerIndicator

```
<LinearLayout ... >  
  
    <com.viewpagerindicator.TitlePageIndicator  
        ...  
    />  
    <android.support.v4.view.ViewPager  
        ...  
    />  
  
</LinearLayout>
```

ViewPagerIndicator

ViewPagerIndicator

viewpagerindicator.com

NineOldAndroids

NineOldAndroids

- Backports most of new animation

NineOldAndroids

- Backports most of new animation
- Limitations are platform-constrained

NineOldAndroids

- Backports most of new animation
- Limitations are platform-constrained
- Most useful for transitional animation

NineOldAndroids

NineOldAndroids

```
// Native
ValueAnimator colorAnim =
    ObjectAnimator.ofInt(this, "backgroundColor", RED, BLUE);
colorAnim.setDuration(3000);
colorAnim.setEvaluator(new ArgbEvaluator());
colorAnim.setRepeatCount(ValueAnimator.INFINITE);
colorAnim.setRepeatMode(ValueAnimator.REVERSE);
colorAnim.start();
```

NineOldAndroids

NineOldAndroids

```
// NineOldAndroids
ValueAnimator colorAnim =
    ObjectAnimator.ofInt(this, "backgroundColor", RED, BLUE);
colorAnim.setDuration(3000);
colorAnim.setEvaluator(new ArgbEvaluator());
colorAnim.setRepeatCount(ValueAnimator.INFINITE);
colorAnim.setRepeatMode(ValueAnimator.REVERSE);
colorAnim.start();
```

NineOldAndroids

NineOldAndroids

```
// Native
import android.animation.ArgbEvaluator;
import android.animation.ObjectAnimator;
import android.animation.ValueAnimator;

ValueAnimator colorAnim =
    ObjectAnimator.ofInt(this, "backgroundColor", RED, BLUE);
colorAnim.setDuration(3000);
colorAnim.setEvaluator(new ArgbEvaluator());
colorAnim.setRepeatCount(ValueAnimator.INFINITE);
colorAnim.setRepeatMode(ValueAnimator.REVERSE);
colorAnim.start();
```

