

Discover Support Library

Masahiko Adachi

@adamrokcer / +Masahiko.Adachi

28th Sep, 2013

About me

- 足立昌彦 (@adamrocker / +Masahiko.Adachi)
 - バイドウ株式会社 技術顧問 (Simeji)
 - 株式会社カブク Co-Founder
 - Google Developer Expert (Android)



便利なライブラリを紹介

- Support Libraryの一部
- Google Under Groundなライブラリ

Support Libraryとは？



なぜSupport Library？

- AndroidのSDKはどんどん進化している
 - 今やr22.2.1
 - 古いSDKを救うのがSupport Library
 - バックポートすべきはどんなクラス？
 - 必要 or 便利なクラス
- つまり、効率よく新SDKをチェックできる

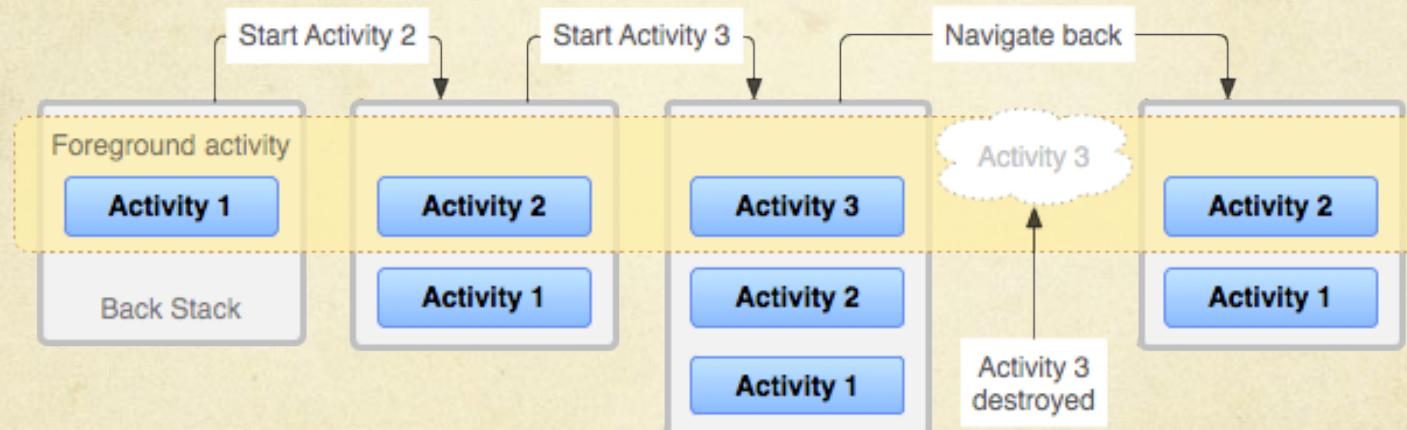
Support Libraryの種類

v4 (API Level 4)	v7 (API Level 7)	v8 (API Level 8)	v13 (API Level 13)
- Android1.6 -	- Android2.1.x -	- Android2.2.x -	- Android3.2 -
15 packages 104 classes	7 packages 42 classes	1 package 51 classes	1 package 4 classes

