

“It’s an Inside Job”

# BUILDING DEBUG FEATURES

• • •



Sam Edwards  
@HandstandSam

# OUTLINE

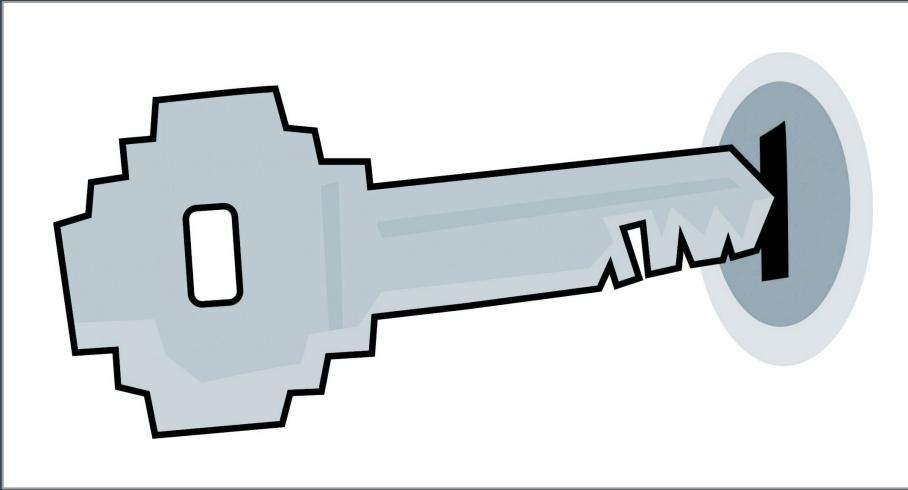
1. WHAT IS A DEBUG FEATURE?
2. GRADLE BUILD VARIANTS
3. EXAMPLE SCENARIOS
4. OPEN SOURCE INSPIRATION
5. DON'T SHIP TO PROD

# WHAT IS A DEBUG FEATURE?

# DEBUG FEATURES

- HELP YOU TEST AND DEBUG
- DON'T SHIP TO PRODUCTION

**AS DEVELOPERS,  
WE HAVE INSIDER ACCESS TO BUILD THEM**



**DEBUG FEATURES IN YOUR APP ENABLE YOUR  
TEAM TO EFFECTIVELY PERFORM TASKS THAT CAN'T BE  
AUTOMATED.**

# DEBUG FEATURE POSSIBILITY MATRIX

Modify    Mock    Monitor    Capture    Inspect

Configurations

Persistent Data

Networking

Resources

Views

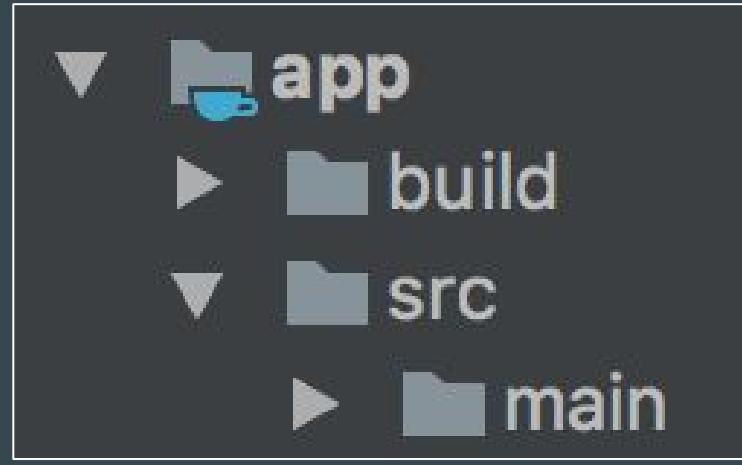
Notifications

Sensors

... and more.

# GRADLE BUILD VARIANTS

# “main” SOURCE FOLDER



# BuildConfig.java

```
public final class BuildConfig {  
    public static final boolean DEBUG = Boolean.parseBoolean( s: "true" );  
}
```

# DEFAULTS

“main”

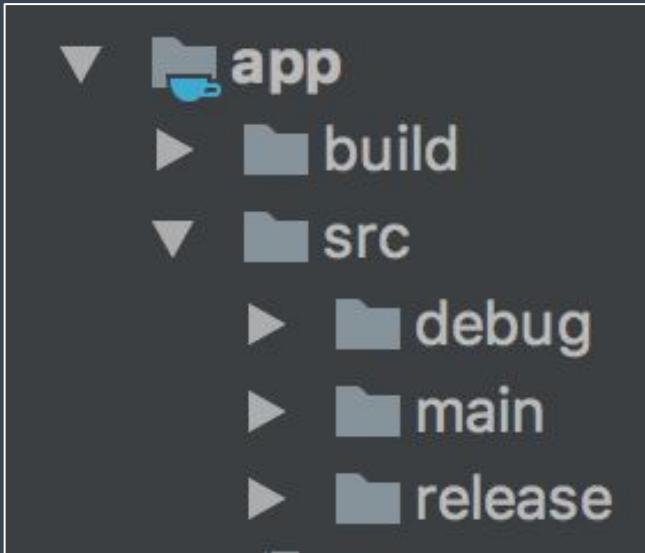
=

DEBUG.APK

=

RELEASE.APK

# “debug” and “release” SOURCE FOLDERS



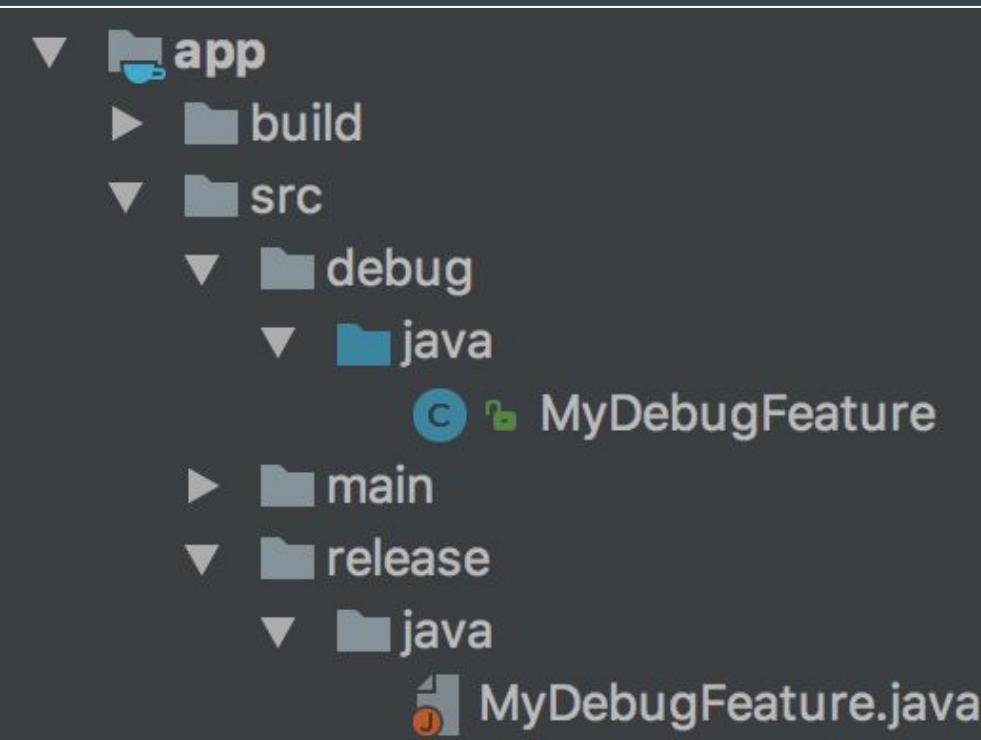
# “debug” & “release” SOURCE FOLDERS

“debug” = DEBUG.APK

“main” +

“release” = RELEASE.APK

# SOURCE STRUCTURE



# “debug”

```
public class MyDebugFeature {  
    public void start(Context context) {  
        Toast.makeText(context, "Debug Version", Toast.LENGTH_SHORT).show();  
    }  
}
```

# “release”

```
public class MyDebugFeature {  
    public void start(Context context) {  
    }  
}
```

# GRADLE COMPILE DEPENDENCIES

```
dependencies {  
    compile 'com.android.support:appcompat-v7:25.2.0'  
    compile 'com.android.support:design:25.2.0'  
}
```

# GRADLE DEPENDENCIES

```
dependencies {  
    compile 'com.android.support:appcompat-v7:25.3.0'  
    debugCompile 'com.example.debugfeature'  
    releaseCompile 'com.example.debugfeature-no-op'  
}
```

# GRADLE DEPENDENCIES

```
dependencies {  
    compile 'com.android.support:appcompat-v7:25.3.0'  
    debugCompile 'com.example.debugfeature'  
    releaseCompile 'com.example.debugfeature-no-op'  
}
```

“debugCompile” = DEBUG.APK

“compile” +

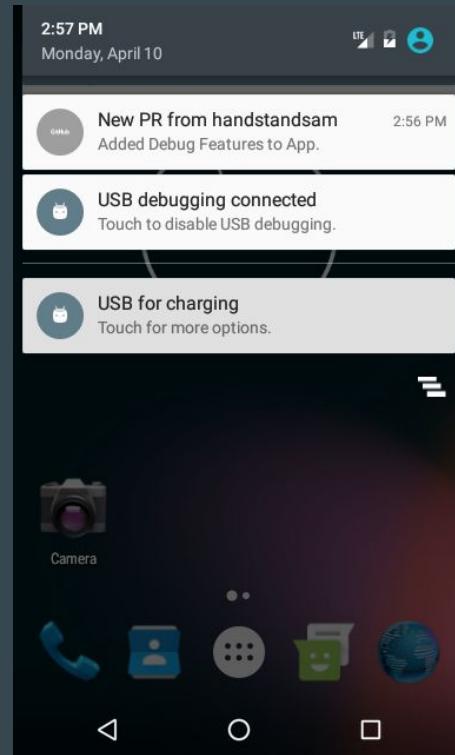
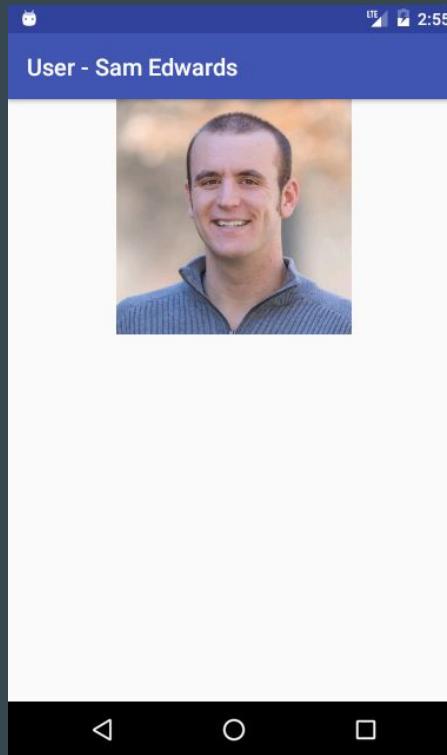
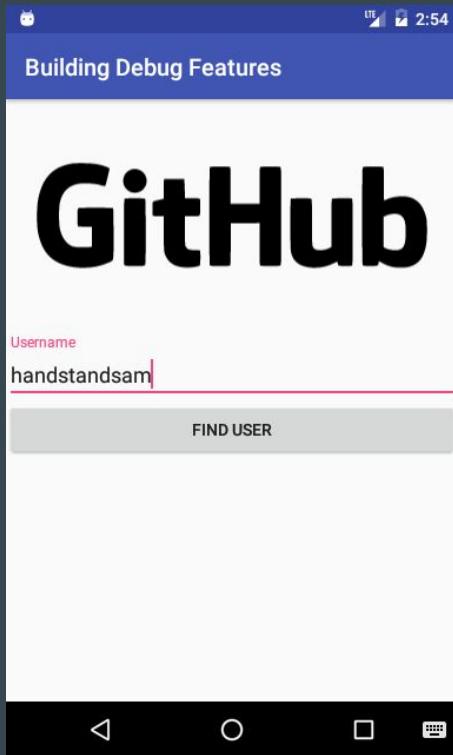
“releaseCompile” = RELEASE.APK

# GENERATED APKS

apk				
Name	^	Date Modified	Size	Kind
 app-debug.apk		Today, 1:50 PM	4.6 MB	Document
 app-release.apk		Today, 1:50 PM	2.8 MB	Document

# EXAMPLE SCENARIO

# SAMPLE APP



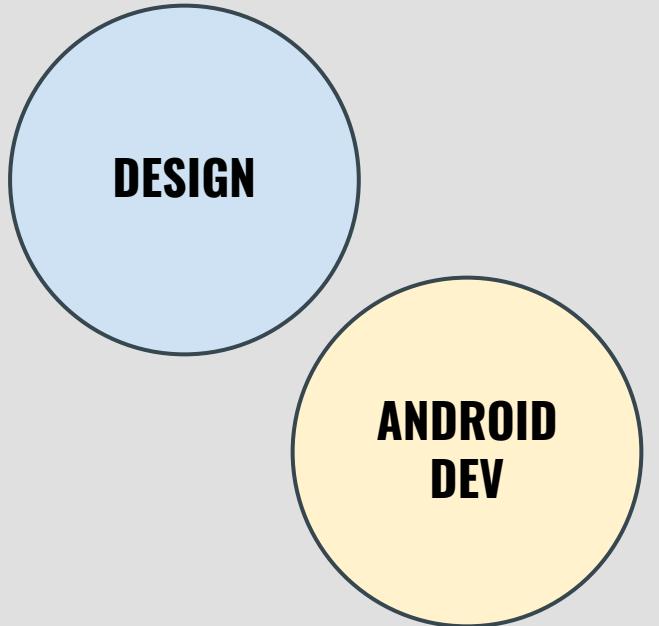
#droidconbos

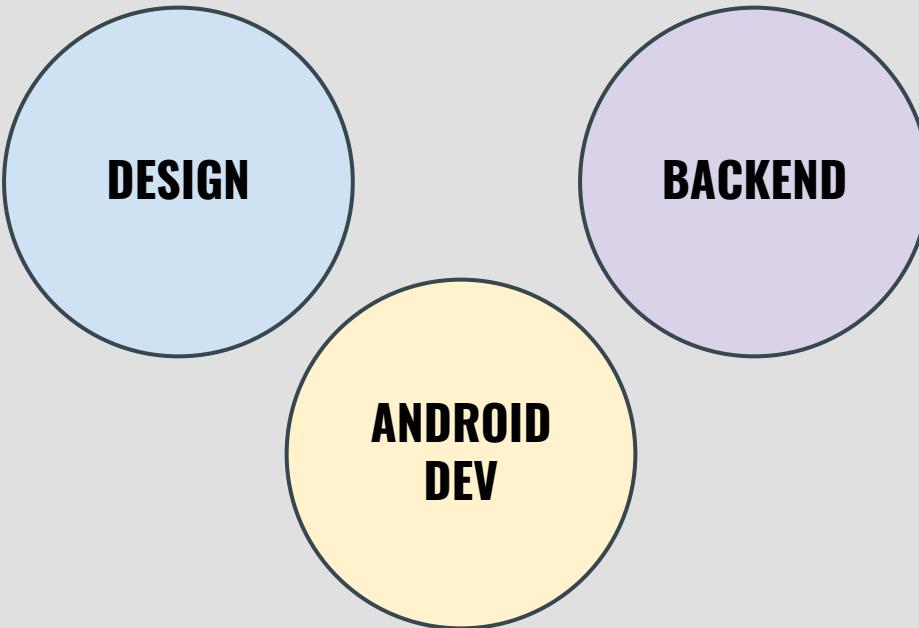
<https://github.com/handstandsam/BuildingDebugFeatures>