NineOldAndroids

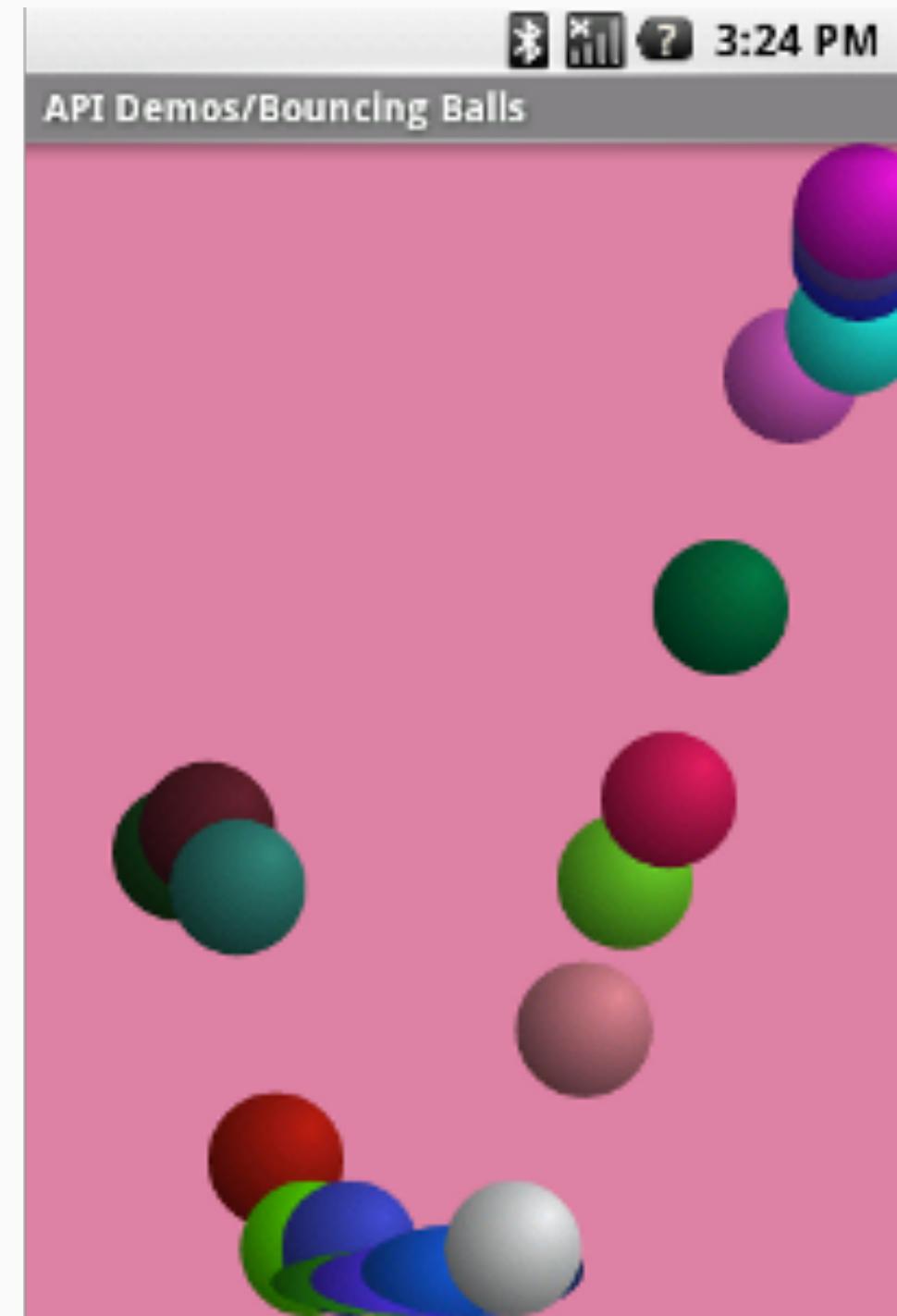
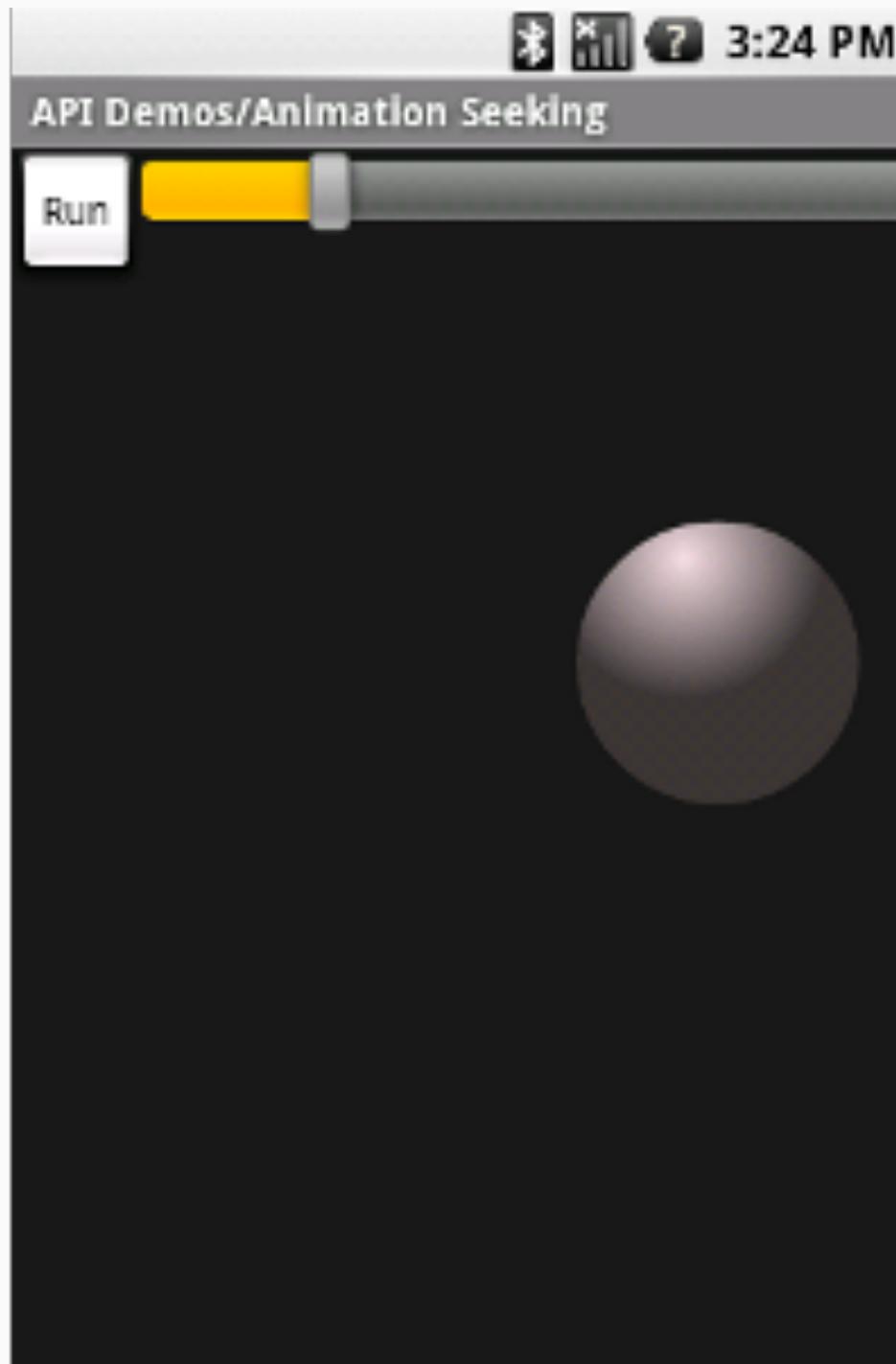
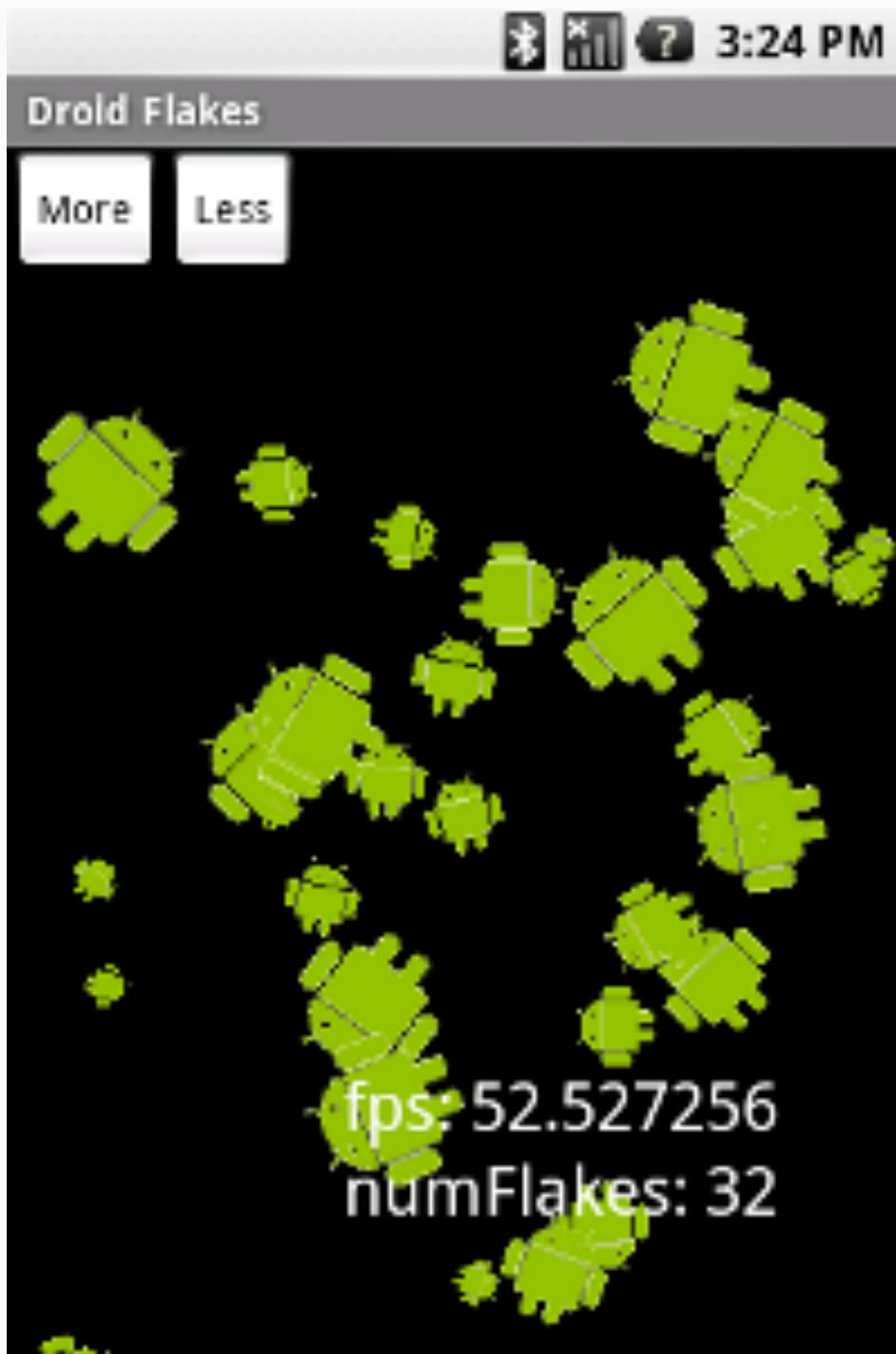
NineOldAndroids

```
// NineOldAndroids
import com.nineoldandroids.animation.ArgbEvaluator;
import com.nineoldandroids.animation.ObjectAnimator;
import com.nineoldandroids.animation.ValueAnimator;

ValueAnimator colorAnim =
    ObjectAnimator.ofInt(this, "backgroundColor", RED, BLUE);
colorAnim.setDuration(3000);
colorAnim.setEvaluator(new ArgbEvaluator());
colorAnim.setRepeatCount(ValueAnimator.INFINITE);
colorAnim.setRepeatMode(ValueAnimator.REVERSE);
colorAnim.start();
```