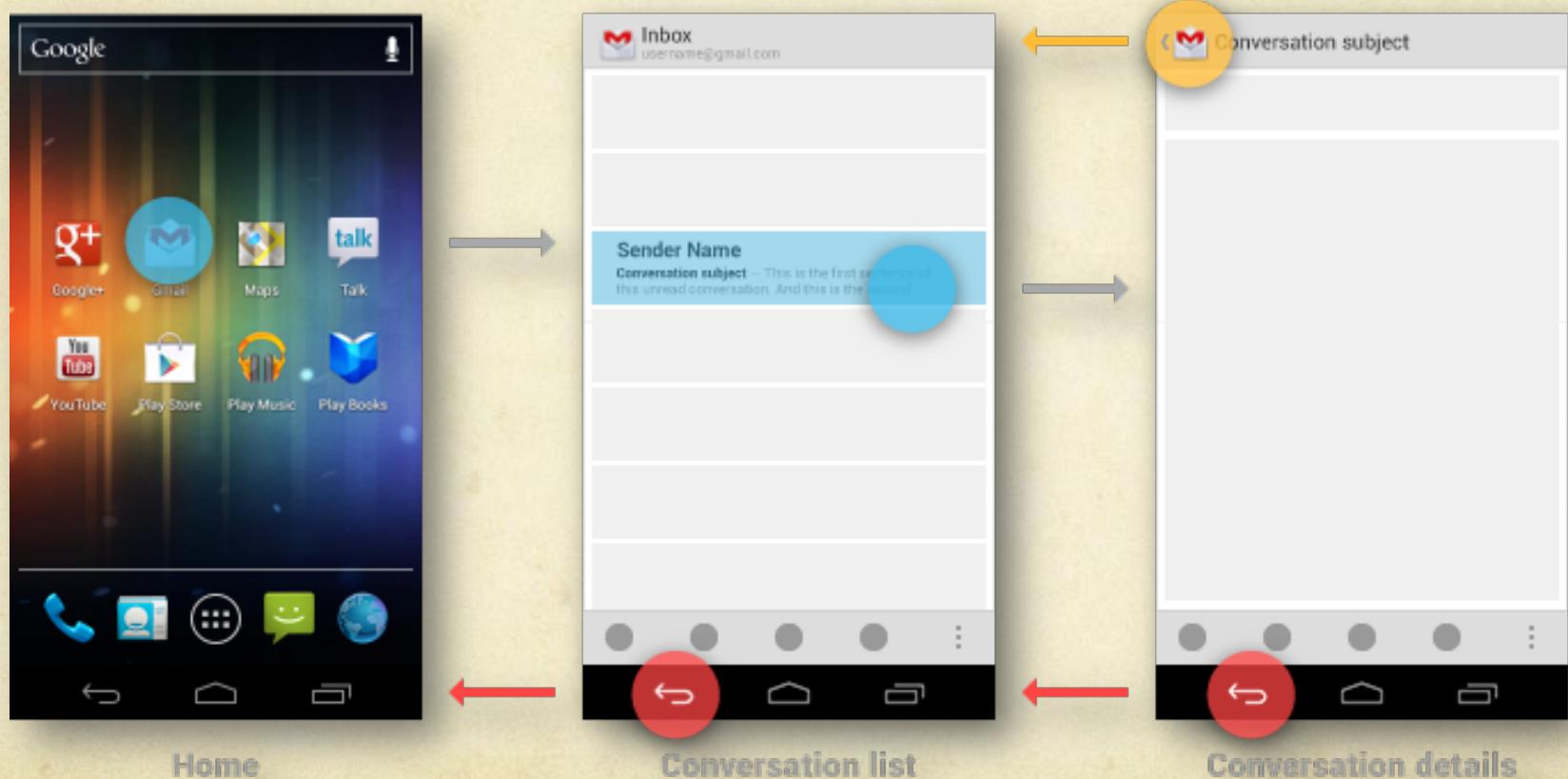
android-support-v4.jar

NavUtils と TaskStackBuilder

