

# 実践 ANDROIDアプリ開発

@JX通信社

酒本伸也

2015/8/21

LIST

# LISTVIEW

- ほとんどのアプリで使われるWidget
- 同じセルを繰り返すデザイン

- ListView

- スクロールはデフォルトである

- cell

- 繰り返す要素





# 必要なもの

- ListView
  - レイアウトで定義済み
- Adapter
  - ListViewやGridViewのような子要素が定義されたView(AdapterView)とDataをつなぐ役割
- Data
  - ListViewに表示させたいデータのリスト(前回取得済み)

```
public class MainListActivity extends AppCompatActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_list);
        Log.d("TAG", "onCreate");

        ListView listView = (ListView)findViewById(R.id.list_view);

        String[] strings = new String[20];
        for (int i = 0; i < 20; i++) {
            strings[i] = String.valueOf(i);
        }
        ArrayAdapter<String> adapter =
            new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, strings);
        listView.setAdapter(adapter);
    }
}
```

ListViewを現在のレイアウトから取得

```
public class MainListActivity extends AppCompatActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_list);
        Log.d("TAG", "onCreate");

        ListView listView = (ListView)findViewById(R.id.list_view);

        String[] strings = new String[20];
        for (int i = 0; i < 20; i++) {
            strings[i] = String.valueOf(i);
        }
        ArrayAdapter<String> adapter =
            new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, strings);
        listView.setAdapter(adapter);
    }
}
```

ダミーのデータを生成  
(取得した記事データを用いる)



```
public class MainListActivity extends AppCompatActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_list);
        Log.d("TAG", "onCreate");

        ListView listView = (ListView)findViewById(R.id.list_view);

        String[] strings = new String[20];
        for (int i = 0; i < 20; i++) {
            strings[i] = String.valueOf(i);
        }
        ArrayAdapter<String> adapter =
            new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, strings);
        listView.setAdapter(adapter);
    }
}
```

Adapterを生成してListViewにセット



```
public class MainListActivity extends AppCompatActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_list);
        Log.d("TAG", "onCreate");

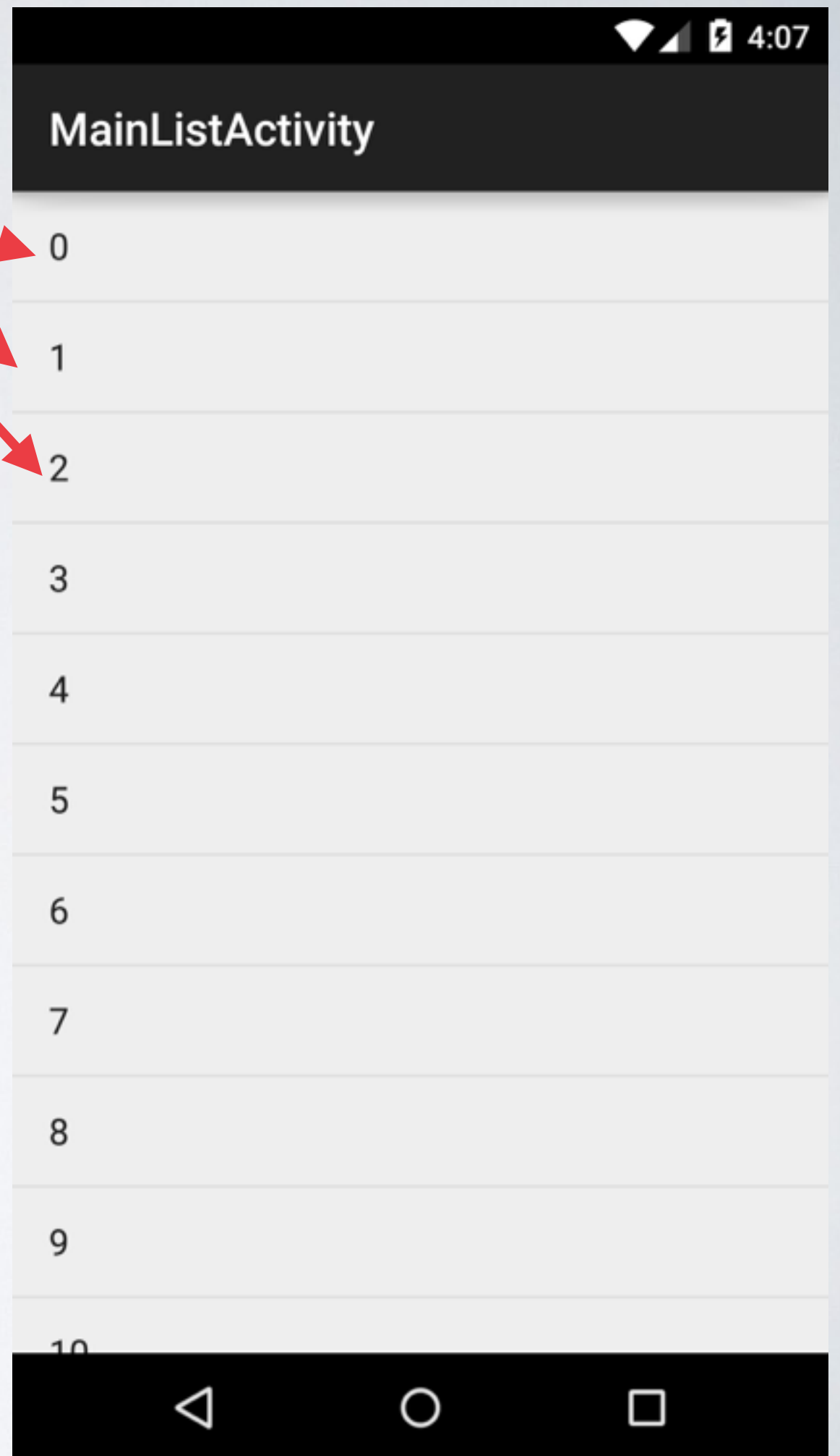
        ListView listView = (ListView)findViewById(R.id.list_view);

        String[] strings = new String[20];
        for (int i = 0; i < 20; i++) {
            strings[i] = String.valueOf(i);
        }

        ArrayAdapter<String> adapter =
            new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, strings);
        listView.setAdapter(adapter);
    }
}
```