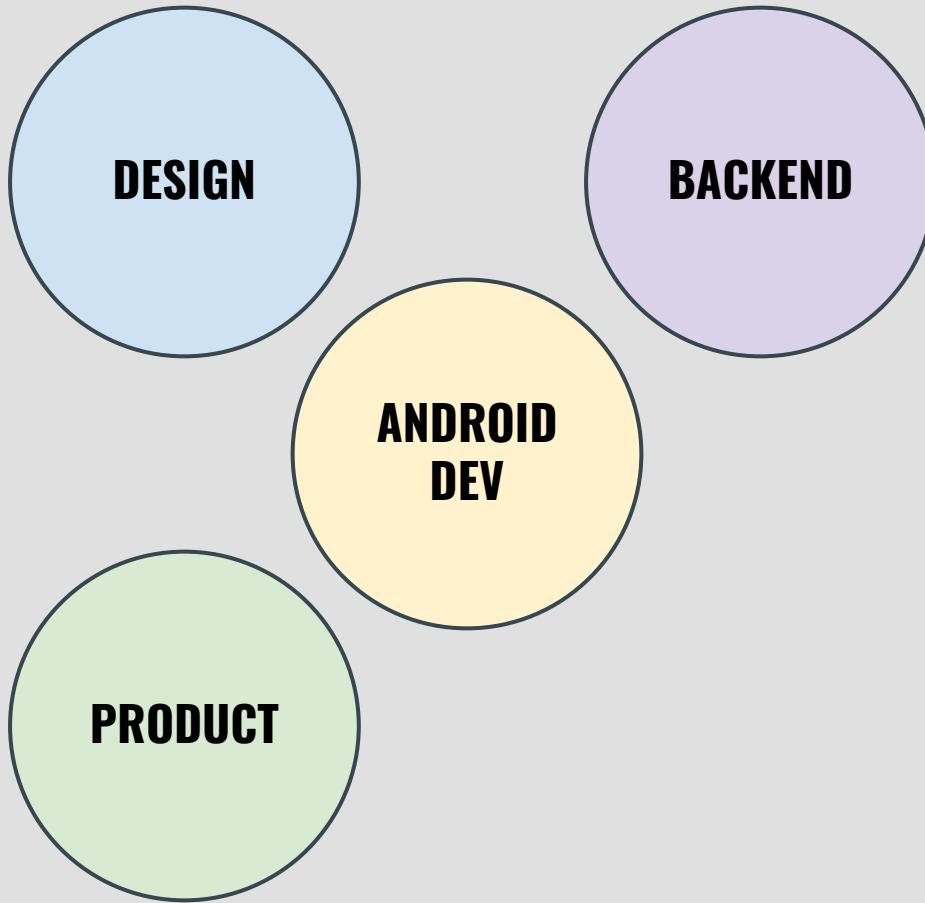
@HandstandSam

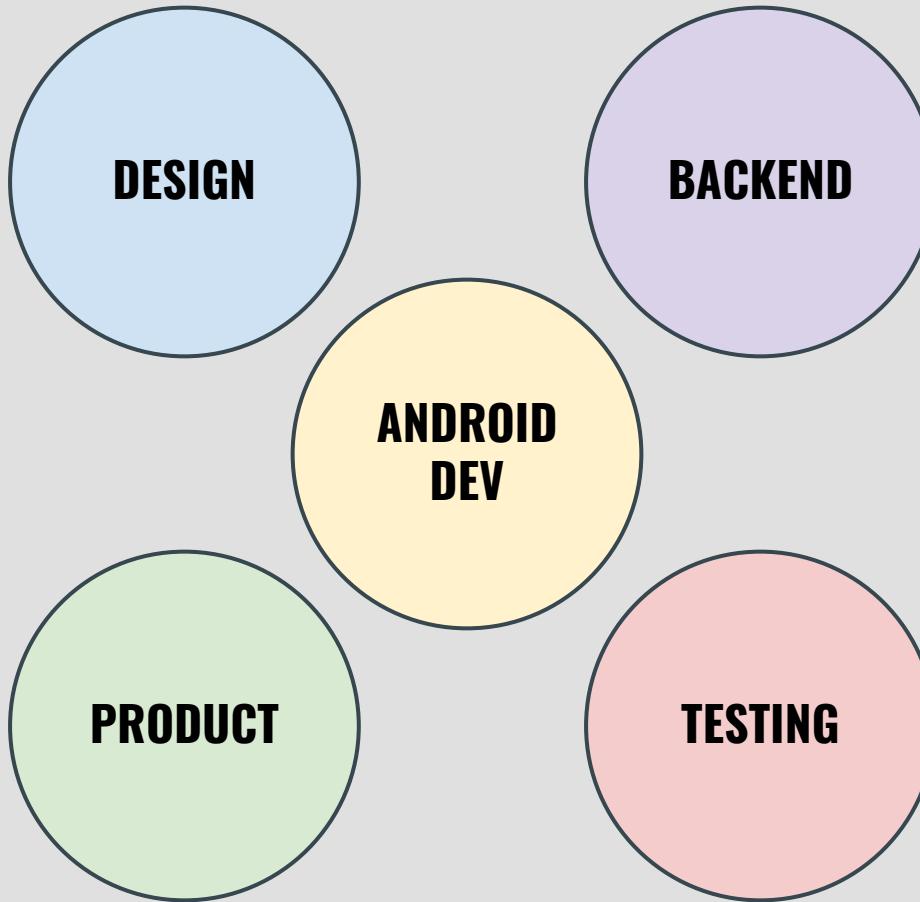


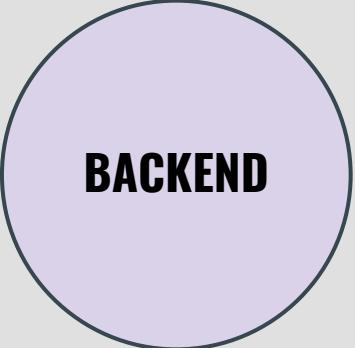
**ANDROID  
DEV**











**BACKEND**

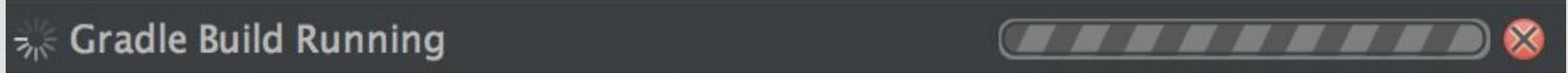
**JOE: “CAN I HAVE A SPECIAL BUILD  
TO TEST A NEW SERVER ENDPOINT?”**

**SURE, BUT IT'LL TAKE A FEW MINUTES.**

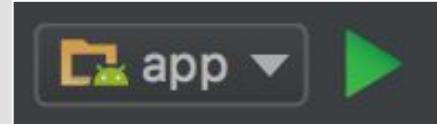
```
@Provides
Retrofit.Builder retrofitBuilder(OkHttpClient.Builder okHttpClientBuilder) {
    Retrofit.Builder builder = new Retrofit.Builder()
        .baseUrl("https://server")
        .addConverterFactory(MoshiConverterFactory.create(new Moshi.Builder().build()))
        .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
        .client(okHttpClientBuilder.build());
    return builder;
}
```

```
@Provides
Retrofit.Builder retrofitBuilder(OkHttpClient.Builder okHttpClientBuilder) {
    Retrofit.Builder builder = new Retrofit.Builder()
        .baseUrl("https://temp-server")
        .addConverterFactory(MoshiConverterFactory.create(new Moshi.Builder().build()))
        .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
        .client(okHttpClientBuilder.build());
    return builder;
}
```