NineOldAndroids

NineOldAndroids



NineOldAndroids

NineOldAndroids

```
view.animate().alpha(0);
view.animate().translationX(500);
view.animate().alpha(0).translationX(500);
```

NineOldAndroids

```
view.animate().alpha(0);
view.animate().translationX(500);
view.animate().alpha(0).translationX(500);
```

```
animate(view).alpha(0);
animate(view).translationX(500);
animate(view).alpha(0).translationX(500);
```

NineOldAndroids

```
view.animate().alpha(0);
view.animate().translationX(500);
view.animate().alpha(0).translationX(500);
```

```
animate(view).alpha(0);
animate(view).translationX(500);
animate(view).alpha(0).translationX(500);
```

```
import static
    com.nineoldandroids.view.ViewPropertyAnimator.animate;
```

NineOldAndroids

NineOldAndroids

nineoldandroids.com

Otto

Otto

- Forked from Guava's EventBus

Otto

- Forked from Guava's EventBus
- Thread enforced API

Otto

- Forked from Guava's EventBus
- Thread enforced API
- Eliminated class hierarchy walking (for now...)

Otto

Otto

```
@Subscribe  
public void locationUpdated(LocationEvent e) {  
    // ...  
}
```

Otto

```
@Subscribe  
public void locationUpdated(LocationEvent e) {  
    // ...  
}  
  
bus.register(this);
```

Otto

```
@Subscribe  
public void locationUpdated(LocationEvent e) {  
    // ...  
}  
  
bus.register(this);  
  
bus.post(new LocationEvent(...));
```

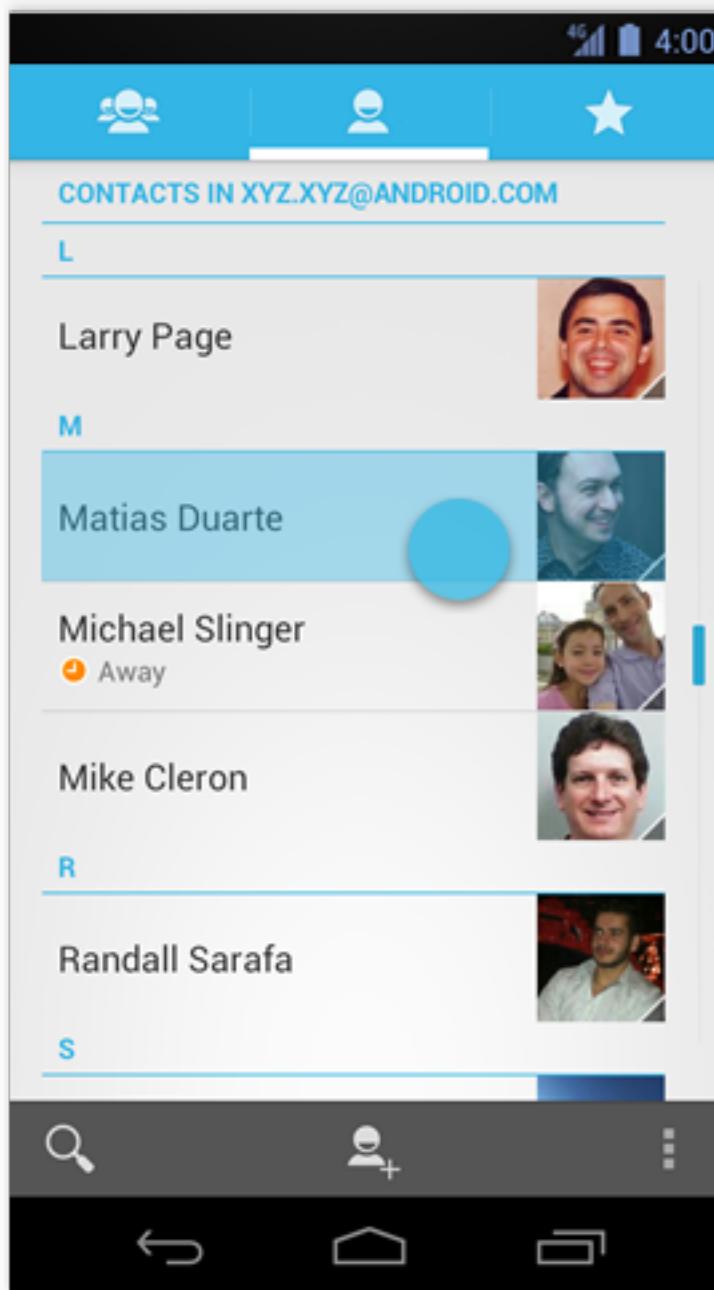
Otto

Otto

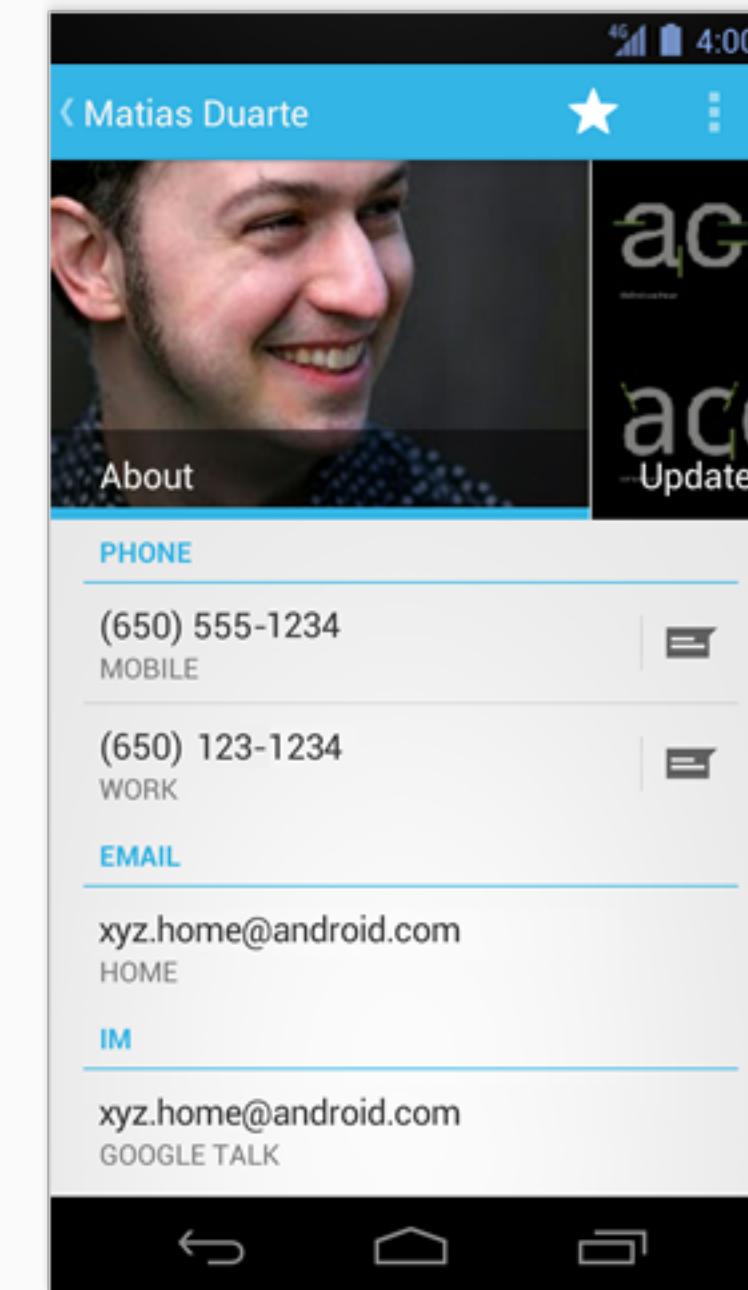
```
public class LocationService extends Service {  
  
    private Location lastLocation;  
  
    @Produce  
    public LocationEvent produceLastLocation() {  
        return lastLocation;  
    }  
  
}
```