android.R.layout.simple\_list\_item\_1  
デフォルトで用意されているcellのレイアウト

- Dataが入っている



android.R.layout.simple\_list\_item\_1

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:gravity="center_vertical"
    android:paddingStart="?android:attr/listPreferredItemPaddingStart"
    android:paddingEnd="?android:attr/listPreferredItemPaddingEnd"
    android:minHeight="?android:attr/listPreferredItemHeightSmall" />
```

TextViewしか用意されていない

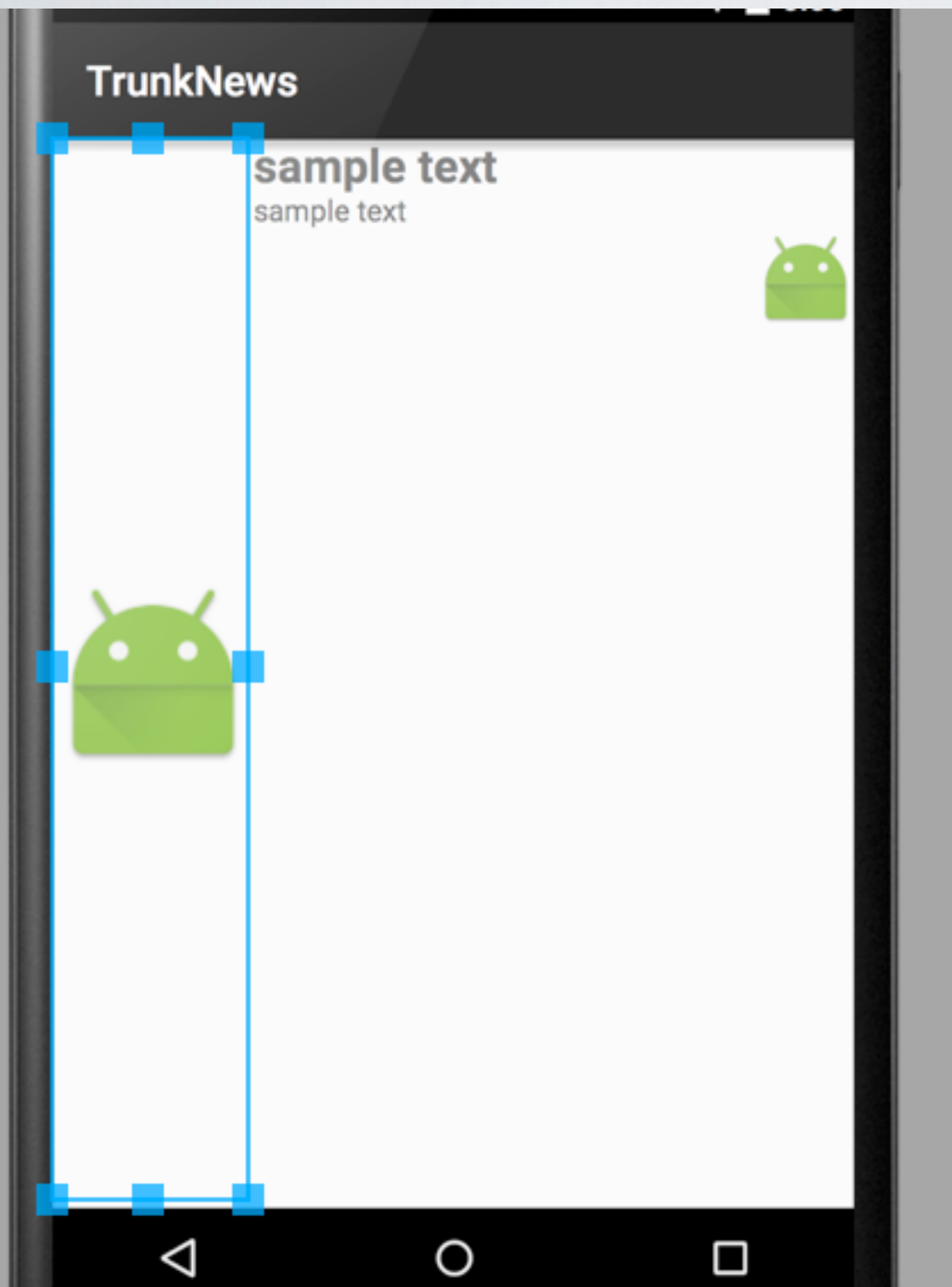
カスタムLISTVIEW



# カスタムLISTVIEW

- cellのレイアウトを定義
- Adapterを定義
- Dataを詰める
- Dataを描画

# CELLレイアウト



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3               xmlns:tools="http://schemas.android.com/tools"
4               android:orientation="horizontal"
5               android:layout_width="match_parent"
6               android:layout_height="match_parent">
7
8     <ImageView
9         android:id="@+id/image_thumbnail"
10        android:layout_weight="3"
11        tools:src="@mipmap/ic_launcher"
12        android:layout_width="match_parent"
13        android:layout_height="match_parent"/>
14
15    <LinearLayout
16        android:id="@+id/layout_article_right"
17        android:orientation="vertical"
18        android:layout_width="match_parent"
19        android:layout_height="match_parent">
20
21        <TextView
22            android:id="@+id/text_title"
23            tools:text="sample text"
24            android:textSize="22sp"