# WAIT FOR GRADLE...



# How many times have you hit



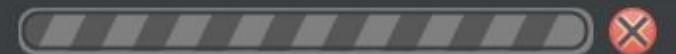
In



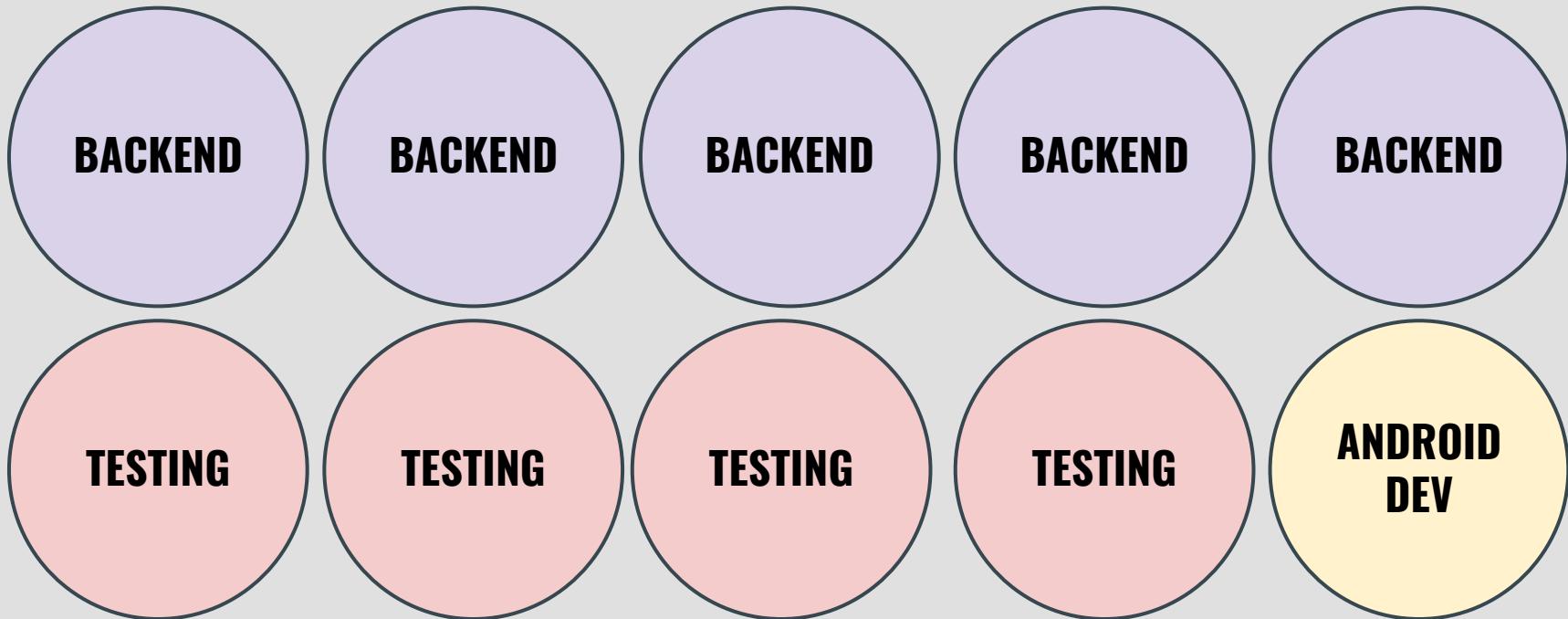
# and then waited for



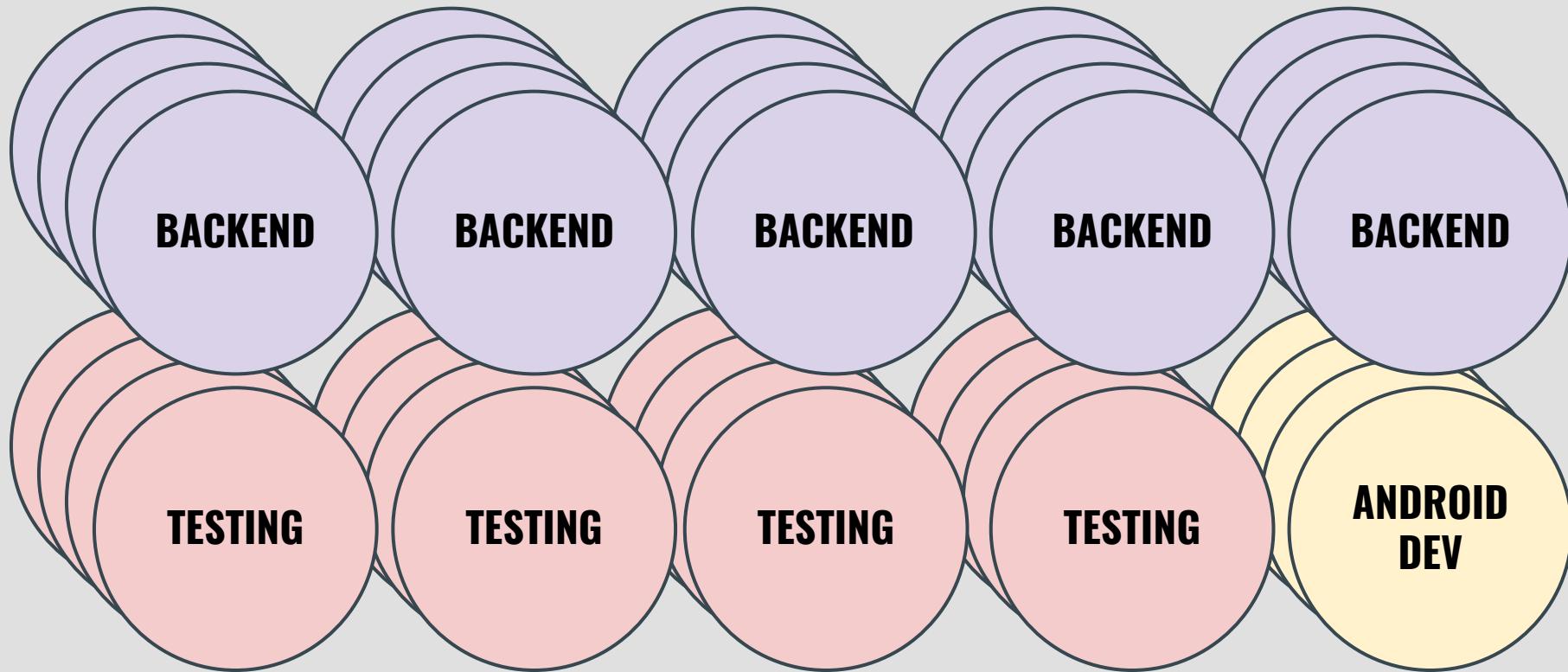
Gradle Build Running



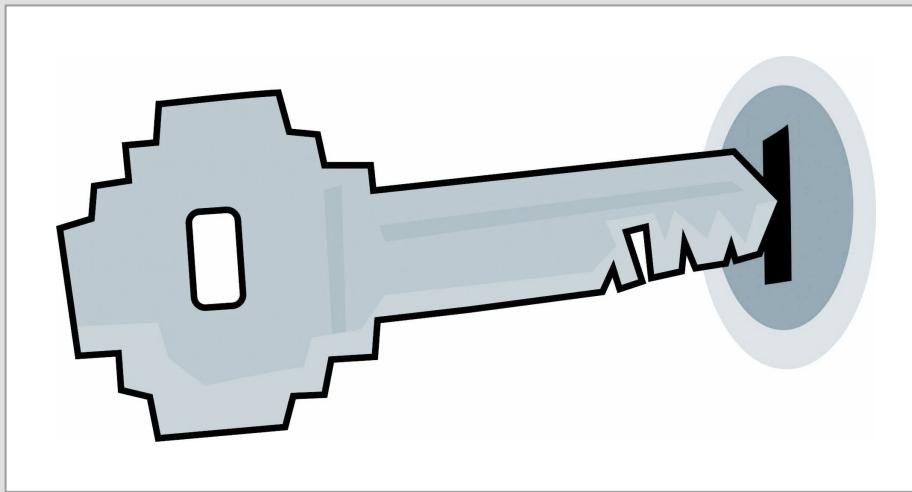
# This **ISN'T** the first time you've been asked for this.



# NOR THE LAST



# YOU CAN ENABLE OTHERS



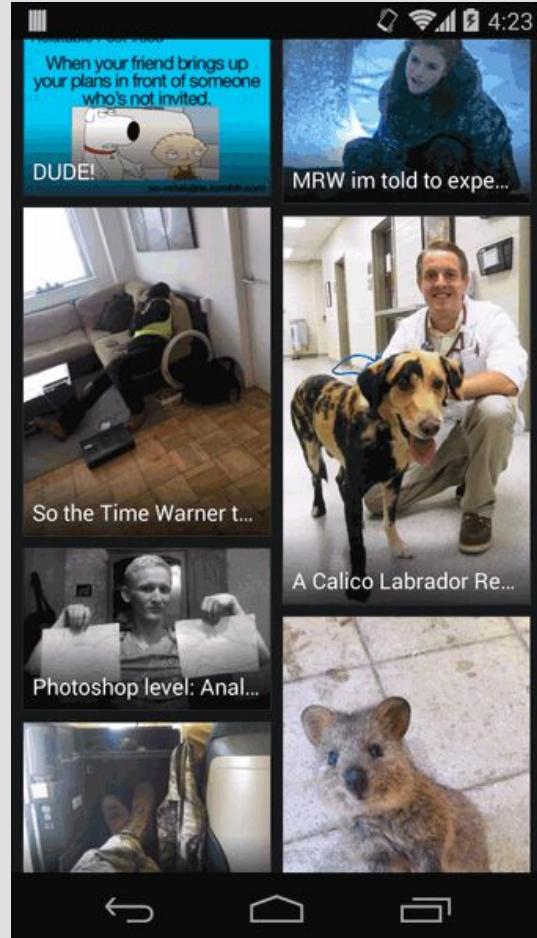
# CUSTOM DEBUG FEATURE “CHANGE ENDPOINT”

# WHAT WILL THIS UI LOOK LIKE?

# DEBUG DRAWER

<https://github.com/JakeWharton/u2020>

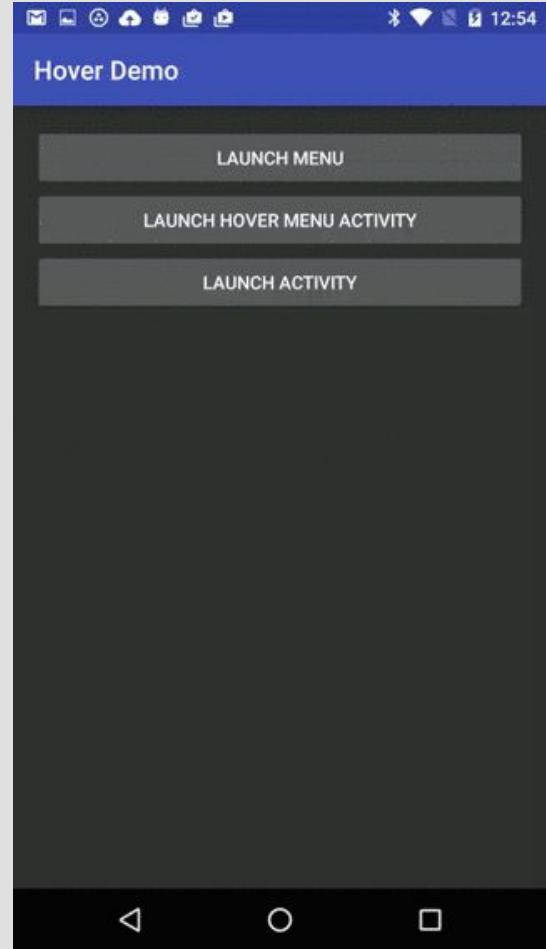
- Requires changes to existing UI hierarchy.
- Could interfere with actions that use side-to-side swiping.