Otto

Otto



List view



Detail view

Otto

Otto

GROUPS ALL FAVORITES

L

Larry Page

M

Marissa Mayer

Martha Stewart

Matias Duarte

Matt Brittin

Matt Waddell

Michael Slinger

Mike Cleron

N

Natascha Bock

O

PHONE

(650) 555-1234 MOBILE

(650) 123-1234 WORK

EMAIL

xyz.home@android.com HOME

IM

xyz.home@android.com GOOGLE TALK

WEBSITE

www.google.com

Matias Duarte

RECENT UPDATES

To kick off the new year, here's a very nice in depth piece on Roboto.

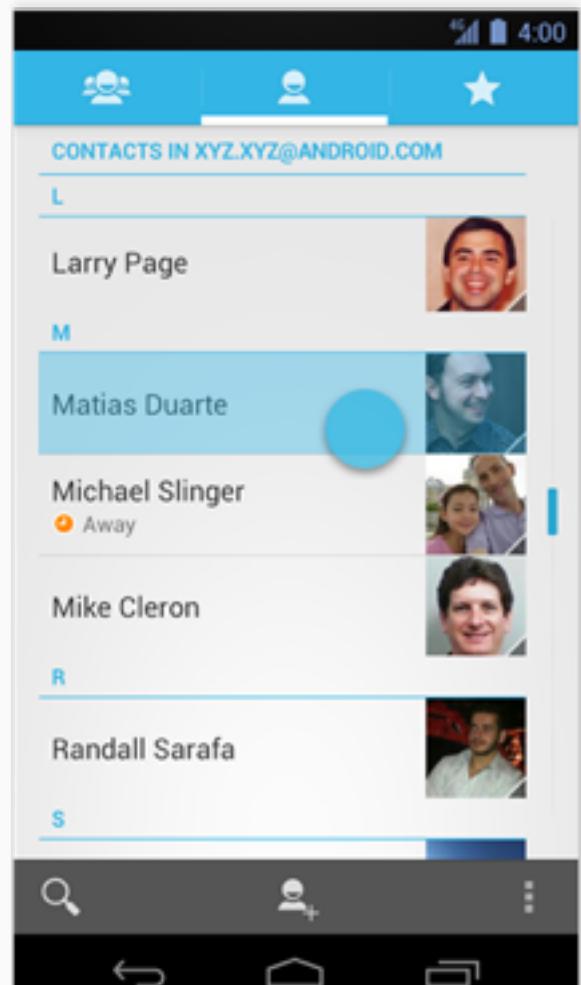
Domo Arigato, Mr Roboto - Boing Boing

Domo Arigato, Mr. Roboto. By Glenn Fleishman. Monday, January 2, 2012 • Prefer dark text? "I can't wake up one morning and say, 'Screw the letter B,'" type designer Matthew Carter told me ... 18 hours ago via Google+ 19 90