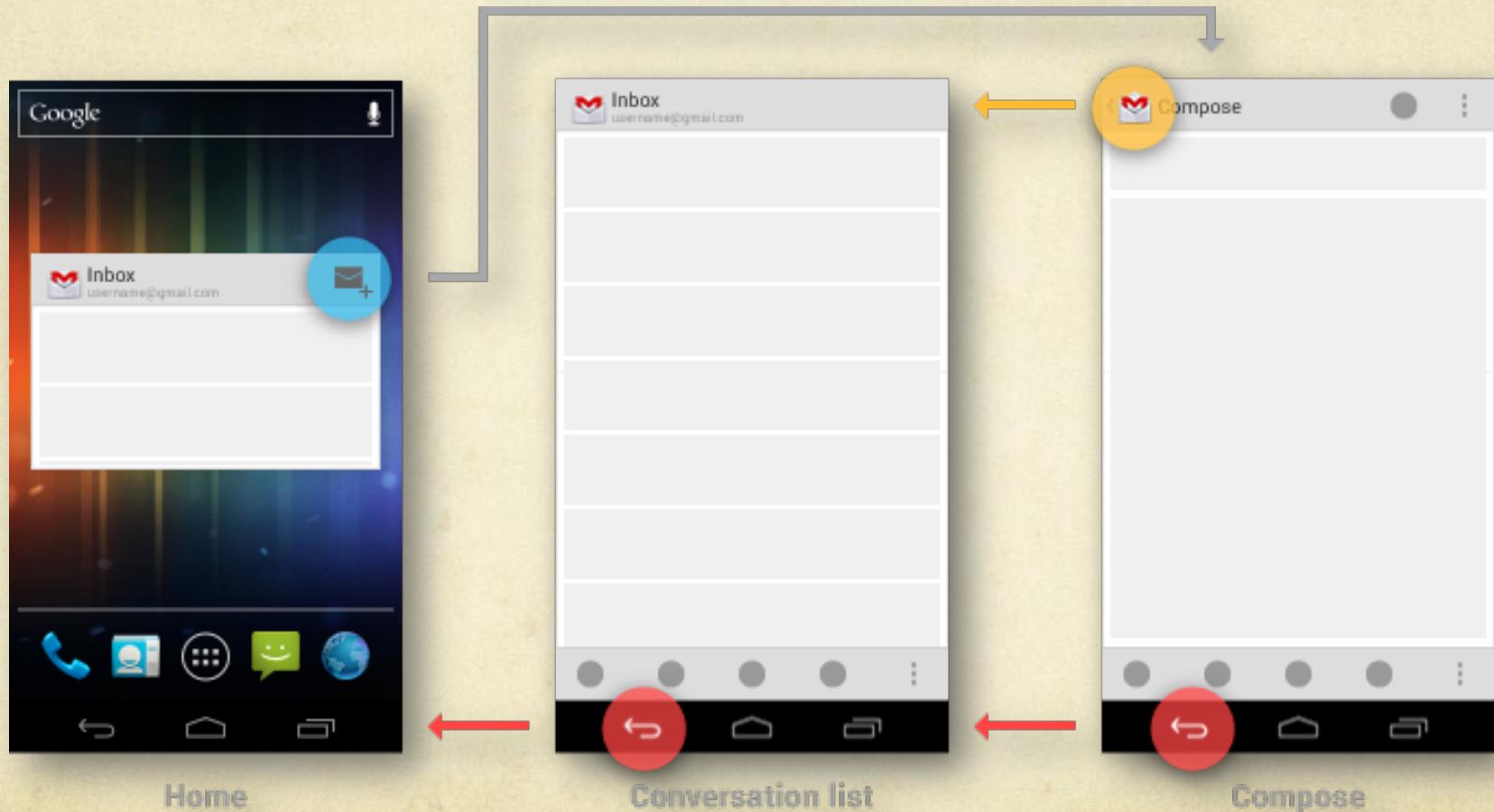
- TaskのBack Stackを制御する
- Androidのページ遷移をデザインする