# HOVERING MENU

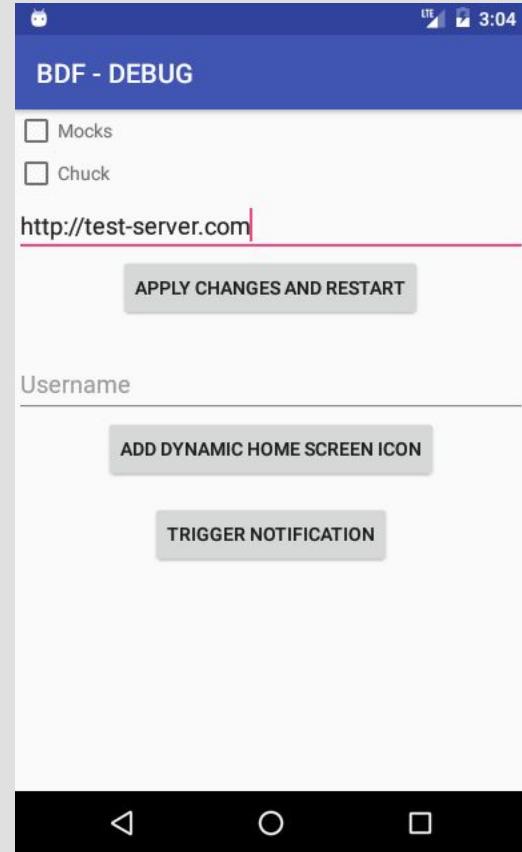
<https://github.com/google-hover>

- REQUIRES NO CODE CHANGES
- LOOKS COOL
- REQUIRES LEARNING A NEW PARADIGM OF USING THE WINDOW MANAGER



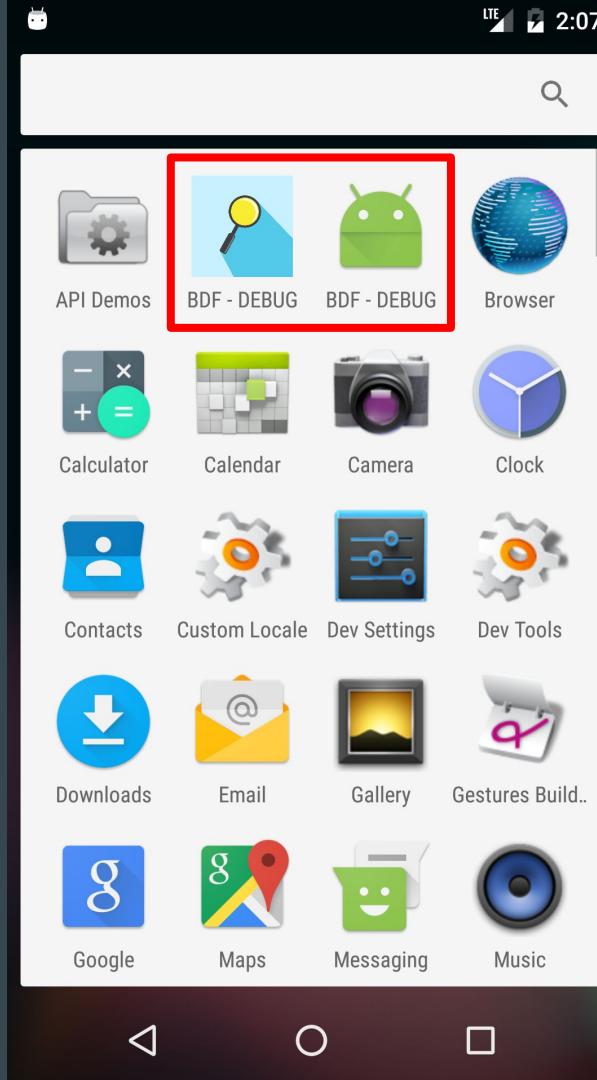
# NEW ACTIVITY

- REQUIRES NO CHANGES TO EXISTING UI CODE
- STRAIGHT FORWARD



**BUT... HOW WILL THEY ACCESS  
THE NEW ACTIVITY?**

# OPTION 1: MULTIPLE LAUNCHER ICONS FOR A SINGLE APP

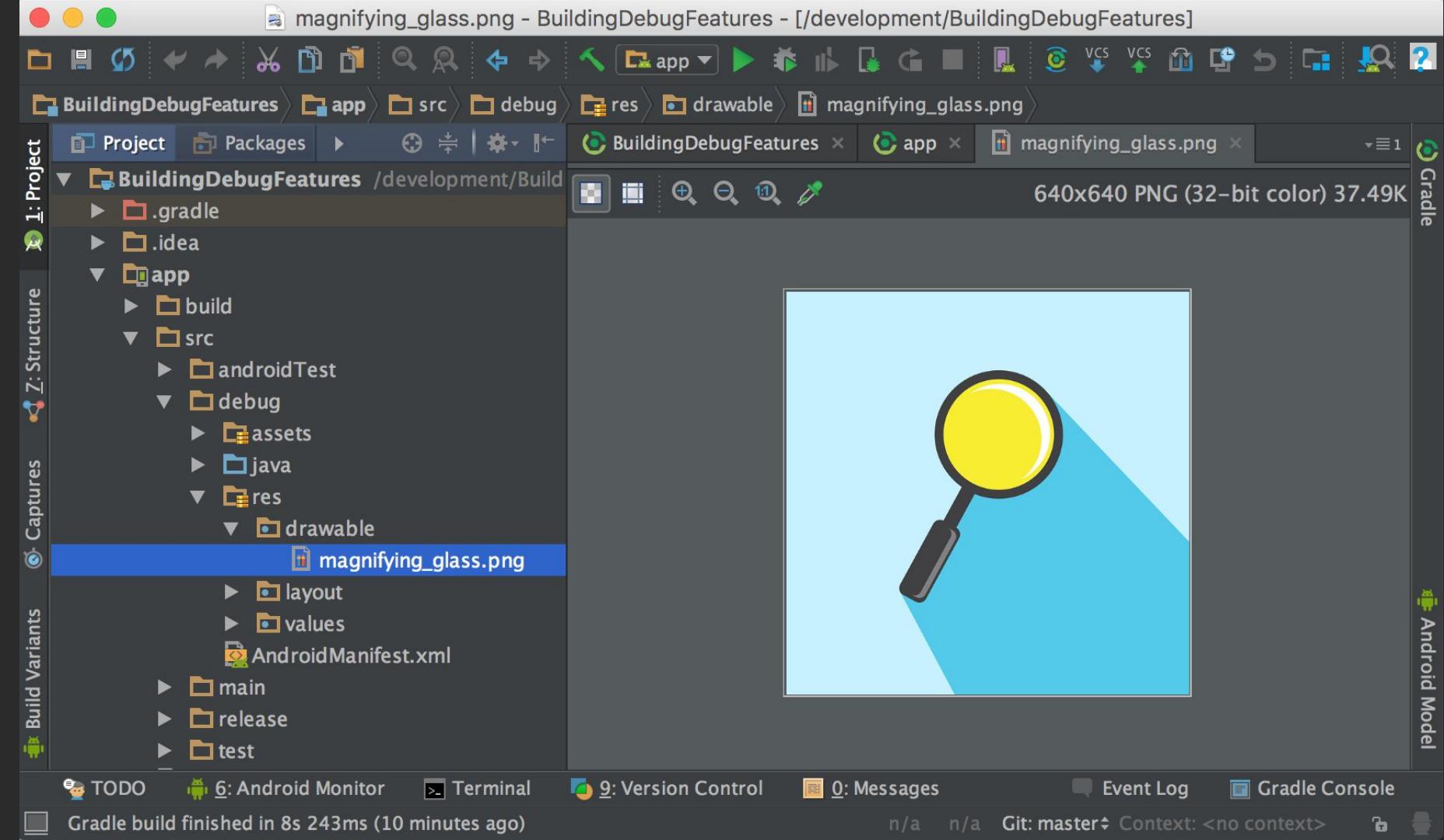


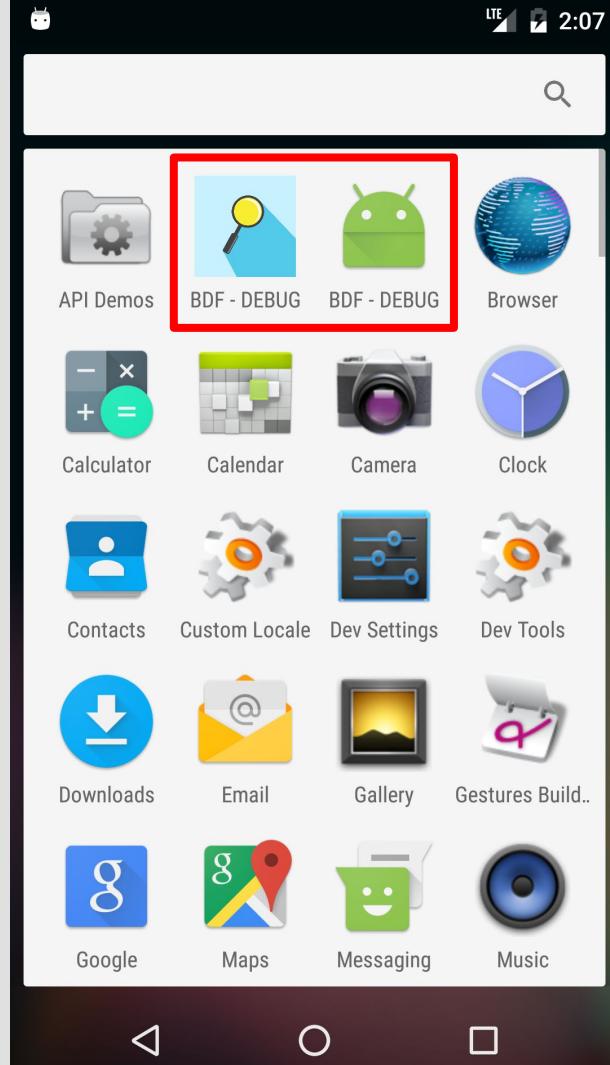
# MAIN ACTIVITY (src/main)

```
<activity
    android:name=".MainActivity"
    android:launchMode="singleTask">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

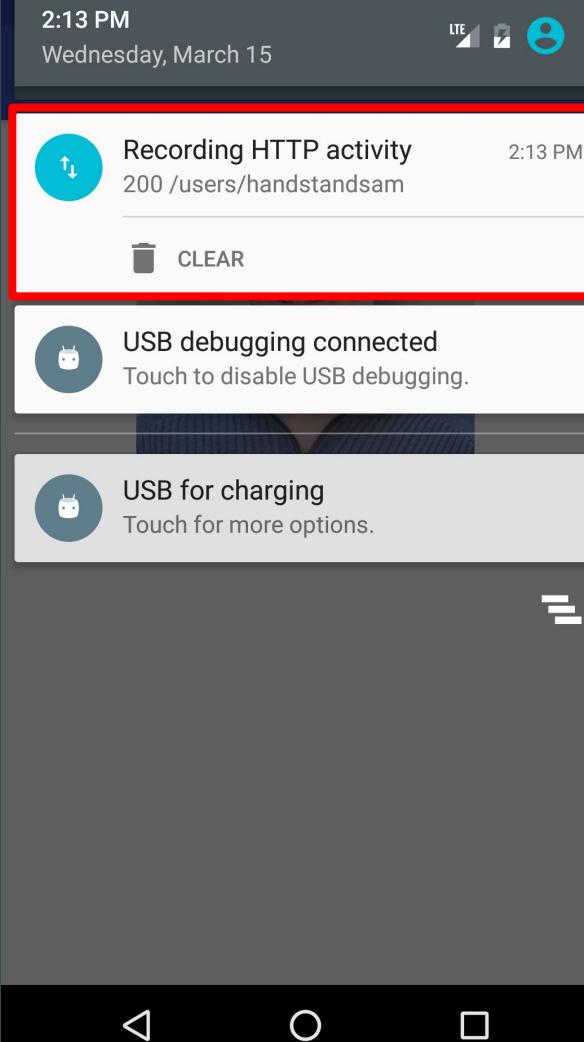
# DEBUG FEATURE ACTIVITY (src/debug)

```
<activity
    android:name=".DebugActivity"
    android:icon="@drawable/magnifying_glass"
    android:label="BDF - DEBUG"
    android:launchMode="singleTask">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```



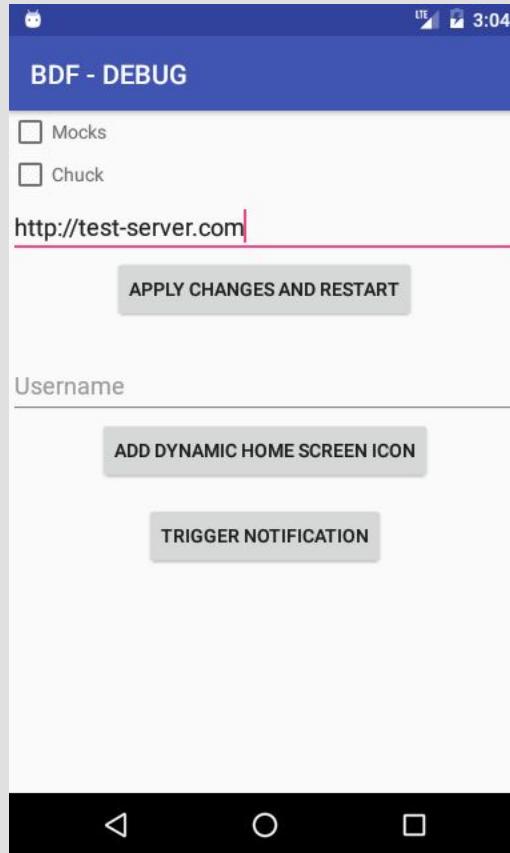


# OPTION 2: PERSISTENT NOTIFICATION



# **HOW WILL WE SAVE AND USE THE NEW ENDPOINT?**

# ALLOW USER TO EDIT ENDPOINT



# SAVE IN A PREFERENCE

```
public class DebugPreferences {  
    public static final String BASE_URL = "base_url";  
    private final SharedPreferences sharedPreferences;  
  
    public DebugPreferences(Context context) {  
        sharedPreferences = PreferenceManager  
            .getDefaultSharedPreferences(context.getApplicationContext());  
    }  
  
    public void setBaseUrl(String baseUrl) {  
        sharedPreferences.edit().putString(BASE_URL, baseUrl).apply();  
    }  
  
    public String getBaseUrl() { return sharedPreferences.getString(BASE_URL, "https://server"); }  
}
```

# FORCE KILL AND RESTART APP

Process Phoenix <https://github.com/JakeWharton/ProcessPhoenix>

Start the default activity in a new process:

```
ProcessPhoenix.triggerRebirth(context);
```

Or, if you want to launch with a specific Intent :

```
Intent nextIntent = //...
ProcessPhoenix.triggerRebirth(context, nextIntent);
```

To check if your application is inside the Phoenix process to skip initialization in onCreate :

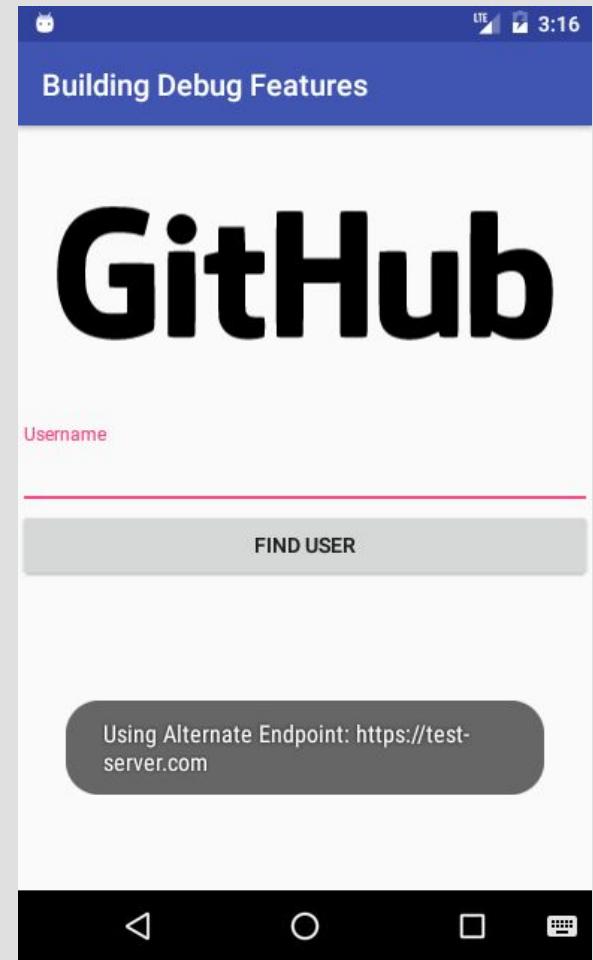
```
if (ProcessPhoenix.isPhoenixProcess(this)) {
    return;
}
```

# APPLY UPON APP INITIALIZATION

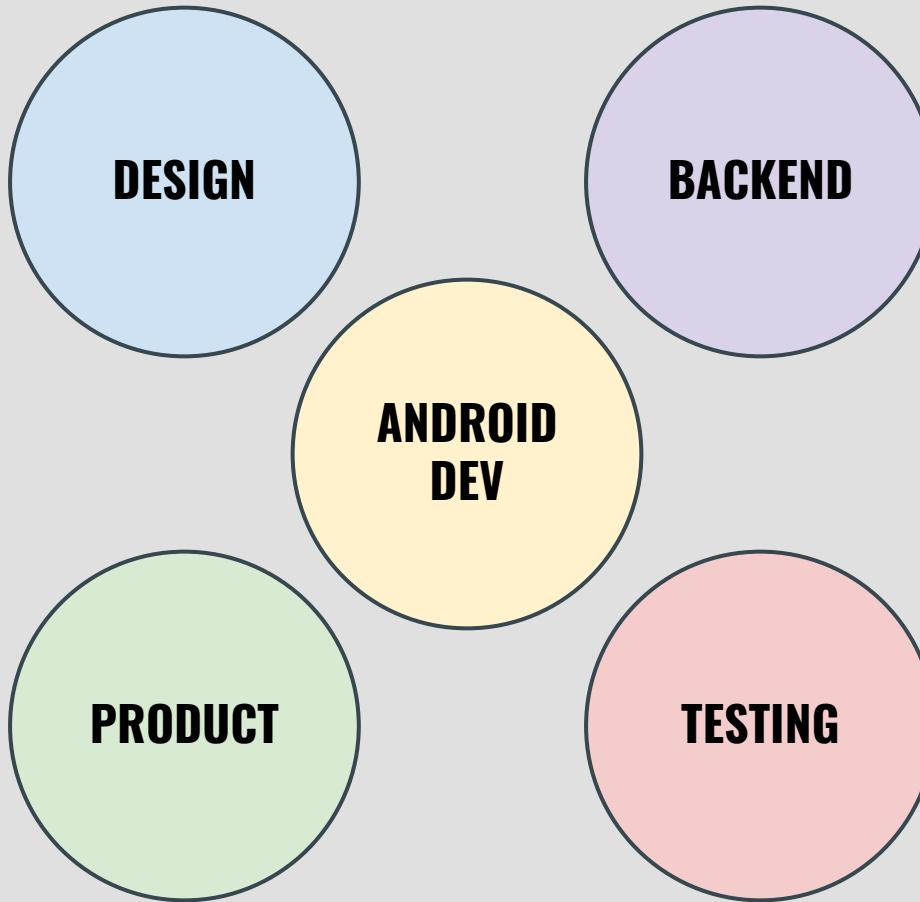
```
@Provides
Retrofit.Builder retrofitBuilder(DebugPreferences debugPreferences,
    OkHttpClient.Builder okHttpClientBuilder) {
    Retrofit.Builder builder = new Retrofit.Builder()
        .baseUrl(debugPreferences.getBaseUrl())
        .addConverterFactory(MoshiConverterFactory.create(new Moshi.Builder().build()))
        .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
        .client(okHttpClientBuilder.build());
    return builder;
}
```

# CONFIG CHANGE TOAST

```
new Handler(Looper.getMainLooper()).post(new Runnable() {
    @Override
    public void run() {
        Toast.makeText(MyApplication.this,
            "Using Alternate Endpoint: " + debugUrl, Toast.LENGTH_SHORT)
            .show();
    }
});
```



# WICKED AWESOME

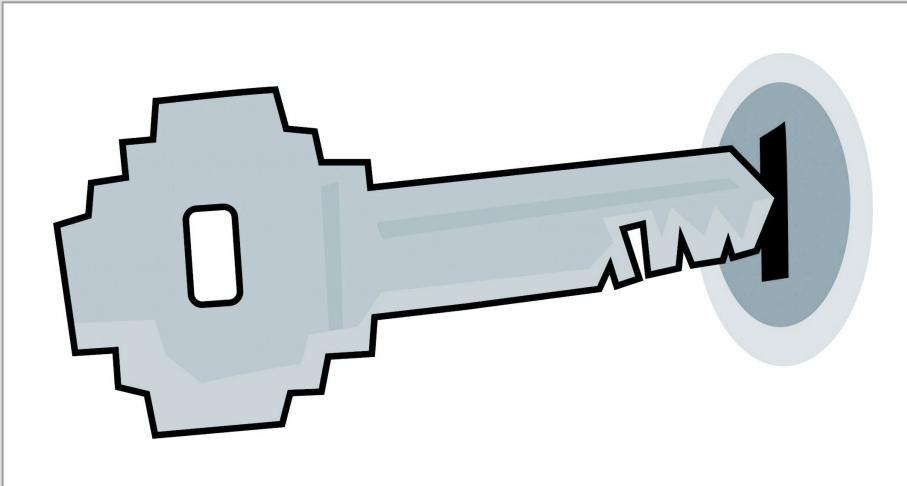




**PRODUCT**

**JEFF: “I NEED AN EASY WAY TO  
DEMO NOTIFICATIONS.”**

# CUSTOM DEBUG FEATURE “NOTIFICATION TESTING”



# TRIGGER A NOTIFICATION WITHOUT GOING THROUGH GCM?

# Broadcast Receivers via ADB

Broadcast Receiver ADB Commands:

```
adb shell "am broadcast -a PR_NOTIFICATION --es username handstandsam"
```

```
adb shell "am broadcast -a PR_NOTIFICATION --es username octocat"
```