25            android:textStyle="bold"
26            android:layout_gravity="center_vertical"
27            android:layout_width="match_parent"
28            android:layout_height="wrap_content"/>
29
30        <TextView
31            android:id="@+id/text_body"
32            tools:text="sample text"
33            android:layout_gravity="center_vertical"
34            android:layout_width="match_parent"
35            android:layout_height="wrap_content"/>
36
37        <ImageView
38            android:id="@+id/image_share"
39            tools:src="@mipmap/ic_launcher"
40            android:layout_gravity="bottom|right"
41            android:layout_width="wrap_content"
42            android:layout_height="wrap_content"/>
43    </LinearLayout>
44</LinearLayout>
```

# ADAPTERを定義

- ArrayAdapterを継承する
- 必要なデータはコンストラクタに含める
- 適切なメソッドをOverrideする

# ARRAYADAPTERを継承

- <>の中はDataの型にする
- コンストラクタの引数はメンバに渡す

```
public class ArticleBadAdapter extends ArrayAdapter<Article>{  
  
    private Context mContext;  
    private int mResourceId;  
    private List<Article> mArticleList;  
  
    public ArticleBadAdapter(Context context, int resource, List<Article> objects) {  
        super(context, resource, objects);  
        this.mContext = context;  
        this.mResourceId = resource;  
        this.mArticleList = objects;  
    }  
}
```



# OVERRRIDEが必要なメソッド

- getCount()
- getItem(int position)
- getView(int position, View convertView, ViewGroup parent)