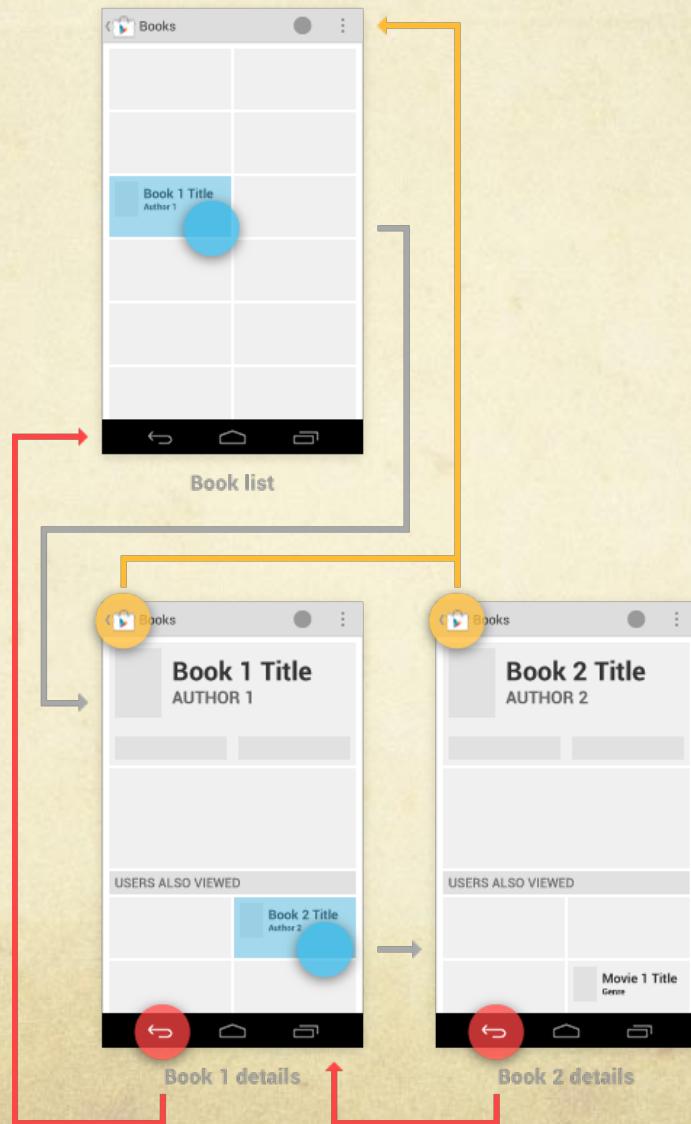
Androidのページ遷移



特殊なBack Stack



特殊なページ遷移



NavUtils と TaskStackBuilder

NavUtils#navigateUpTo(srcActivity, upIntent)

Back Stack内で指定したActivityを起動

TaskStackbuilder#addNextIntent(nextIntent)

TaskStackBuilder#startActivities(bundle)

addNextIntent()メソッドでBack Stackを構築して

startActivities()メソッドでTopのActivityを起動

AsyncTaskLoader

- 非同期でデータがロードできる
- Android3.0と同じFrameworkをわざわざSupport Libraryに実装
- AsyncTaskとの違いはコンテンツの更新管理機能

AsyncTaskLoader

```
public Loader<Cursor> onCreateLoader(int id, Bundle args) {  
    return new CursorLoader(this, Contract.TABLE1.contentUri, null, null, null, null);  
}
```

コンテンツが更新されると
自動で呼び出される

```
public void onLoadFinished(Loader<Cursor> loader, Cursor c) {  
    c.moveToFirst();  
    do {  
        for (int i = 0; i < c.getColumnCount(); i++) {  
            String v = c.getString(i);  
        }  
    } while (c.moveToNext());  
}
```

LocalBroadcastManager

- アプリ内で閉じたBroadcast Intentが発行できる
- 他のアプリがデータを受け取ることがないのでActivityを疎結合させたままセキュアなデータ通信ができる

```
LocalBroadcastManager.getInstance(context).sendBroadcast(intent);
```

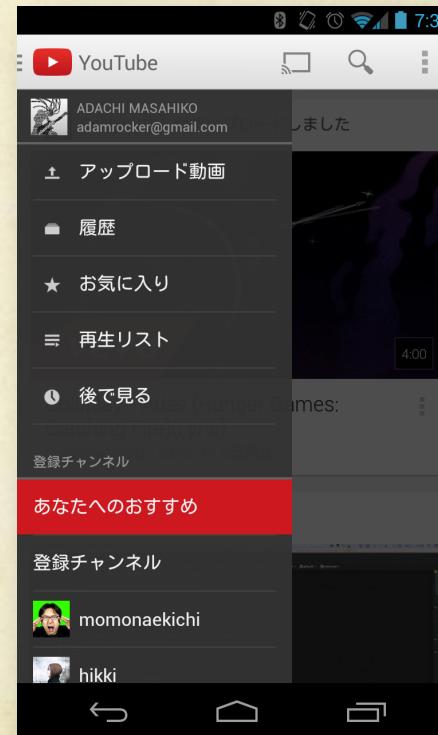
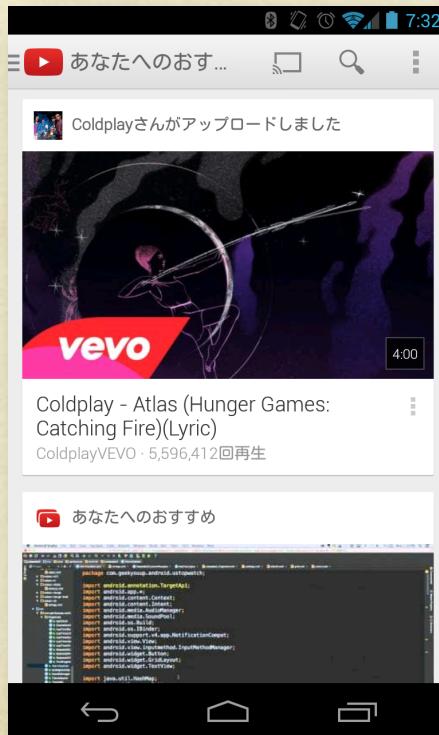
SparseArray