# BROADCAST RECEIVER (src/debug/AndroidManifest.xml)

```
<receiver android:name=".DebugBroadcastReceiver">
    <intent-filter>
        <action android:name="PR_NOTIFICATION" />
    </intent-filter>
</receiver>
```

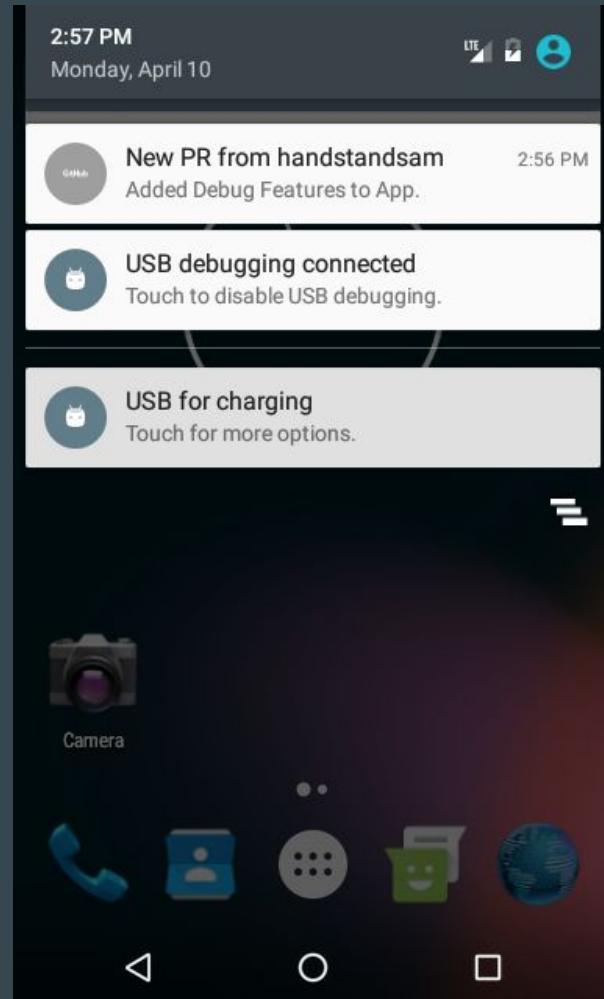
# BROADCAST RECEIVER

```
public class DebugBroadcastReceiver extends BroadcastReceiver {

    private static final String PR_NOTIFICATION = "PR_NOTIFICATION";

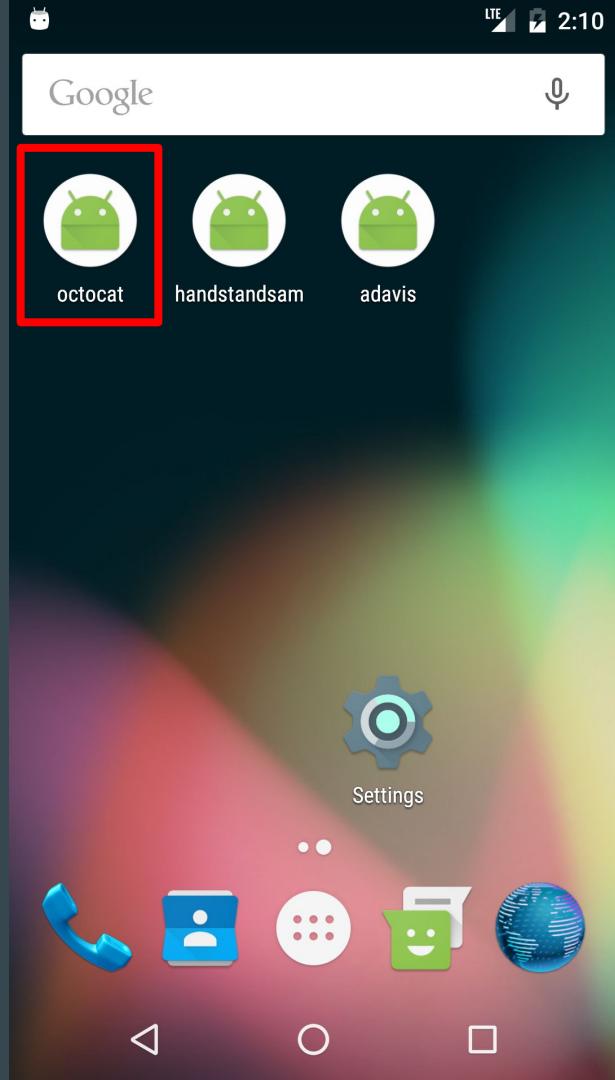
    @Override
    public void onReceive(Context context, Intent intent) {
        String intentAction = intent.getAction();
        if (intentAction == null || !PR_NOTIFICATION.equals(intentAction)) {
            return;
        }
        Timber.d("processing " + PR_NOTIFICATION);
        Bundle extras = intent.getExtras();
        if (extras == null) {
            return;
        }
        final String username = extras.getString("username");
        Timber.d("username " + username);

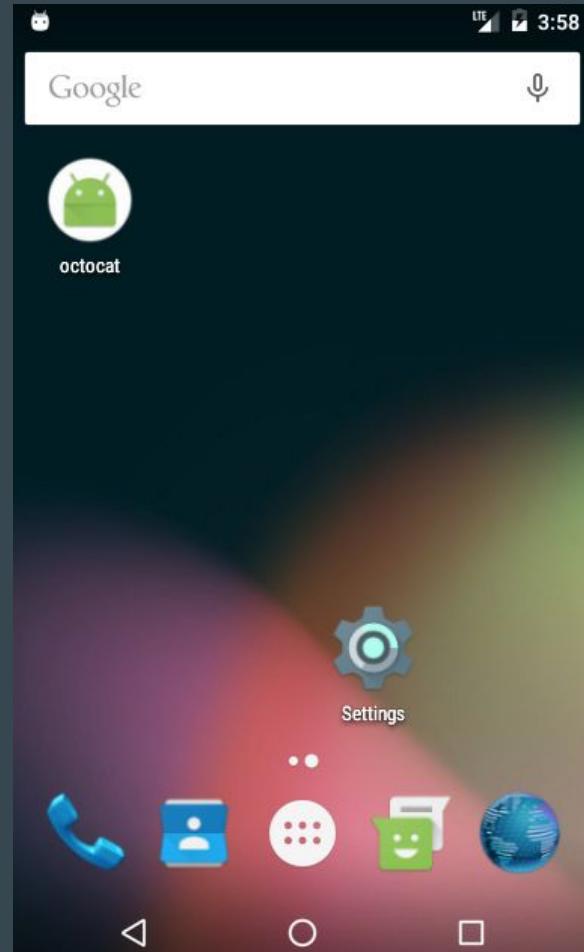
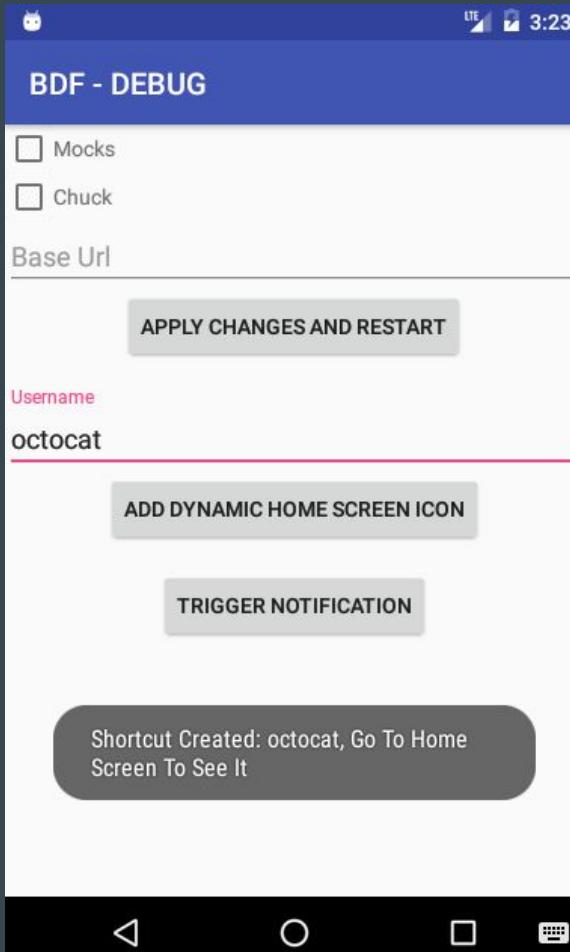
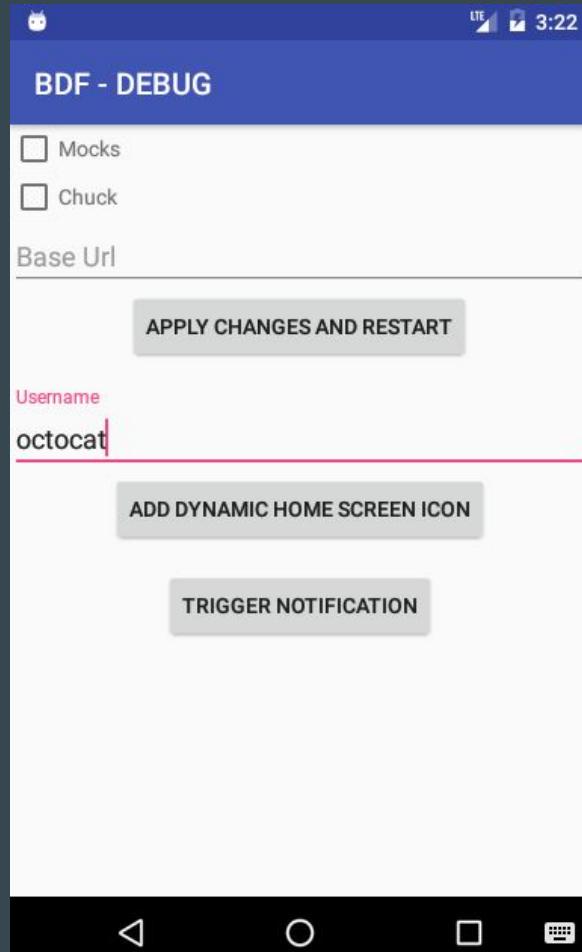
        IntentUtils.triggerPRNotification(context, username);
    }
}
```



# DYNAMIC HOME SCREEN SHORTCUTS

#droidconbos





# DYNAMIC HOME SCREEN SHORTCUTS

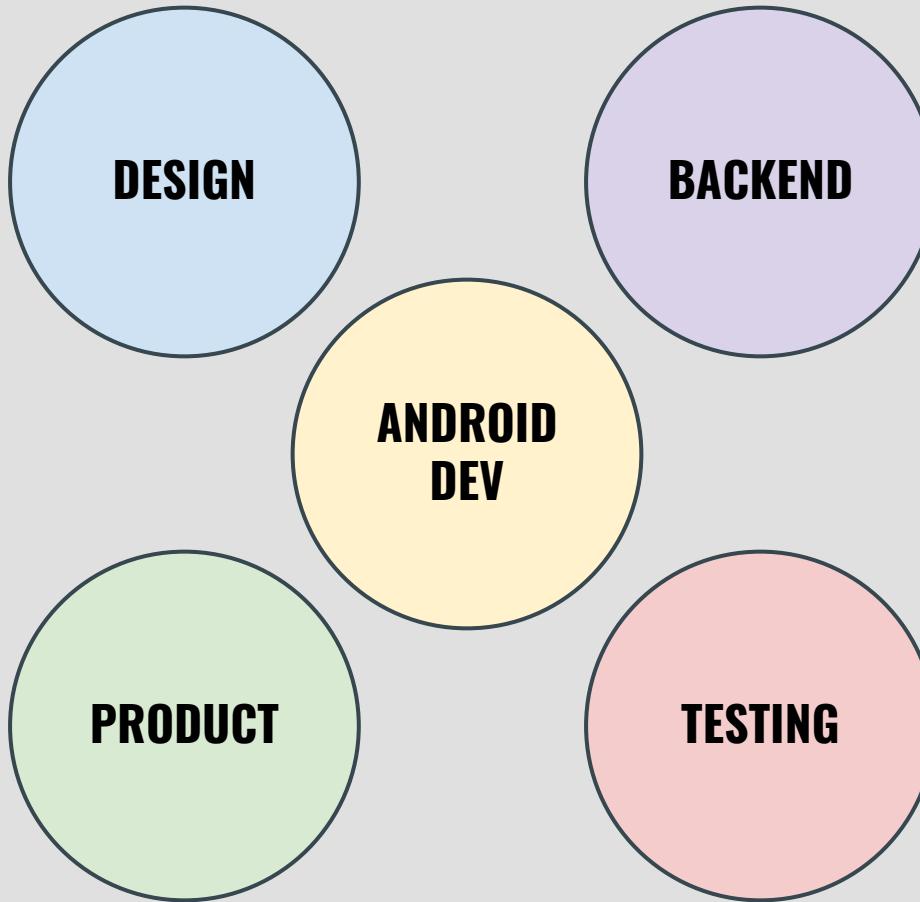
```
Intent addShortcutIntent = new Intent();
addShortcutIntent.putExtra(Intent.EXTRA_SHORTCUT_INTENT, deepLinkIntent);
addShortcutIntent.putExtra(Intent.EXTRA_SHORTCUT_NAME, "My Shortcut");
addShortcutIntent.putExtra(Intent.EXTRA_SHORTCUT_ICON_RESOURCE,
    Intent.ShortcutIconResource.fromContext(applicationContext, R.drawable.shortcut));
addShortcutIntent.setAction("com.android.launcher.action.INSTALL_SHORTCUT");
applicationContext.sendBroadcast(addShortcutIntent);
```

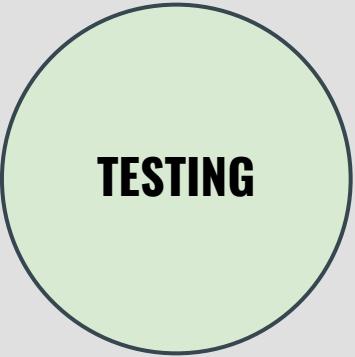
# DYNAMIC HOME SCREEN SHORTCUTS

```
Intent addShortcutIntent = new Intent();
addShortcutIntent.putExtra(Intent.EXTRA_SHORTCUT_INTENT, deepLinkIntent);
addShortcutIntent.putExtra(Intent.EXTRA_SHORTCUT_NAME, "My Shortcut");
addShortcutIntent.putExtra(Intent.EXTRA_SHORTCUT_ICON_RESOURCE,
    Intent.ShortcutIconResource.fromContext(applicationContext, R.drawable.shortcut));
addShortcutIntent.setAction("com.android.launcher.action.INSTALL_SHORTCUT");
applicationContext.sendBroadcast(addShortcutIntent);
```

```
<uses-permission
    android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
```