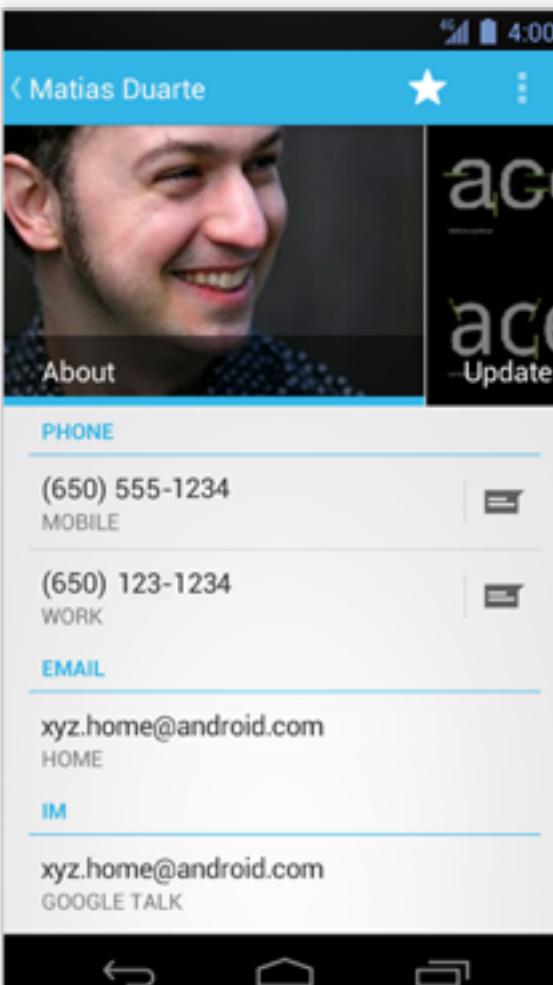
4:00

Otto

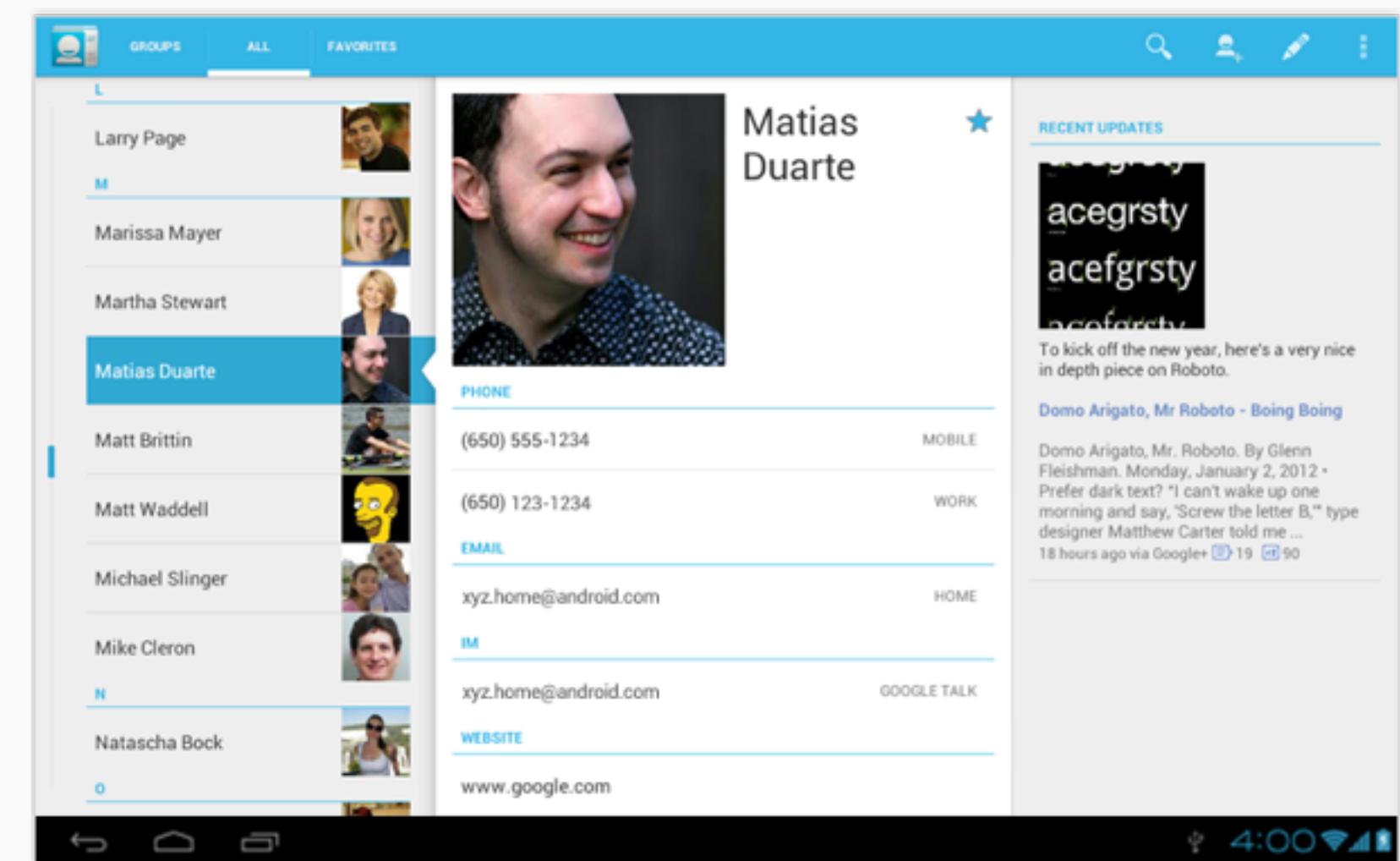
Otto



List view



Detail view



4:00

Otto

Otto

square.github.com/otto

Dagger

Dagger

- JSR330 implementation using APT

Dagger

- JSR330 implementation using APT
- No reflection

Dagger

- JSR330 implementation using APT
- No reflection
- Android helpers for activities

Dagger

Dagger

```
@Module
public class ThisModule {

    @Provide @Singleton Bus provideBus() {
        return new Bus();
    }

    @Provide CoolThings provideAllTheThings() {
        return new CoolThings("Devoxx 2012");
    }
}
```

Dagger

Dagger

```
og = ObjectGraph.get(  
    new ThisModule(),  
    new ThatModule(),  
    new ActivitiesModule()  
) ;
```

Dagger

Dagger

```
public class BaseActivity extends Activity {  
  
    public void onCreate(..) {  
        super.onCreate(..);  
        ((MyApp) getApplication()).inject(this);  
    }  
  
}
```

Dagger

Dagger

```
public class DevoxxActivity extends BaseActivity
{
    @Inject CoolThings coolThings;
    @Inject Bus bus;

    public void onCreate(..) {
        super.onCreate(..)
        // At this point we are injected so...
        coolThings.doThem();
        bus.post(new DidCoolThingsEvent());
    }
}
```

Dagger

Dagger

square.github.com/dagger

Retrofit

Retrofit

- HTTP API as interfaces

Retrofit

- HTTP API as interfaces
- Declarative annotations

Retrofit

- HTTP API as interfaces
- Declarative annotations
- Type-safe, pluggable converters

Retrofit

Retrofit

```
class Tweet {  
    String text;  
}
```

Retrofit

```
class Tweet {  
    String text;  
}  
  
interface TweetService {  
    @GET("statuses/user_timeline.json")  
    List<Tweet> tweets(@Named("screen_name") String user);  
}
```

Retrofit

Retrofit

```
TweetService tweetService =  
    restAdapter.create(TweetService.class);
```

Retrofit

```
TweetService tweetService =  
    restAdapter.create(TweetService.class);
```

```
List<Tweet> tweets =  
    tweetService.tweets("JakeWharton");
```

Retrofit

Retrofit

```
interface TweetService {  
    @GET("statuses/user_timeline.json")  
    void tweets(@Named("screen_name") String user,  
                Callback<List<Tweet>> c);  
}
```

Retrofit

```
interface TweetService {
    @GET("statuses/user_timeline.json")
    void tweets(@Named("screen_name") String user,
                Callback<List<Tweet>> c);
}

tweetService.tweets("JakeWharton",
    new Callback<List<Tweet>>() {
        @Override public void call(List<Tweet> tweets) {
            // ...
        }
    }
);
```

Retrofit

Retrofit

github.com/square/retrofit

Dagger + Otto + Retrofit + GSON

Dagger + Otto + Retrofit + GSON

```
class FishPantsModule {  
    @Provide @Singleton RestAdapter provideRestAdapter() {  
        return new RestAdapter.Builder()  
            .setThings(..)  
            .build();  
    }  
  
    @Provide @Singleton Bus provideBus() {  
        return new Bus();  
    }  
  
    @Provide @Singleton TweetService provideTweets(RestAdapter ra) {  
        return ra.create(TweetService.class);  
    }  
}
```

Dagger + Otto + Retrofit + GSON

Dagger + Otto + Retrofit + GSON

```
class ErmahgerdActivity extends BaseActivity {

    @Inject TweetService tweetService;
    @Inject Bus bus;

    @Override public void onCreate(..) {
        super.onCreate(..);
        tweetService.tweets("JakeWharton",
            new Callback<List<Tweet>>() {
                @Override public void call(List<Tweet> tweets) {
                    bus.post(new TweetsLoadedEvent(tweets));
                }
            }
        );
    }
}
```