- getCount()
  - 描画するcellの数を決める。  
Dataの個数を返す。
- getItem(int position)
  - 現在のcellの位置に応じたDataを返す
- getView(int position, View convertView, ViewGroup parent)
  - 描画するcellを返す

- getCount()
- getItem(int position)

```
@Override
public int getCount() {
    return mArticleList.size();
}

@Override
public Article getItem(int position) {
    return mArticleList.get(position);
}
```

- position: リスト内での位置
- convertView: cell
- parent: ListView

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if (convertView == null) {
        convertView = View.inflate(getContext(), mResourceId, null);
    }

    TextView title = (TextView)convertView.findViewById(R.id.text_title);
    title.setText(mArticleList.get(position).getTitle());
    TextView body = (TextView)convertView.findViewById(R.id.text_body);
    body.setText(mArticleList.get(position).getBody().substring(0, 100));

    ImageView thumb = (ImageView)convertView.findViewById(R.id.image_thumbnail);
    thumb.setImageResource(R.mipmap.ic_launcher);

    ImageView share = (ImageView)convertView.findViewById(R.id.image_share);
    share.setImageResource(R.mipmap.ic_launcher);

    return convertView;
}
```



- convertViewに

定義したLayoutを読み込ませる

- Layout(RelativeとかLinearをか)を定義してある

のでnull

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if (convertView == null) {
        convertView = View.inflate(getContext(), mResourceId, null);
    }

    TextView title = (TextView)convertView.findViewById(R.id.text_title);
    title.setText(mArticleList.get(position).getTitle());
    TextView body = (TextView)convertView.findViewById(R.id.text_body);
    body.setText(mArticleList.get(position).getBody().substring(0, 100));

    ImageView thumb = (ImageView)convertView.findViewById(R.id.image_thumbnail);
    thumb.setImageResource(R.mipmap.ic_launcher);

    ImageView share = (ImageView)convertView.findViewById(R.id.image_share);
    share.setImageResource(R.mipmap.ic_launcher);

    return convertView;
}
```

- mArticleList.get(position) でDataを取得
- 定義したLayoutへアクセスし、Dataを詰める

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if (convertView == null) {
        convertView = View.inflate(getContext(), mResourceId, null);
    }

    TextView title = (TextView)convertView.findViewById(R.id.text_title);
    title.setText(mArticleList.get(position).getTitle());
    TextView body = (TextView)convertView.findViewById(R.id.text_body);
    body.setText(mArticleList.get(position).getBody().substring(0, 100));

    ImageView thumb = (ImageView)convertView.findViewById(R.id.image_thumbnail);
    thumb.setImageResource(R.mipmap.ic_launcher);

    ImageView share = (ImageView)convertView.findViewById(R.id.image_share);
    share.setImageResource(R.mipmap.ic_launcher);

    return convertView;
}
```

- 定義したLayoutの各要素に  
都度アクセスするのは  
パフォーマンスが悪い

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if (convertView == null) {
        convertView = View.inflate(getContext(), mResourceId, null);
    }

    TextView title = (TextView)convertView.findViewById(R.id.text_title);
    title.setText(mArticleList.get(position).getTitle());
    TextView body = (TextView)convertView.findViewById(R.id.text_body);
    body.setText(mArticleList.get(position).getBody().substring(0, 100));

    ImageView thumb = (ImageView)convertView.findViewById(R.id.image_thumbnail);
    thumb.setImageResource(R.mipmap.ic_launcher);

    ImageView share = (ImageView)convertView.findViewById(R.id.image_share);
    share.setImageResource(R.mipmap.ic_launcher);

    return convertView;
}
```



# VIEWHOLDERパターン

- ViewHolder(名前はなんでも良い)クラスを内部に作成し、要素へのアクセスを高速化する



# VIEWHOLDER

- 各Widgetへのアクセスを持つのみ
- インナークラスとして定義

```
private static class ViewHolder {  
    private TextView mTextTitle;  
    private TextView mTextBody;  
    private ImageView mImageThumbnail;  
    private ImageView mImageShare;  
  
    public ViewHolder(View view) {  
        mTextTitle = (TextView) view.findViewById(R.id.text_title);  
        mTextBody = (TextView) view.findViewById(R.id.text_body);  
        mImageThumbnail = (ImageView) view.findViewById(R.id.image_thumbnail);  
        mImageShare = (ImageView) view.findViewById(R.id.image_share);  
    }  
}
```