- API Level 1 から存在するSparseArrayの拡張
- LongSparseArrayなども使える
- SQLiteの_IDがLongなので、 キャッシュに効果大
- Hashmapのように使えるが性能は絶大に良い。
Key値がPrimitiveなのでオブジェクトを作らずメモリ効率も高いので絶必！

```
SparseArray<String> map = new SparseArray<String>();  
map.put( 11, "eleven" );  
map.delete( 11 );
```

DrawerLayout

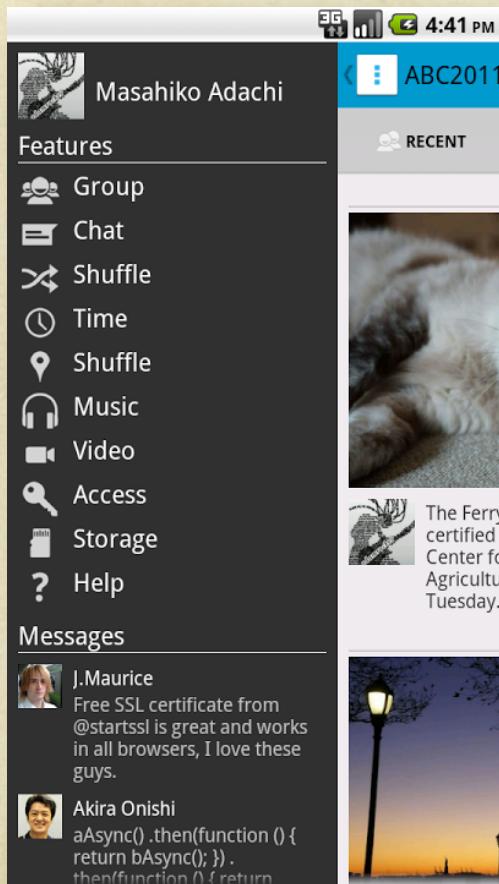
- 憧れのDrawerナビゲーションを実装できる



DrawerLayout

```
getActionBar().setDisplayHomeAsUpEnabled(true);  
getActionBar().setHomeButtonEnabled(true);  
  
drawerToggle = new ActionBarDrawerToggle(  
    this, mDrawerLayout,  
    R.drawable.icon, R.string.open_label, R.string.close_label);  
mDrawerLayout.setDrawerListener(drawerToggle);
```

おまけ:simple-side-drawer

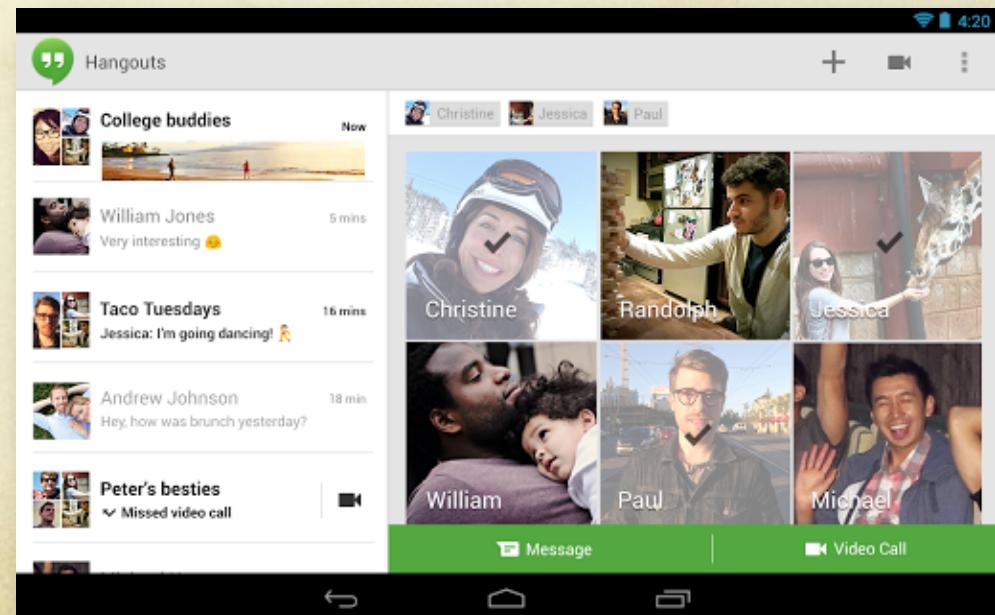
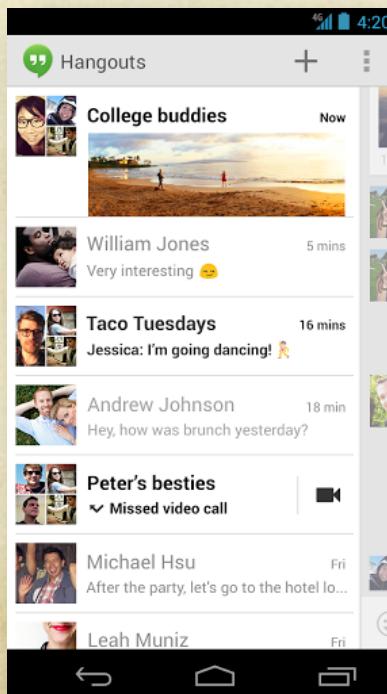


```
protected void onCreate(Bundle data) {  
    super.onCreate( data );  
    setContentView( R.layout.main );  
  
    mSide = new SimpleSideDrawer( this );  
    mSide.setLeftBehindContentView( R.layout.behind_menu_left );  
}
```