Dagger + Otto + Retrofit + GSON

Dagger + Otto + Retrofit + GSON

```
@Module(overrides = true)
class TestModule {
    @Provide @Singleton TweetService provideTweets() {
        return new TestTweetService();
    }
}
```

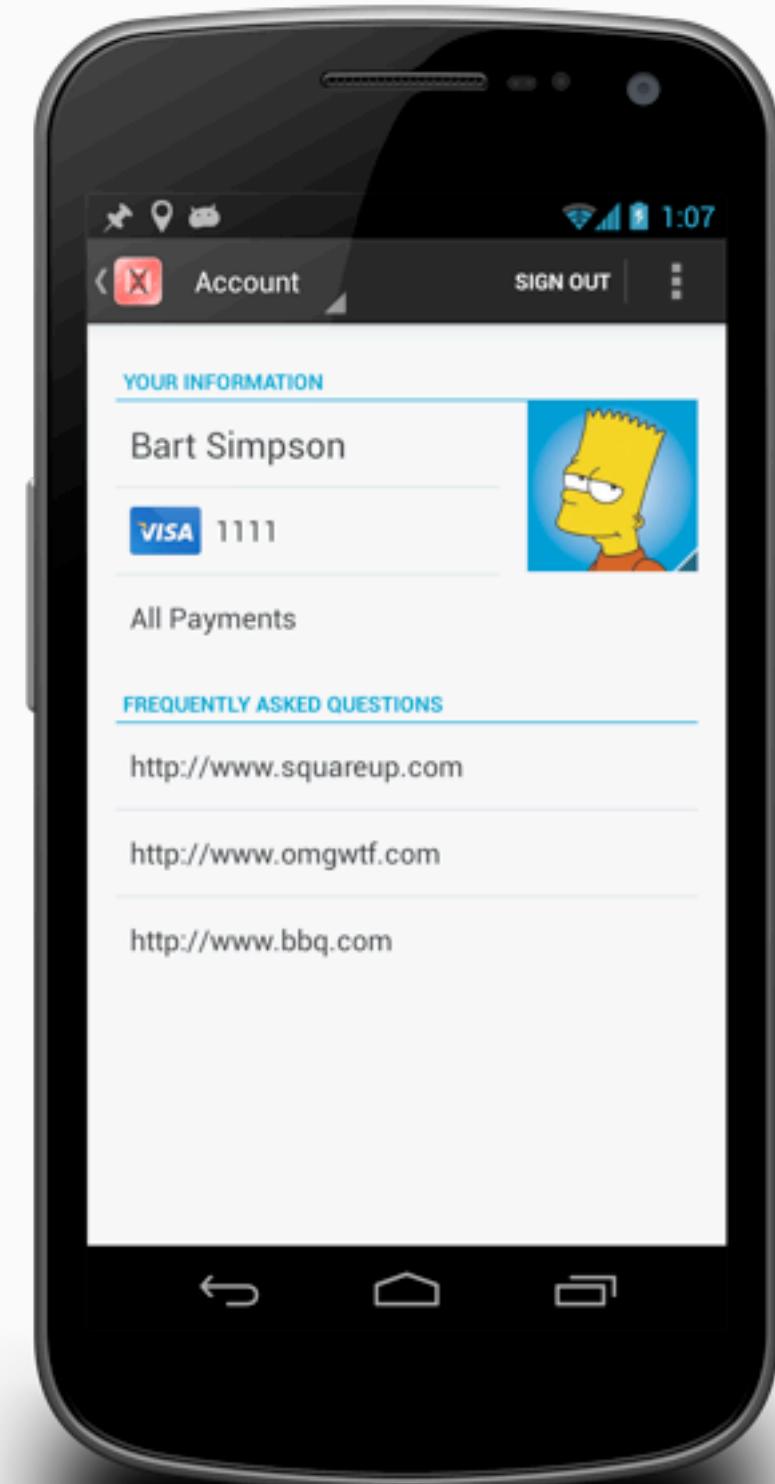
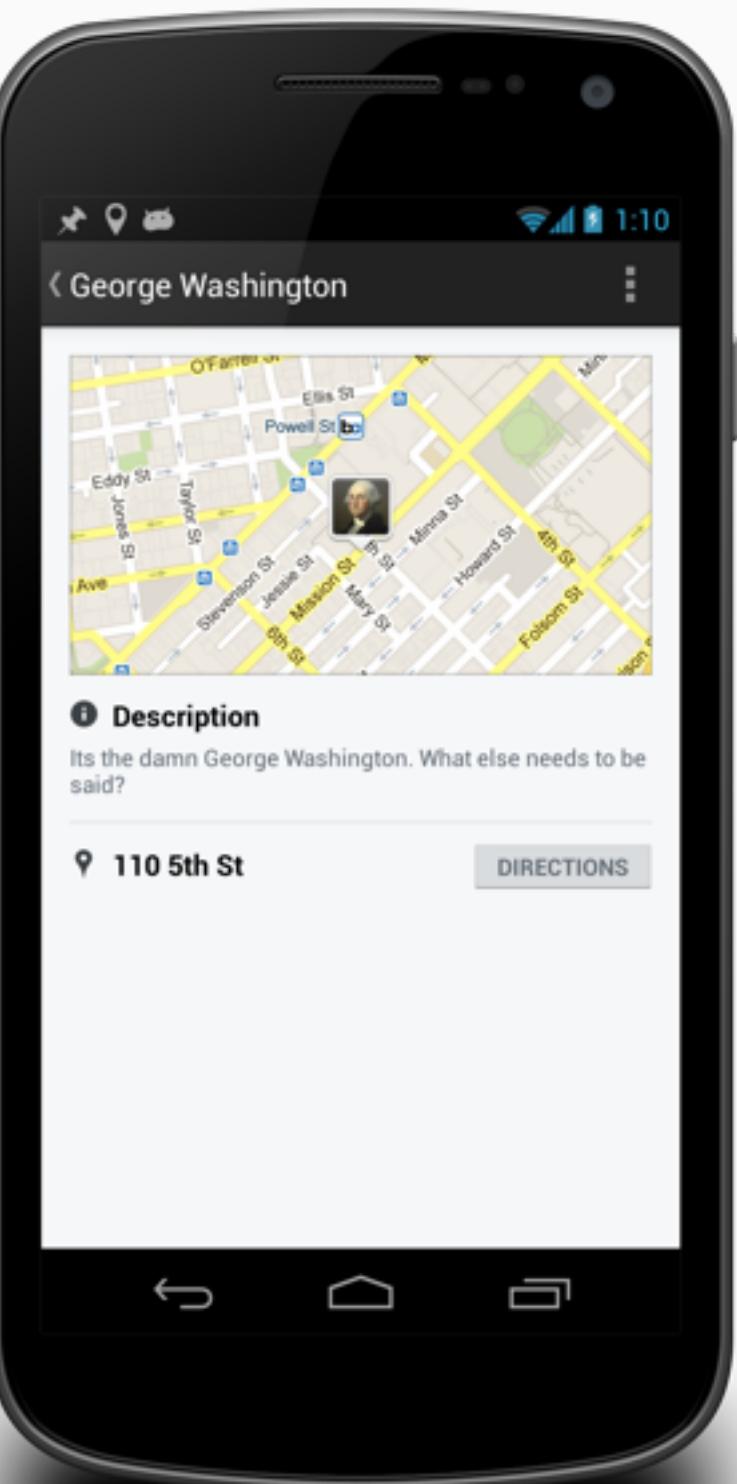
Dagger + Otto + Retrofit + GSON

```
@Module(overrides = true)
class TestModule {
    @Provide @Singleton TweetService provideTweets() {
        return new TestTweetService();
    }
}

class TestTweetService implements TweetService {
    @Override public void tweets(String name, Callback<List<Tweet>> cb) {
        cb.call(Arrays.asList(
            new Tweet("Hello"),
            new Tweet("Test"),
            new Tweet("Data!")
        ));
    }
}
```