- setTag/getTag(Object) で ViewHolder を保持

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

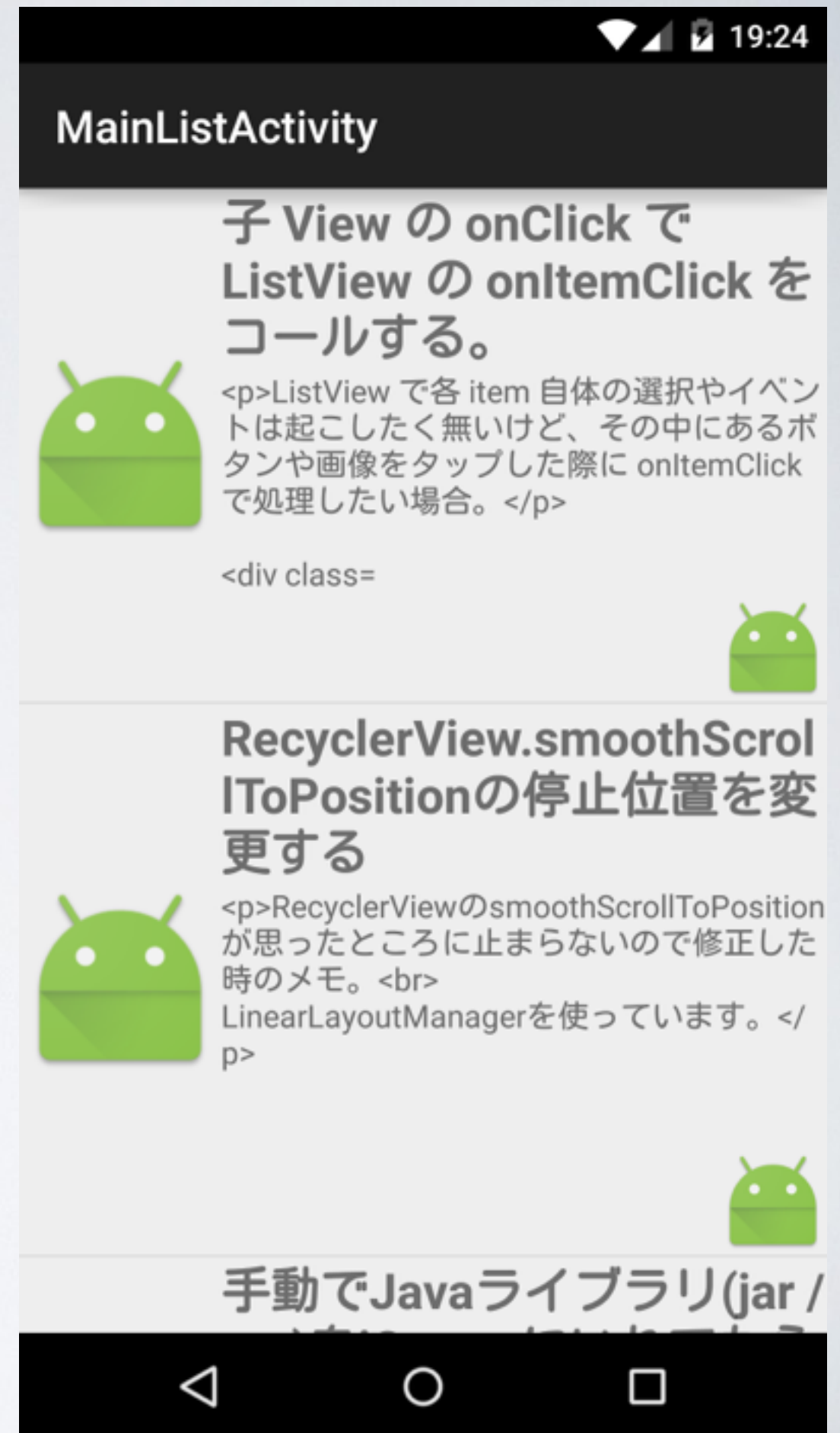
    ViewHolder vh;
    if (convertView == null) {
        convertView = View.inflate(getContext(), mResourceId, null);
        vh = new ViewHolder(convertView);
        convertView.setTag(vh);
    } else {
        vh = (ViewHolder)convertView.getTag();
    }