<https://github.com/adamrocker/simple-side-drawer>

SlidingPaneLayout

- マルチペインのメニューが作れる
- 画面が大きい時はメニューも自動で表示される



android-support-v7.jar

ActionBar

- ActionBarがAndroid 2.1にキター！

```
class ActionActivity extends ActionBarActivity {  
}
```

/res/values/styles.xml

```
<style name="AppBaseTheme" parent="Theme.ActionCompat">
```

Volley

What's Volley

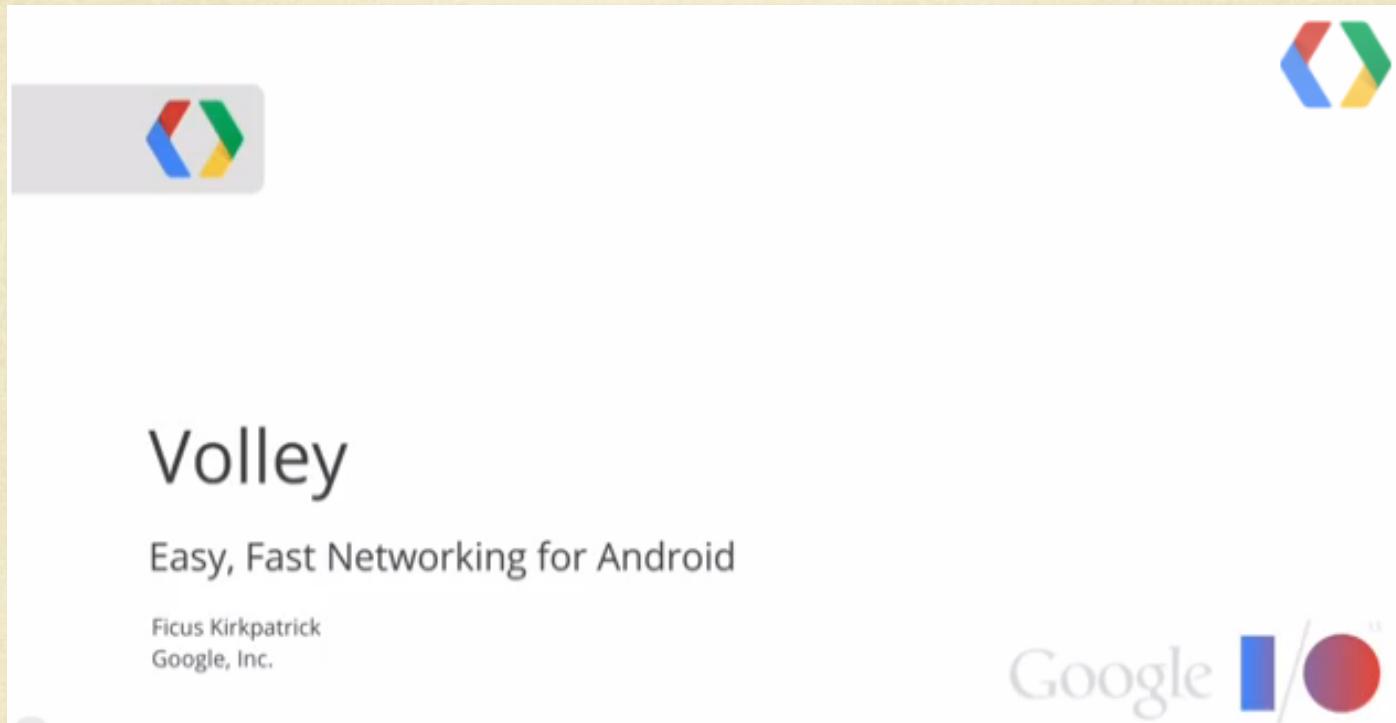
What is Volley?