Dagger + Otto + Retrofit + GSON

Dagger + Otto + Retrofit + GSON



Tape

Tape

- File-based FIFO

Tape

- File-based FIFO
- Serialization of choice

Tape

- File-based FIFO
- Serialization of choice
- Queue helpers

Tape

Tape

```
public class ChargeCreditCard implements Runnable
{
    private final Money amount;
    private final String to;

    public ChargeCreditCard(..) { .. }

    @Override public void run() {
        // Send the money.
    }
}
```

Tape

Tape

```
public class ChargeCreditCard
    implements Task<Callback> {

    private final Money amount;
    private final String to;

    public ChargeCreditCard(..) { .. }

    @Override public void execute(Callback c) {
        // Send the money
        c.success();
    }
}
```

Tape

Tape

square.github.com/tape

Use Case: Paying with Square

Use Case: Paying with Square

- Dagger

Use Case: Paying with Square

- Dagger
- GSON

Use Case: Paying with Square

- Dagger
- GSON
- Otto

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing
subset)

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape
- JUnit

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape
- JUnit
- Fest

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape
- JUnit
- Fest
- Robolectric (BOO!)

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape
- JUnit
- Fest
- Robolectric (BOO!)
- Robotium

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape
- JUnit
- Fest
- Robolectric (BOO!)
- Robotium
- Mockito

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape
- JUnit
- Fest
- Robolectric (BOO!)
- Robotium
- Mockito
- MockWebServer

Use Case: Paying with Square

- Dagger
- GSON
- Otto
- ActionBarSherlock
- Retrofit
- OkHttp
- Pollexor
- ActivityCompat2
- PonyCrossing (ZXing subset)
- Tape
- JUnit
- Fest
- Robolectric (BOO!)
- Robotium
- Mockito
- MockWebServer
- Spoon

Spoon

Spoon

- Facilitates instrumentation test execution

Spoon

- Facilitates instrumentation test execution
- Aggregates results for high-level and drill-down view

Spoon

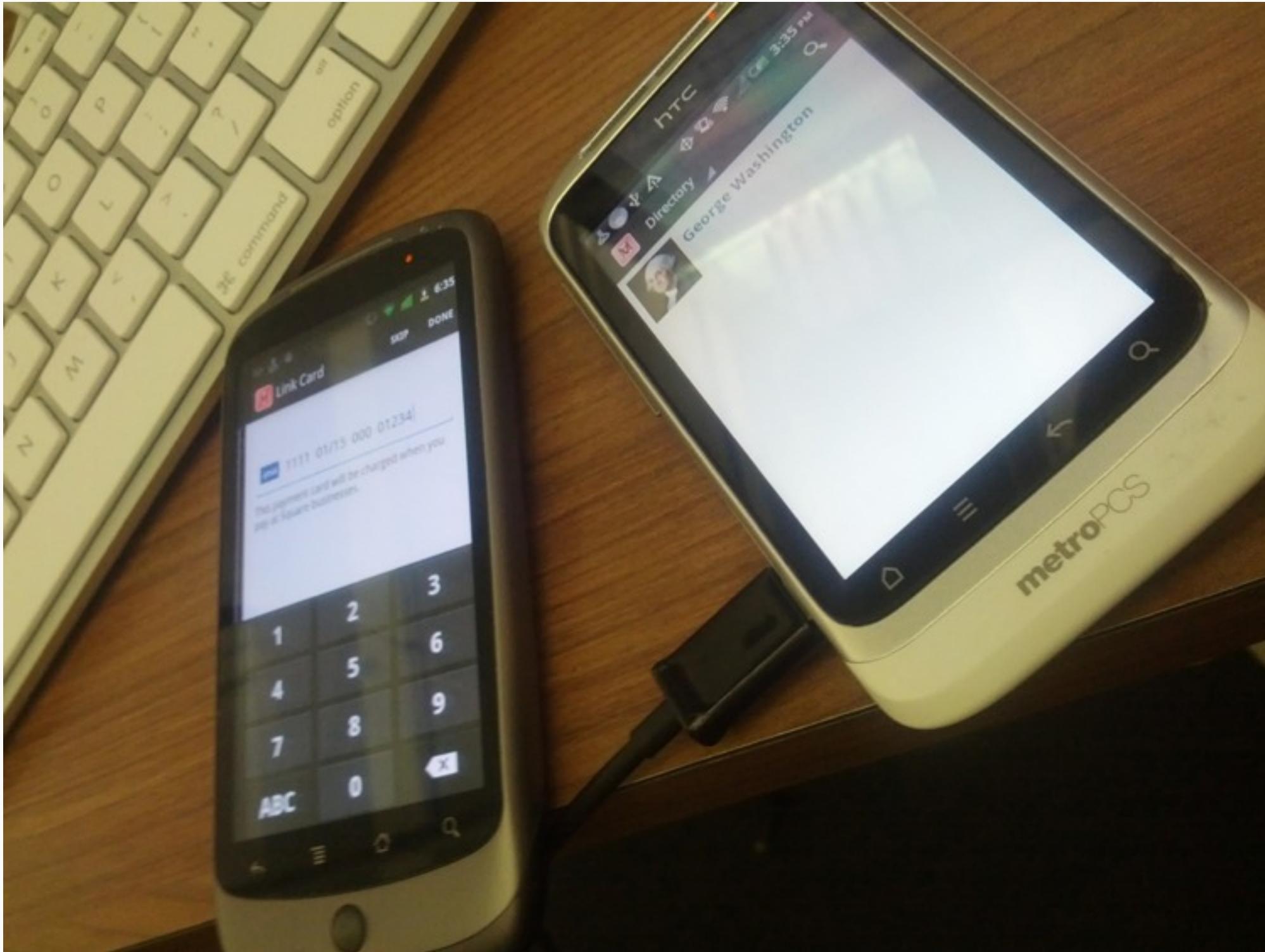
- Facilitates instrumentation test execution
- Aggregates results for high-level and drill-down view
- Captures and compares screenshots