# WICKED AWESOME



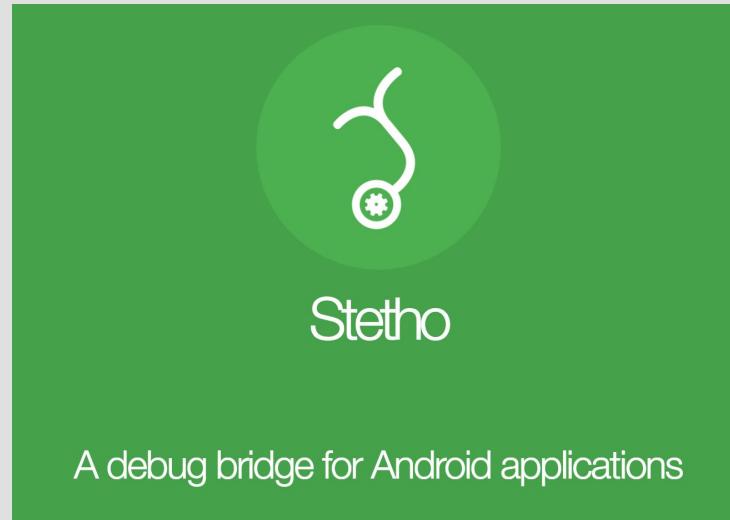


**TESTING**

**I WANT TO VALIDATE  
NETWORKING TRAFFIC IN A  
VISUAL WAY.**

# OPTION 1: STETHO

- CHROME DEVELOPER TOOLS
- NETWORK INSPECTION
- PREFERENCE INSPECTION & MODIFICATION
- DB INSPECTION & MODIFICATION
- VIEW HIERARCHY
- JAVASCRIPT CONSOLE
- DUMP APP





Chrome | chrome://inspect/#devices



Apps



Bookmarks

## DevTools

[Devices](#)[Pages](#)[Extensions](#)[Apps](#)[Shared workers](#)[Service workers](#)[Other](#)

## Devices

 Discover USB devices[Port forwarding...](#) Discover network targets[Configure...](#)[Open dedicated DevTools for Node](#)

Android SDK built for x86\_64 #EMULATOR-5554

**com.handstandsam.buildingdebugfeatures.debug (1.0)**

 BDF - DEBUG (powered by Stetho)  
[inspect](#)

Chrome File Edit View History Bookmarks People Window Help

Developer Tools -

Elements Network Sources Timeline Profiles Resources Audits Console

Styles Computed Event Listeners »

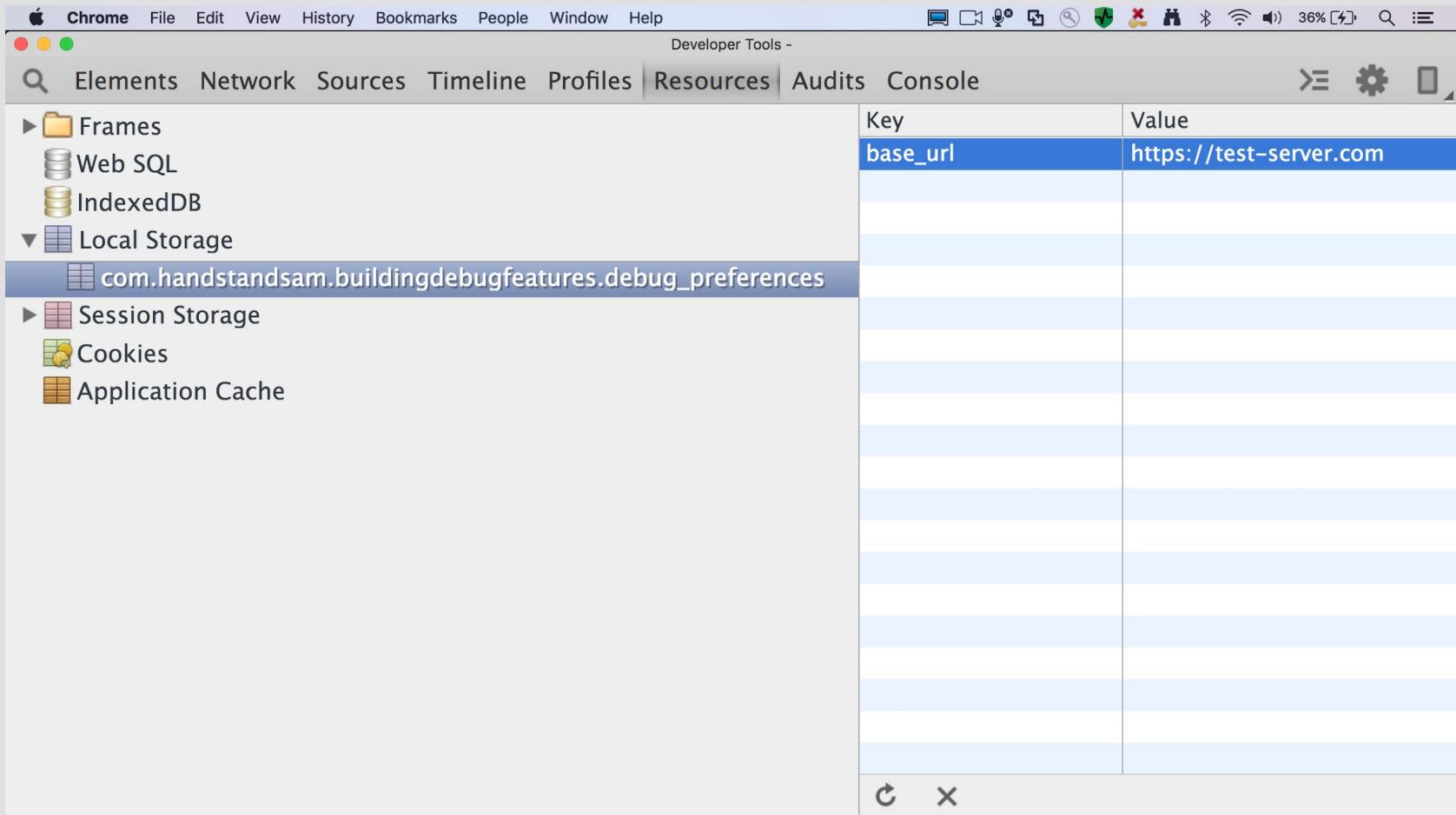
Accessibility Properties {  
    `ignored: false;`  
    `focuseable: true;`  
    `focuseable-reasons: View is actionable and has no children.;`  
    `focused: false;`  
    `actions: select, clear-selection, clear-focus, accessibility-focus, click, long-click, unknown;`  
}

<this\_element> {  
    `alpha: 1.0;`  
    `baseline: 30;`  
    `bottom: 46;`  
    `filter-touches-when-obscured: false;`  
    `fits-system-windows: false;`  
    `has-focus: true;`  
    `has-overlapping-rendering: true;`  
    `has-shadow: false;`  
    `has-transient-state: false;`  
    `height: 46;`  
    `id: @id/username;`  
    `important-for-accessibility: 1 (yes);`  
    `is-activated: false;`  
    `is-clickable: true;`  
    `is-drawing-cache-enabled: false;`  
    `is-enabled: true;`  
    `is-focusable: true;`  
    `is-focusable-in-touch-mode: true;`  
    `is-focused: true;`  
    `is-haptic-feedback-enabled: true;`  
    `is-hardware-accelerated: true;`  
    `is-hovered: false;`  
    `is-in-touch-mode: true;`  
    `is-layout-rtl: false;`  
    `is-opaque: false;`  
    `is-pressed: false;`  
    `is-selected: false;`  
    `is-sound-effects-enabled: true;`  
    `label-for: -1;`  
    `layer-type: 0 (NONE);`  
    `layout-direction: 0`

... framelayout android.support.v7.widget.appcompatedittext#@id/username

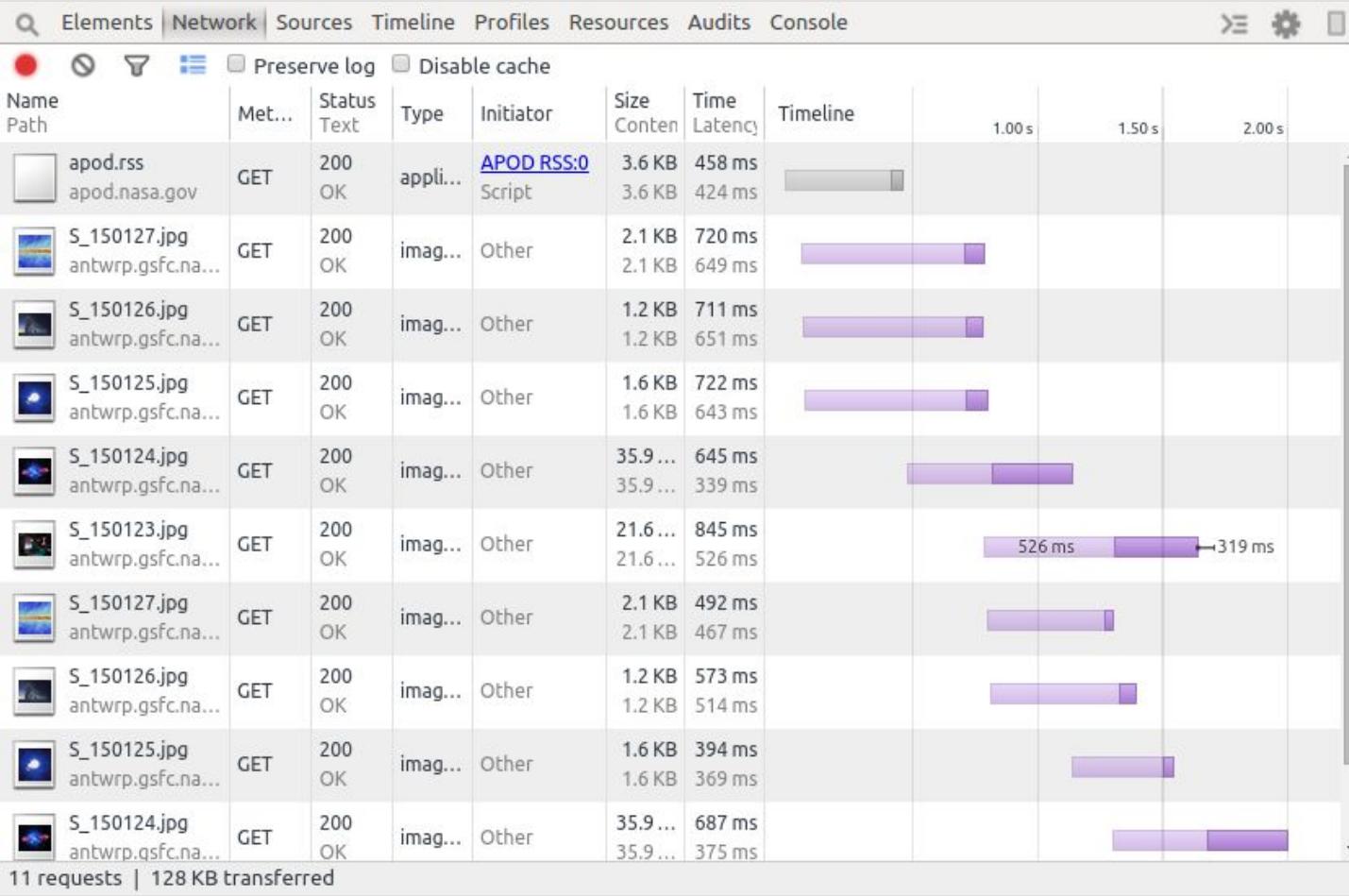
Find in Styles

The screenshot shows the Chrome Developer Tools open over a GitHub search interface. The left pane displays the page's content with a large 'GitHub' logo and a search bar. The right pane shows the DOM tree and the computed styles for the search input field. The styles panel lists numerous properties for the element, including accessibility states like 'ignored' and 'focuseable', and visual properties like 'alpha', 'baseline', and 'bottom'. The 'Computed' tab is selected, showing the final style values applied to the element.