volley (vä-lē), n.:

a burst or
emission of many
things or a large
amount at once



What's Volley



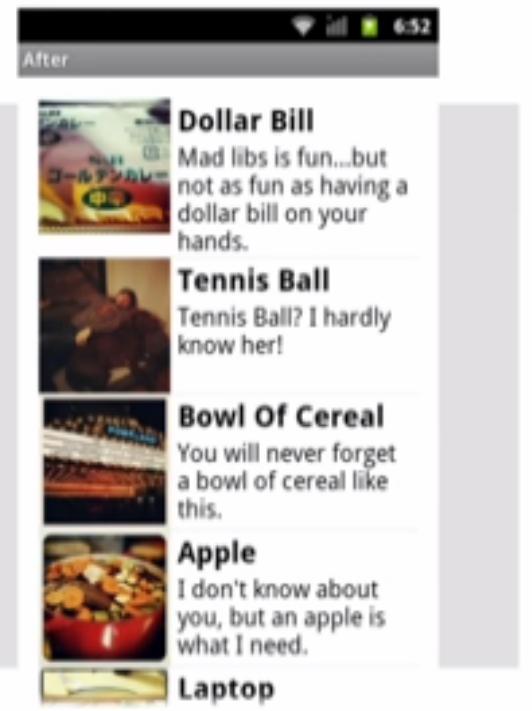
簡単で速い**非同期**HTTP通信ライブラリ
実はGoogle Playアプリも使ってるらしい

例：大量に通信する

Simple JSON protocol

GET /api/list HTTP/1.1

```
{"items": [
  {
    "title": "Dollar Bill",
    "description": "Please. Mr. Y'all was my father.",
    "image_url": "/static/24.jpg"
  },
  {
    "title": "Tennis Ball",
    "description": "Every dog's favorite.",
    "image_url": "/static/60.jpg"
  },
  ...
],
"next": "10_10" }
```



最近良く見かけるTime Line型のアプリとか

AsyncTaskとの違い

- AsyncTaskよりシンプル
- Viewとの連携でより簡単
- 通信まわりの実装が限りなく隠蔽されている
- Androidのバグも回避してくれるw

Volley実装例

Volley implementation

Setup

Java

```
// Somewhere common; app startup or adapter constructor  
  
mRequestQueue = Volley.newRequestQueue(context);  
mImageLoader = new ImageLoader(mRequestQueue, new BitmapLruCache());
```

Volley実装例

Volley implementation

loadMoreData()

Java

```
mRequestQueue.add(new JsonObjectRequest(Method.GET, url, null,
    new Listener<JSONObject>() {
        public void onResponse(JSONObject jsonRoot) {
            mNextPageToken = jsonGet(jsonRoot, "next", null);
            List<Items> items = parseJson(jsonRoot);
            appendItemsToList(item);
            notifyDataSetChanged();
        }
    }
})
```