Spoon

- Facilitates instrumentation test execution
- Aggregates results for high-level and drill-down view
- Captures and compares screenshots
- Maintains test result and screenshot history

Spoon

Spoon

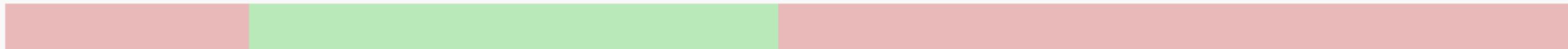


Spoon

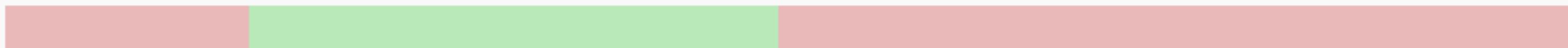
Square Wallet

12 tests run across 2 devices with 4 passing and 8 failing in 12378 seconds at 2012-10-05 07:00:35 PM

Nexus One



HTC-PG762

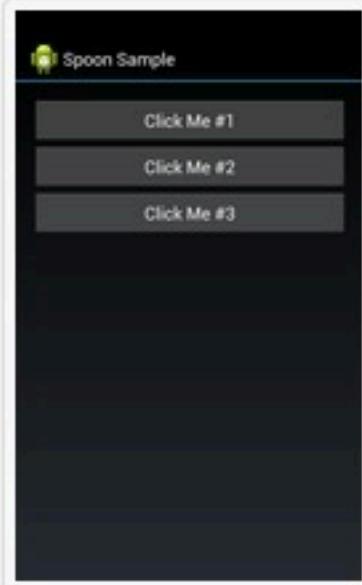


Spoon

Galaxy Nexus

4 tests run with 3 passing and 1 failing in 17 seconds at 2012-11-15 12:50:01 PM

testAlwaysFailingForFun SampleActivityTest



testClickButton1_displaysHello1 SampleActivityTest

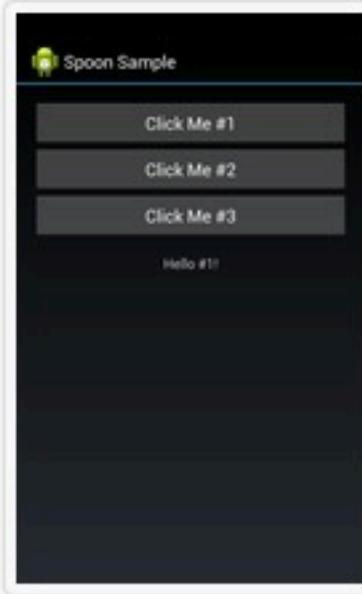


Spoon

testAlwaysFailingForFun SampleActivityTest



testClickButton1_displaysHello1 SampleActivityTest



Why You Should Open Source

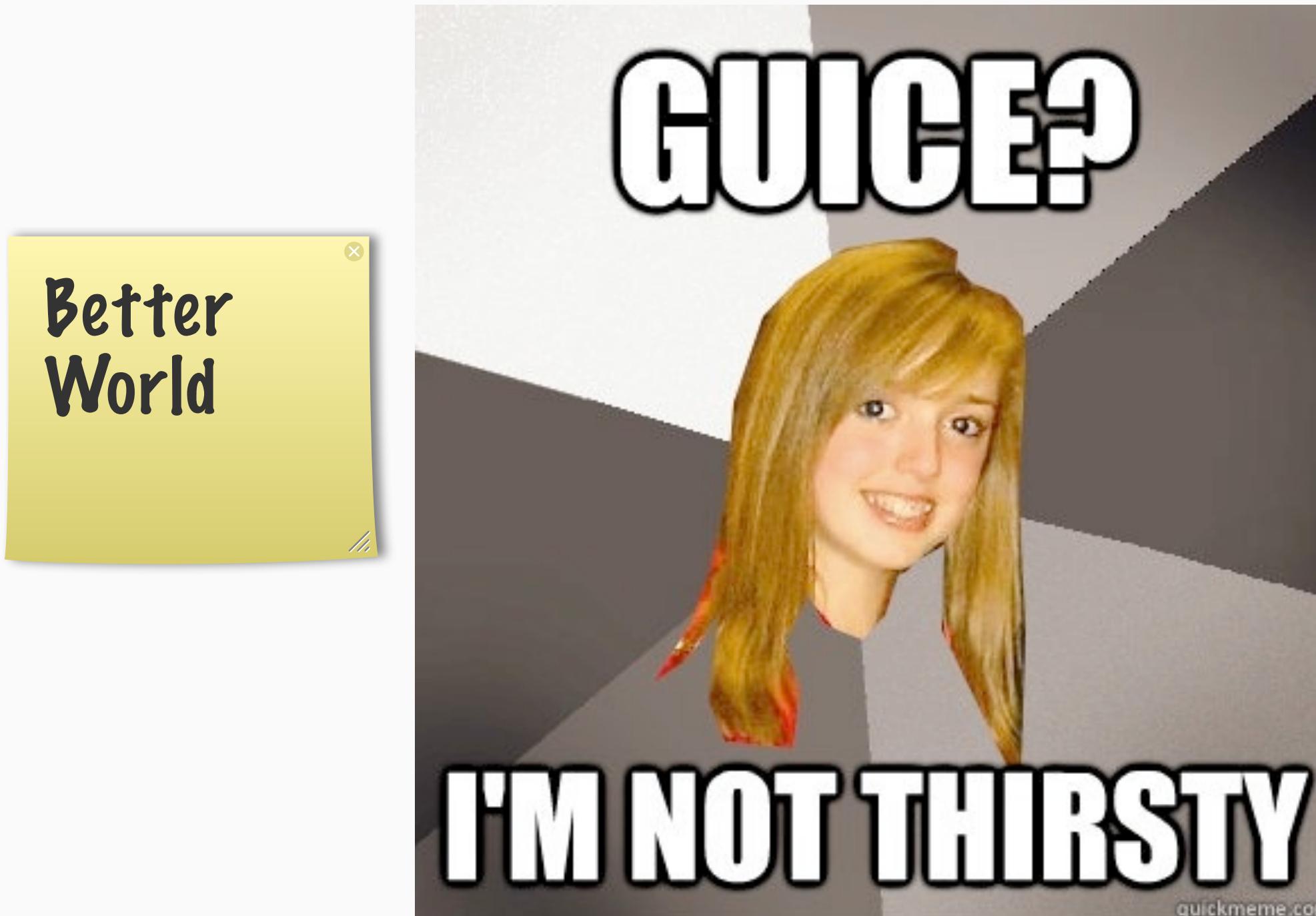
Why You Should Open Source



Why You Should Open Source



Why You Should Open Source



How You Should Open Source

How You Should Open Source



jakewharton.github.com/EffectiveOpenSource/





squareup.com/jobs



Q/A