#droidconbos

@HandstandSam



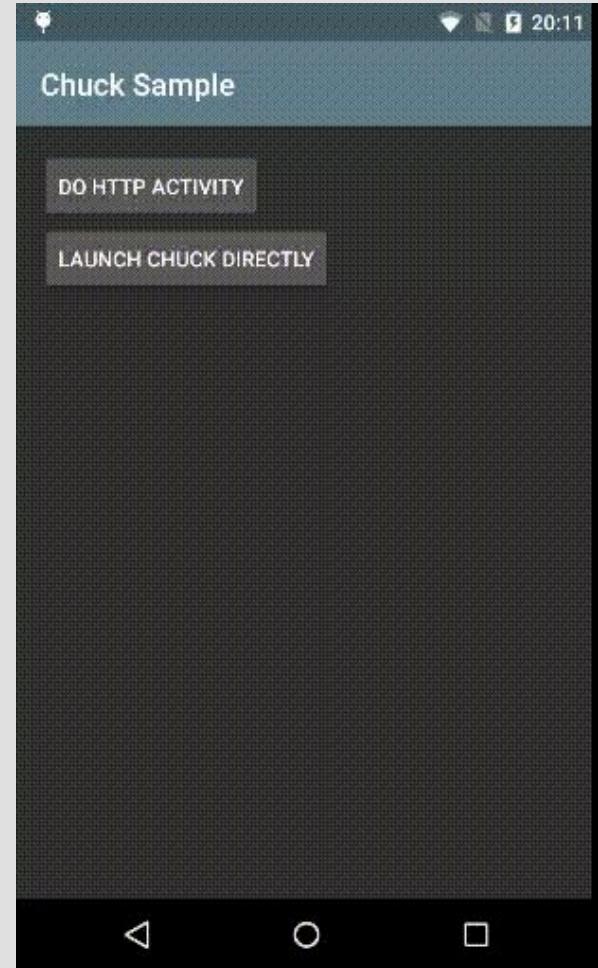
**“I’M NOT GOING TO BE  
CONNECTED TO A COMPUTER.”**

# OPTION 2: CHUCK

<https://github.com/jgilfelt/chuck>

```
dependencies {  
    debugCompile 'com.readystatesoftware.chuck:library:1.0.4'  
    releaseCompile 'com.readystatesoftware.chuck:library-no-op:1.0.4'  
}
```

```
OkHttpClient client = new OkHttpClient.Builder()  
.addInterceptor(new ChuckInterceptor(context))  
.build();
```



**SURE, GIVE ME A FEW MINUTES,  
I'LL ADD IT TO ALL FUTURE DEBUG BUILDS.**

 Gradle Build Running



# WICKED AWESOME

# OPEN SOURCE INSPIRATION

DON'T RE-INVENT THE WHEEL

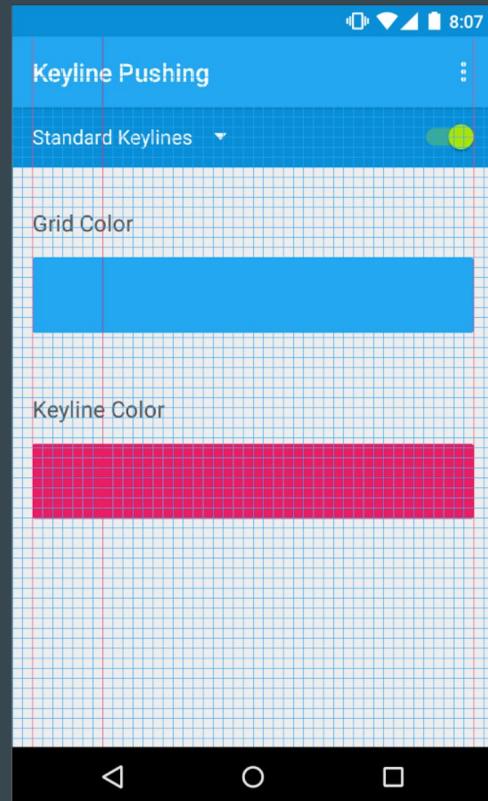
LEVERAGE EXISTING  
OPEN SOURCE LIBRARIES

# UI DESIGN & VIEW INSPECTION

# Keyline Pushing

<https://play.google.com/store/apps/details?id=com.faizmalkani.keylines>

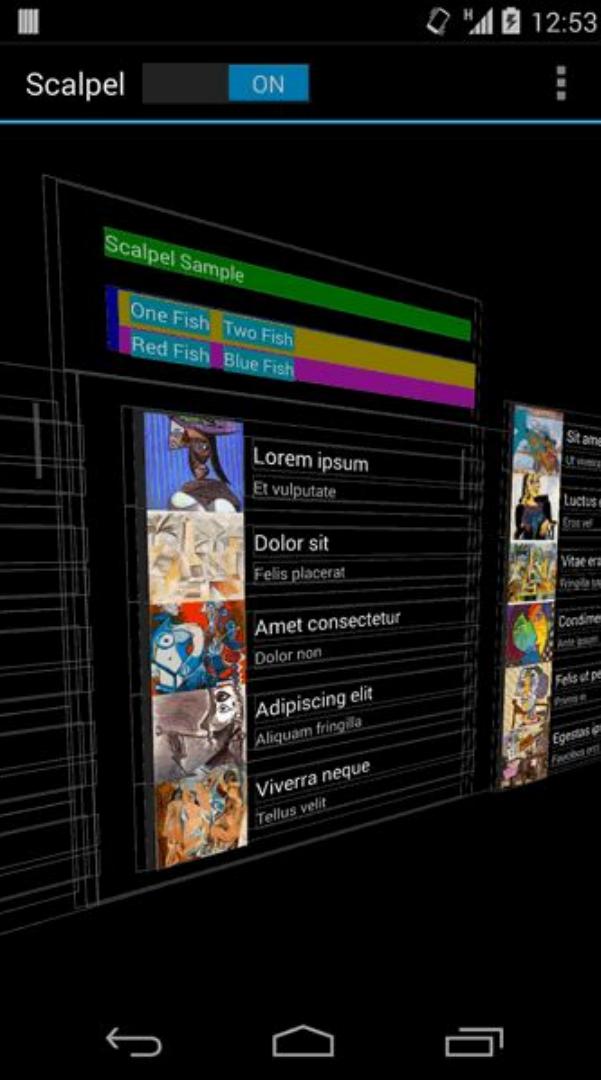
- Overlays Material Design Keylines



# Scalpel

<https://github.com/JakeWharton/scalpel>

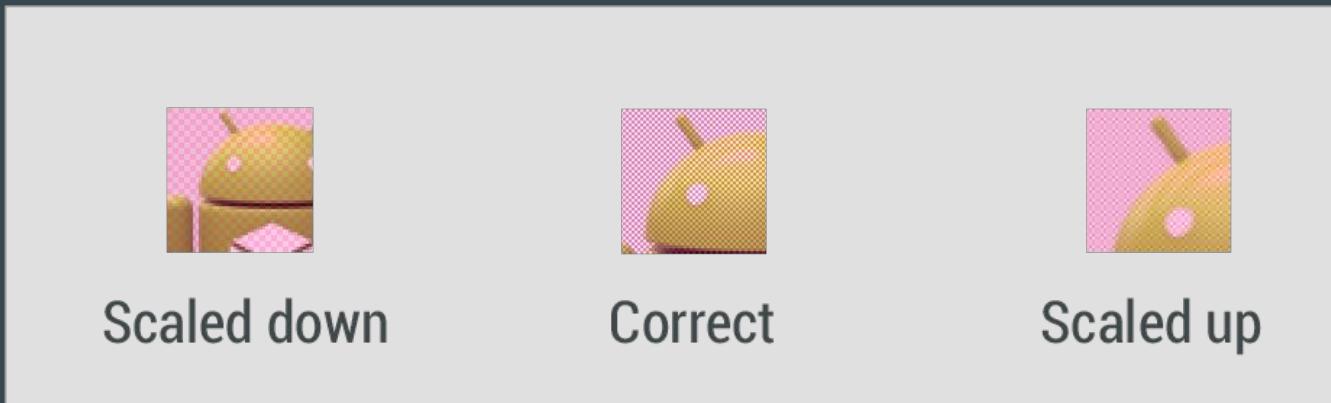
- A surgical debugging tool to uncover the layers under your app.



# Madge

<https://github.com/JakeWharton/madge>

- Overlays for debugging whether assets are drawing at their native resolution.



# Logging

# Logcat

Android Monitor

Emulator Nexus\_5\_API\_23 Android 6.0, API 23 ▾ No Debuggable Processes

Verbose ▾ 🔎 RegEx No Filters ▾

logcat Monitors ↗

04-07 17:16:10.955 1274-1945/? I/WvCam: CamEngine::closeSession  
04-07 17:16:11.079 1597-2011/system\_process I/AccountManagerService: getTypesVisibleToCaller: isPermitted? true  
04-07 17:16:11.133 2641-2641/? W/Binder\_1: type=1400 audit(0.0:12): avc: denied { search } for name="1277" dev="proc" ino=5304 scontext=u:r:u  
04-07 17:16:11.143 2641-2641/? W/Binder\_1: type=1400 audit(0.0:13): avc: denied { read } for name="/" dev="tmpfs" ino=3073 scontext=u:r:u  
04-07 17:16:14.057 2362-2628/com.google.android.gms I/CheckinService: Done disabling old GoogleServicesFramework version  
04-07 17:16:14.204 1597-1638/system\_process I/InputReader: Reconfiguring input devices. changes=0x00000010  
04-07 17:16:14.232 2362-2989/com.google.android.gms D/Wear\_Controller: Received broadcast action=android.intent.action.PACKAGE\_CHANGED an  
04-07 17:16:14.279 2106-3110/com.google.android.googlequicksearchbox:search I/UpdateIcingCorporaServi: Updating corpora: APPS=com.google.  
04-07 17:16:14.299 2106-3110/com.google.android.googlequicksearchbox:search I/UpdateIcingCorporaServi: UpdateCorporaTask done [took 20 ms  
04-07 17:16:14.355 2851-2851/com.google.android.gms.persistent I/GCoreNlp: shouldConfirmNlp, NLP off. Ensuring opt-in disabled  
04-07 17:16:14.452 2362-2722/com.google.android.gms I/Icing: updateResources: need to parse snt{com.google.android.gms}  
04-07 17:16:14.452 2362-3036/com.google.android.gms W/IcingInternalCorpora: getNumBytesRead when not calculated.  
04-07 17:16:14.562 2362-2722/com.google.android.gms I/Icing: Usage reports 0 indexed 0 rejected 0 imm upload false  
04-07 17:16:14.567 2362-2722/com.google.android.gms I/Icing: Usage reports 0 indexed 0 rejected 0 imm upload false  
04-07 17:16:15.591 2362-2680/com.google.android.gms I/Icing: Indexing DD2D63678CDBEABAAAF66167C9B94A797A910F25 from com.google.android.gm  
04-07 17:16:15.591 2362-2680/com.google.android.gms I/Icing: Indexing done DD2D63678CDBEABAAAF66167C9B94A797A910F25  
04-07 17:16:15.999 2362-2975/com.google.android.gms W/PlatformStatsUtil: Could not retrieve Usage & Diagnostics setting. Giving up.  
04-07 17:16:16.022 1597-2010/system\_process I/AccountManagerService: getTypesVisibleToCaller: isPermitted? true  
04-07 17:16:17.084 1597-1761/system\_process I/ActivityManager: Killing 2493:com.android.keychain/1000 (adj 15): empty #17

Find Run TODO Version Control Terminal Messages Event Log Gradle Console

#droidconbos @HandstandSam

# Timber

<https://github.com/JakeWharton/timber>

- Abstraction on top of Android Log
  - `Log.d(TAG, message);`
- More concise syntax
  - `Timber.d(message);`
  - `d(message);` - Static Import
- Ability to toggle logging on/off.
- Ability to keep logs in memory.



# Hugo <https://github.com/JakeWharton/hugo>

- Annotation-triggered method call logging for your debug builds.

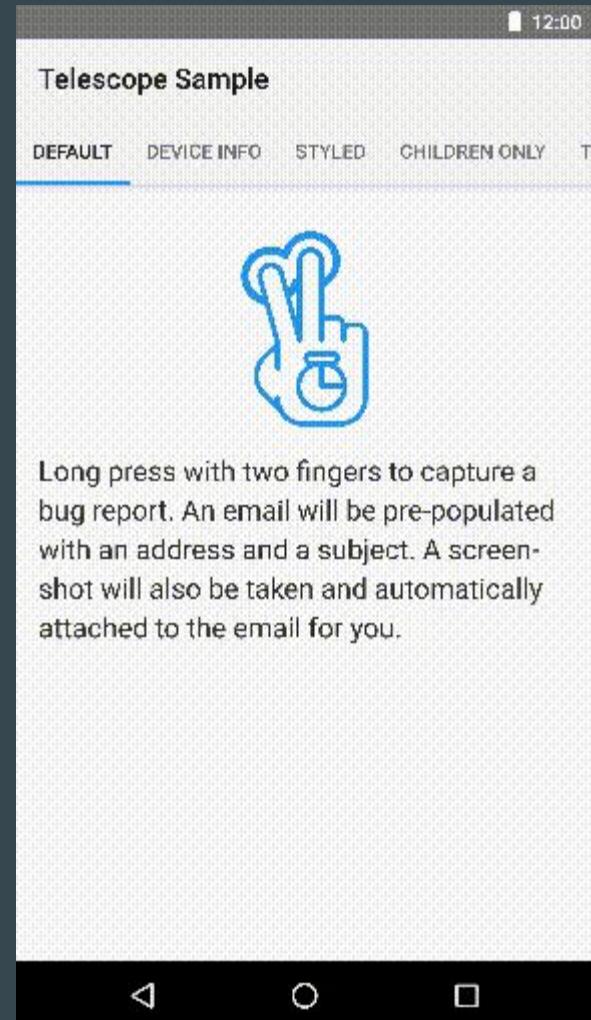
```
@DebugLog  
public String getName(String first, String last) {  
    SystemClock.sleep(15); // Don't ever really do this!  
    return first + " " + last;  
}
```

```
V/Example: → getName(first="Jake", last="Wharton")  
V/Example: ← getName [16ms] = "Jake Wharton"
```

# Bug Reporting

# Telescope <https://github.com/mattprecious/telescope>

- A simple tool to allow easy bug report capturing within your app.