Volleyで画像の取得と表示

Volley implementation

Using NetworkImageView

- <ImageView

XML

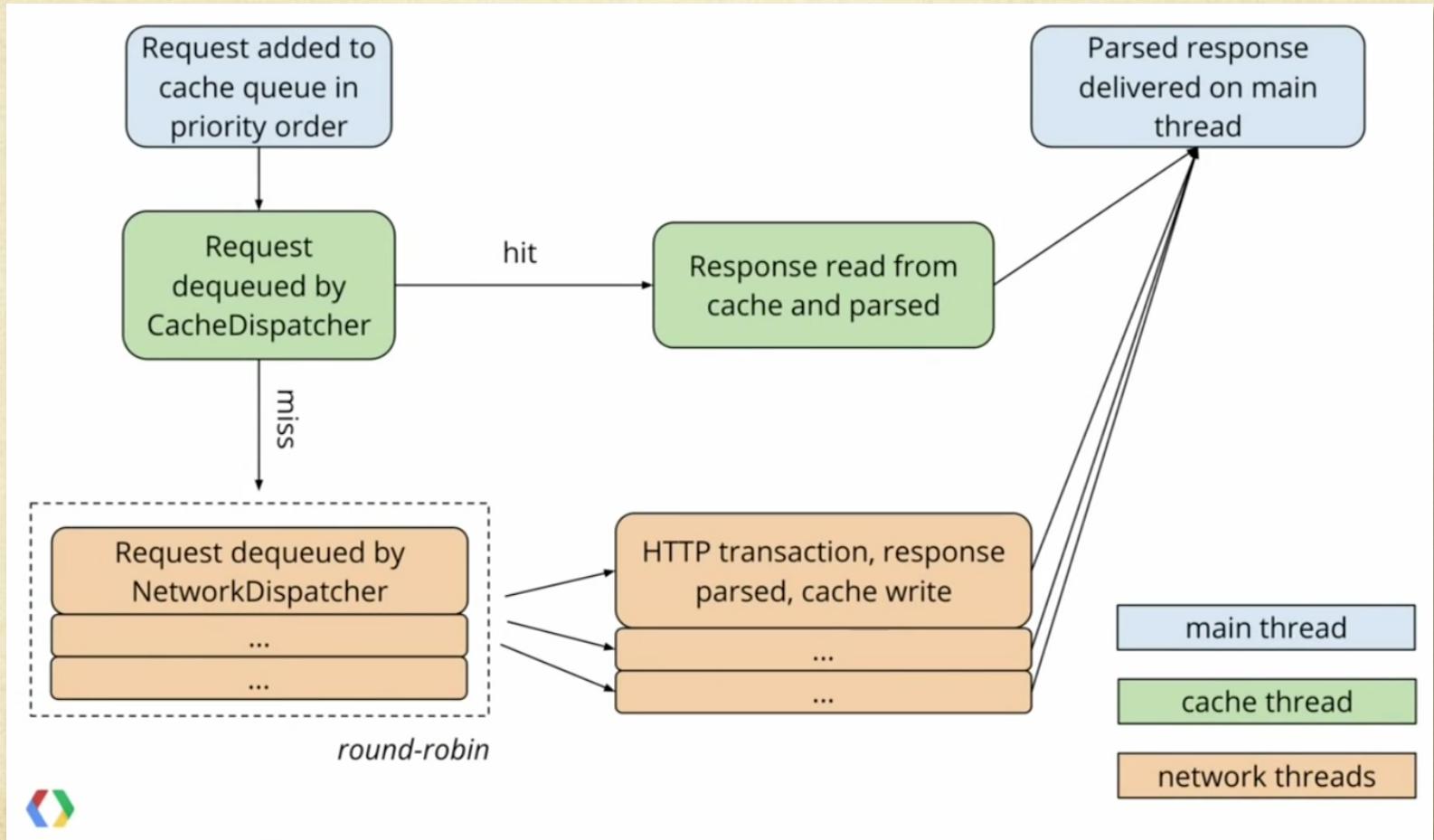
+ <com.android.volley.NetworkImageView

mImageView.setImageUrl(BASE_URL + item.image_url, mImageLoader);

Java

```
mRequestQueue = Volley.newRequestQueue(context);
mImageLoader = new ImageLoader(mRequestQueue, new BitmapLruCache());
```

Volleyの実装



とてもシンプル

Volleyのいいところ

- 通信順序を入れ替えられる
- キャッシュできる、キャッシュアルゴリズムも変更できる
- HttpClientとHttpURLConnectionのバグを回避してくれる
- 通信のデバッグが簡単

```
adb shell setprop log.tag.Volley VERBOSE
```

Volleyのはじめ方

- GitのVolleyプロジェクトをClone
- Volleyのコードを自分のプロジェクトにImport
- Volley.newRequestQueue(context);

```
git clone https://android.googlesource.com/platform/frameworks/volley
```

サンプルアプリつくりました

<https://github.com/adamrocker/volley>

And more

- 世界ライブラリランキング
- AppBrain

<http://www.appbrain.com/stats/libraries/dev>

まとめ

- NavUtils と TaskStackBuilder で Page Navigation をデザイン
- AsyncTaskLoader で非同期データアクセス
- LocalBroadcastManager でアプリ内で暗黙の Intent
- SparseArray で高速・軽量 Hashmap
- DrawerLayout で憧れの Drawer Navigation
- Android 2.1 からでも簡単に ActionBar
- Volley で超簡単に大量の非同期 HTTP 通信

Thank you

Discover Support Library

Masahiko Adachi

@adamrokcer / +Masahiko.Adachi

28th Sep, 2013