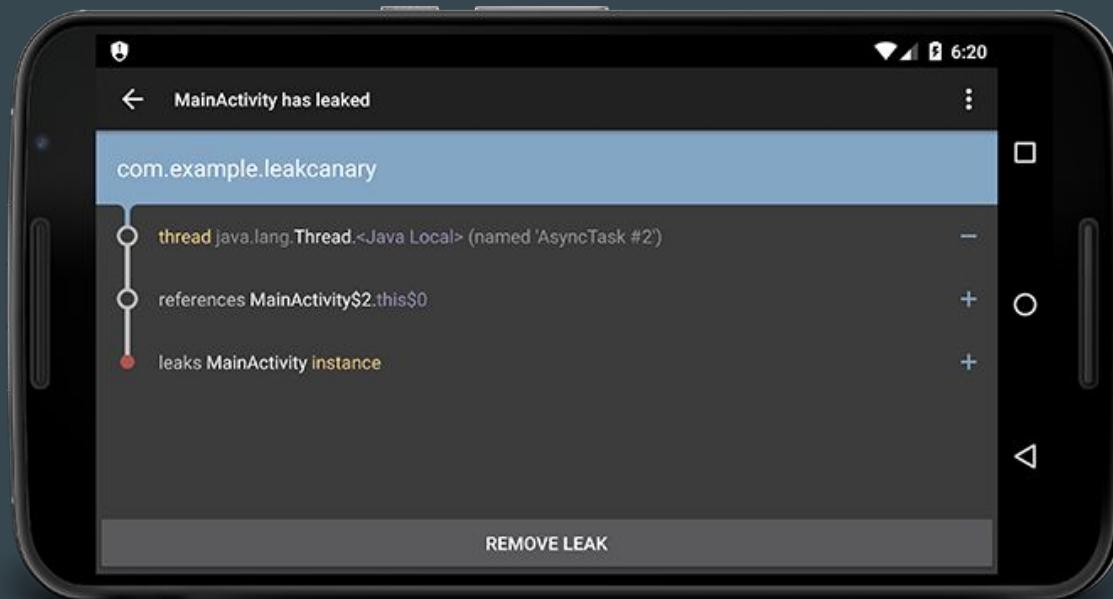


# Memory Leaks

# Leak Canary

<https://github.com/square/leakcanary>

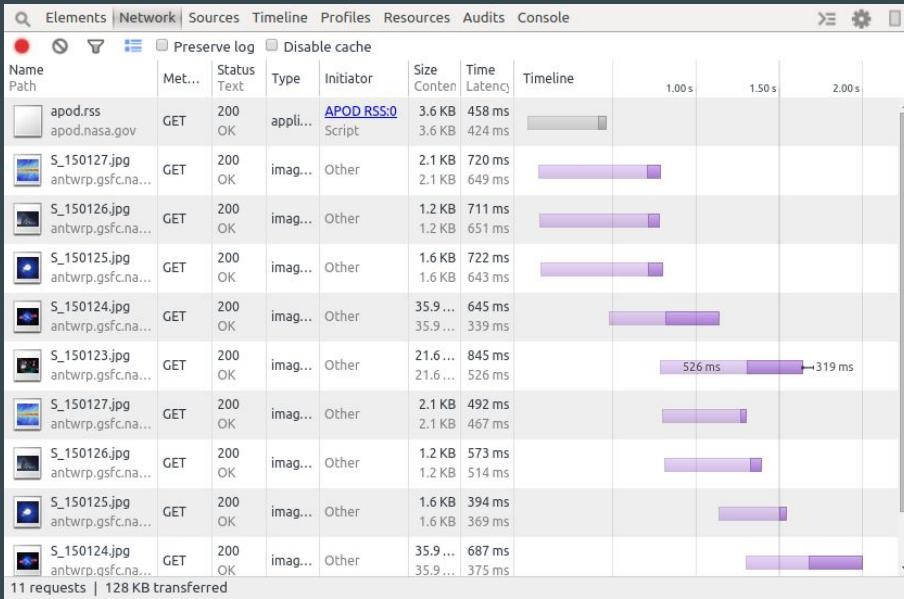
- A memory leak detection library for Android and Java.



# Network Inspection

# Network Debugging

- Stetho
- Chuck



# DON'T SHIP TO PROD

# APK ANALYZER

app-debug.apk - BuildingDebugFeatures - [/development/BuildingDebugFeatures]

BuildingDebugFeatures app build outputs apk app-debug.apk

# APK ANALYZER - DEBUG BUILD

1: Project app-debug.apk x app-release.apk x

com.handstandsam.buildingdebugfeatures.debug (version 1.0)

Raw File Size: 4.4 MB, Download Size: 4 MB Compare with...

File	Raw File Size	Download Size	% of Total Download
classes.dex	8.5 MB	3.3 MB	89%
res	452.4 KB	308.1 KB	8%
resources.arsc	268.5 KB	59.1 KB	1.5%
META-INF	108.9 KB	33.5 KB	0.9%
org	90.3 KB	12.5 KB	0.3%

This dex file defines 8182 classes with 52030 methods, and references 60954 methods.

Class	Defined Methods	Referenced Methods
handstandsam	186	186
buildingdebugfeatures	186	186
di	55	55
debug	15	15
DebugPreferences	5	5
HomeScreenShortcutBuilder	5	5
DebugDispatcher	3	3
HomeScreenShortcutBuilder\$1	2	2
MainActivity	13	13

Android Model

TODO 6: Android Monitor Terminal 9: Version Control Event Log Gradle Console

n/a n/a Git: master Context: <no context>

app-release.apk - BuildingDebugFeatures - [/development/BuildingDebugFeatures]

BuildingDebugFeatures app build outputs apk app-release.apk

# APK ANALYZER - RELEASE BUILD

Project: app-debug.apk x app-release.apk x

com.handstandsam.buildingdebugfeatures (version 1.0)

i Raw File Size: 2.7 MB, Download Size: 2.4 MB Compare with...

File	Raw File Size	Download Size	% of Total Download
classes.dex	5.2 MB	2.1 MB	86.1%
res	388 KB	259.9 KB	10.4%
resources.arsc	250.6 KB	56.3 KB	2.3%
META-INF	98.5 KB	29.9 KB	1.2%
AndroidManifest.xml	2.7 KB	915 B	0%

This dex file defines 4658 classes with 32342 methods, and references 39337 methods.

Class	Defined Methods	Referenced Methods
handstandsam	135	135
buildingdebugfeatures	135	135
di	53	53
MainActivity	13	13
UserActivity	9	9
MainActivity\$2	6	6
UserActivity_MembersInjector	6	6
MainActivity_MembersInjector	6	6
network	4	4

# CLASSY SHARK

<https://github.com/google/android-classyshark>

Main

app-debug.apk

AndroidManifest.xml

Classes Methods count

app-debug.apk  
AndroidManifest.xml  
classes  
res

# CLASSY SHARK - DEBUG BUILD

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest
    versionCode='1'
    versionName='1.0'
    package='com.handstandsam.buildingdebugfeatures.debug'
    platformBuildVersionCode='25'
    platformBuildVersionName='7.1.1'>
    <uses-sdk
        minSdkVersion='16'
        targetSdkVersion='25'>
    </uses-sdk>
    <uses-permission
        name='com.android.launcher.permission.INSTALL_SHORTCUT'>
    </uses-permission>
    <uses-permission
        name='android.permission.INTERNET'>
    </uses-permission>
    <application
        theme='@res/0x7F0900A8'
        label='@res/0x7F070040'
        icon='@res/0x7F030000'
        name='com.handstandsam.buildingdebugfeatures.MyApplication'
        debuggable='true'
        screenOrientation='1'
        allowBackup='true'
        supportsRtl='true'
        roundIcon='@res/0x7F030001'>
        <activity
            android_name='MainActivity'>
    </activity>
</application>
</manifest>
```

Main

app-release.apk

AndroidManifest.xml

Classes Methods count

app-release.apk

AndroidManifest.xml

classes

res

# CLASSY SHARK - RELEASE BUILD

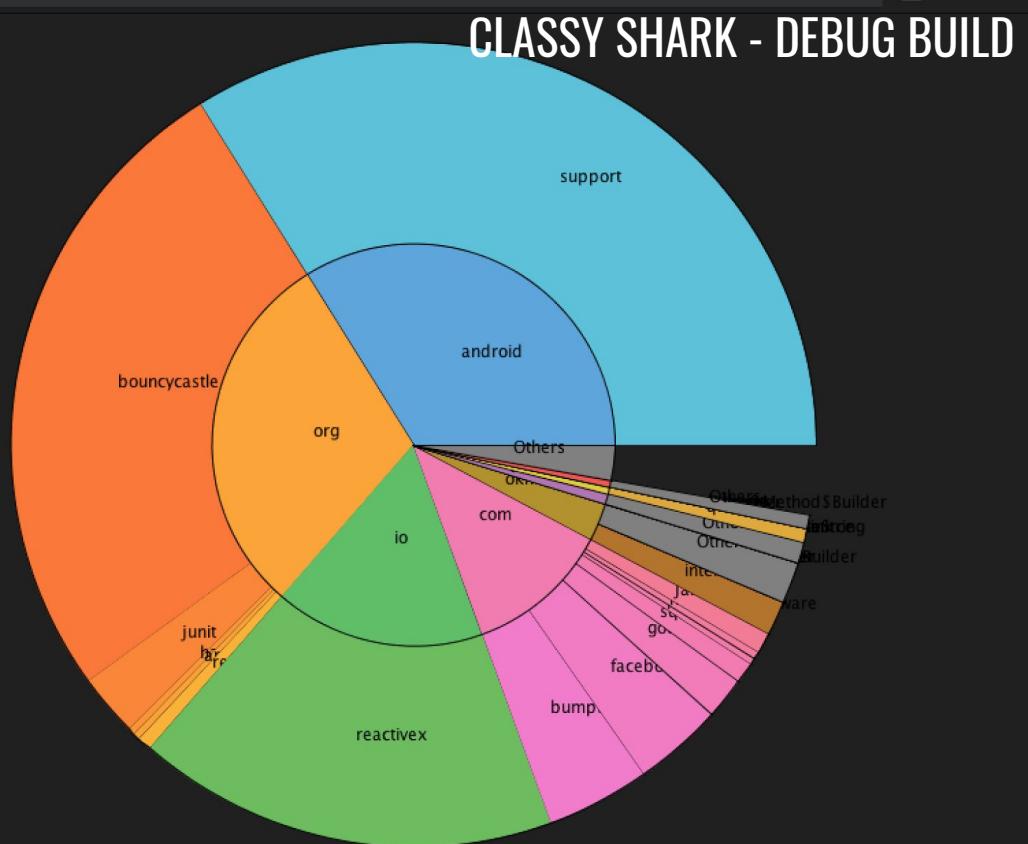
```
<?xml version="1.0" encoding="UTF-8"?>
<manifest
    versionCode='1'
    versionName='1.0'
    package='com.handstandsam.buildingdebugfeatures'
    platformBuildVersionCode='25'
    platformBuildVersionName='7.1.1'>
    <uses-sdk
        minSdkVersion='16'
        targetSdkVersion='25'>
    </uses-sdk>
    <uses-permission
        name='android.permission.INTERNET'>
    </uses-permission>
    <application
        theme='@res/0x7F0900A6'
        label='@res/0x7F07002A'
        icon='@res/0x7F030000'
        name='com.handstandsam.buildingdebugfeatures.MyApplication'
        screenOrientation='1'
        allowBackup='true'
        supportsRtl='true'
        roundIcon='@res/0x7F030001'>
        <activity
            name='com.handstandsam.buildingdebugfeatures.MainActivity'
            launchMode='2'>
            <intent-filter>
                <action
                    name='android.intent.action.MAIN'>
```

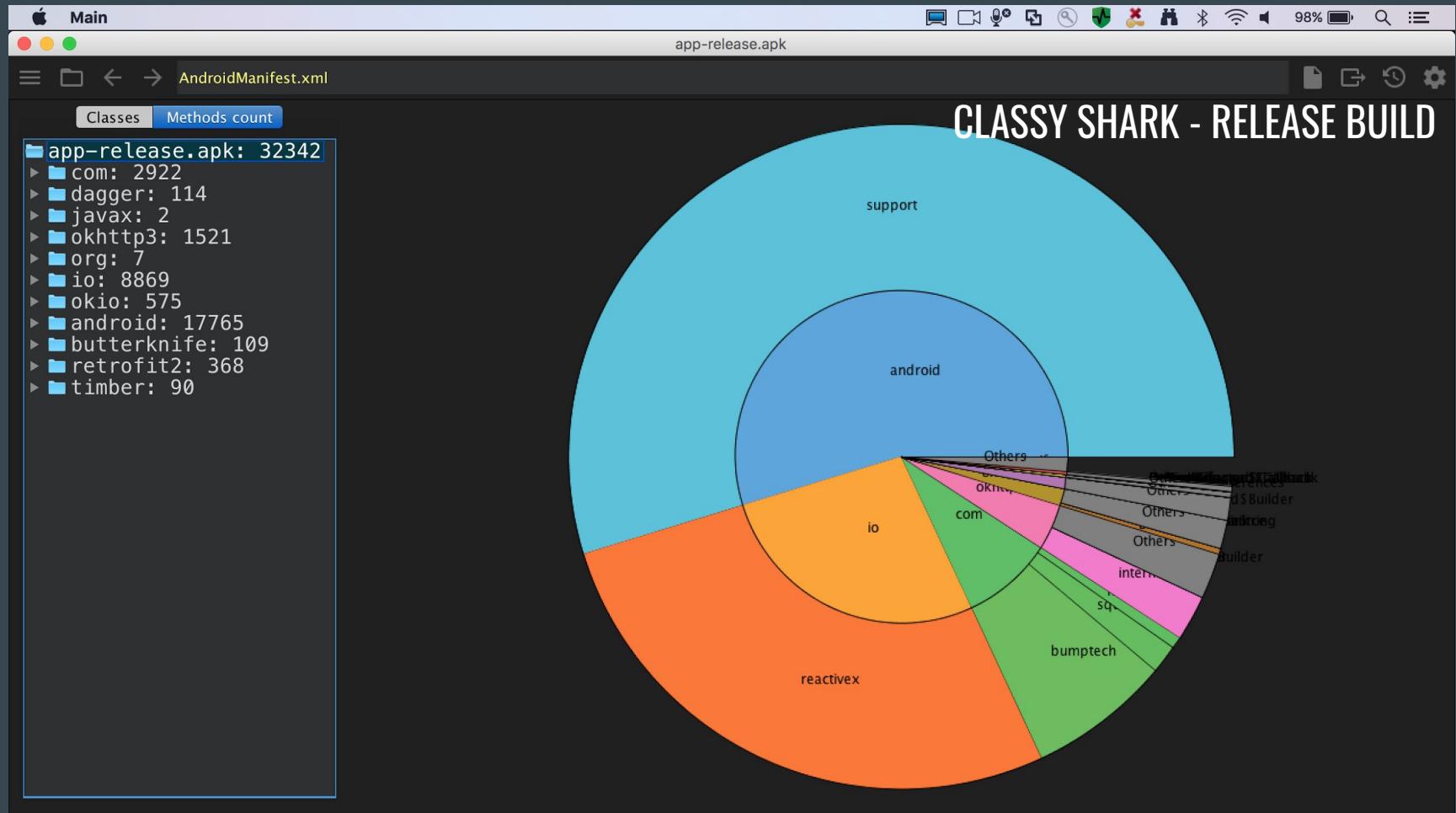
≡ ⌂ ⌄ ⌅ ⌆ AndroidManifest.xml



Classes Methods count

app-debug.apk: 52030  
  com: 6189  
  dagger: 114  
  junit: 296  
  org: 15523  
  io: 8869  
  android: 17765  
  retrofit2: 368  
  javax: 32  
  okhttp3: 1719  
  okio: 575  
  butterknife: 109  
  timber: 90  
  nl: 381





#droidconbos

@HandstandSam

# RECAP

- USE GRADLE BUILD VARIANTS
- USE EXISTING LIBRARIES IF THEY WORK
- START SIMPLE & BE PRAGMATIC

**IN MY OPINION:**  
DEBUG FEATURES THAT ENABLE YOUR TEAM IS JUST AS  
VALUABLE AS WHAT GETS SHIPPED TO PROD.

# THINK OUTSIDE THE BOX

# THE END