    Article article = mArticleList.get(position);

    vh.mTextTitle.setText(article.getTitle());
    String planeBody = article.getBody().replaceAll("<.+?>", "");
    vh.mTextBody.setText(planeBody); // substring(0, 100));
    vh.mImageThumbnail.setImageResource(R.mipmap.ic_launcher);
    vh.mImageShare.setImageResource(R.mipmap.ic_launcher);

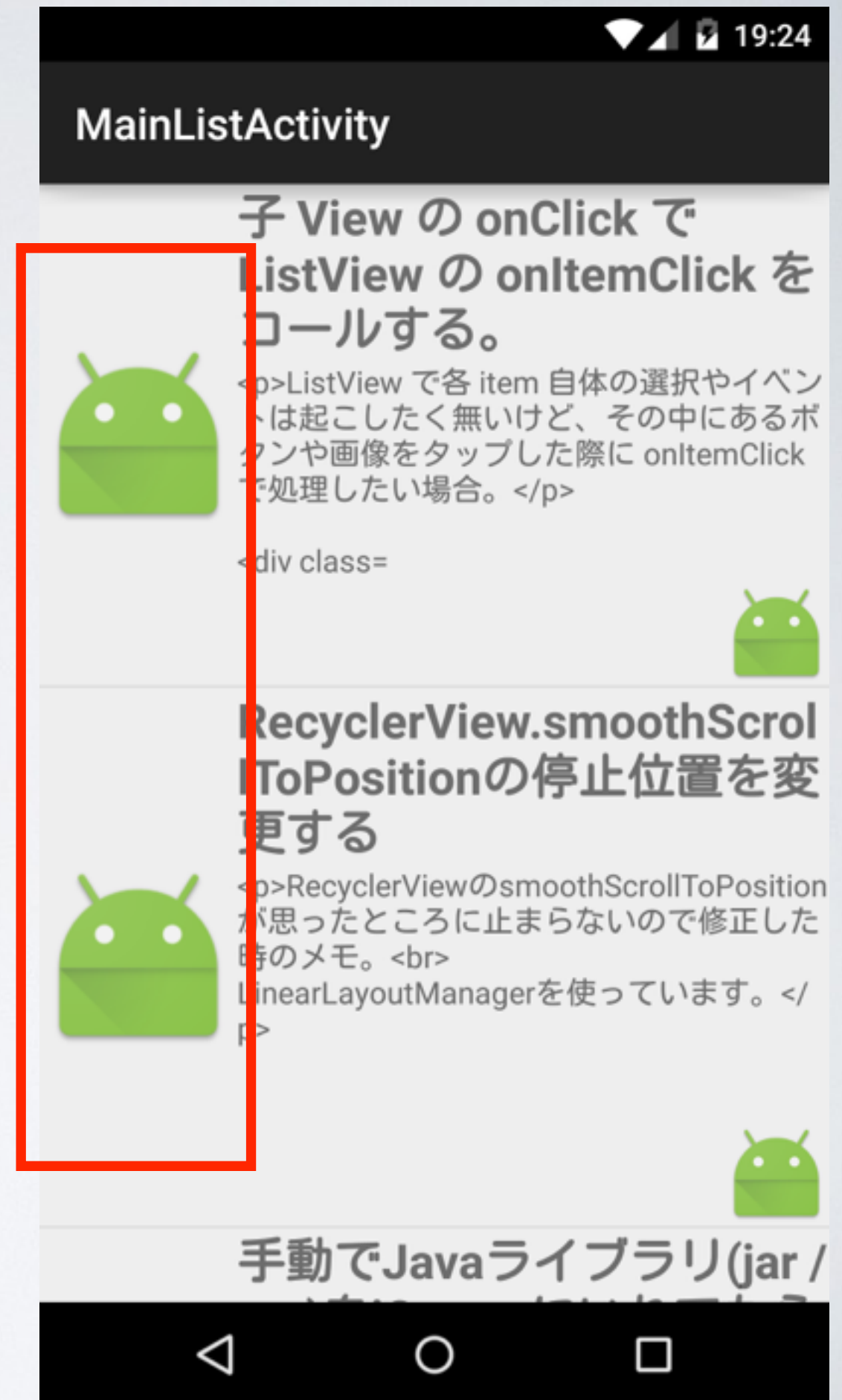
    return convertView;
}
```

- こんな感じ





- 画像はどうした





# 非同期

- 画像は端末内にない場合、http通信をして取得してくる必要がある
- 非同期で処理を行わないとListViewが固まる

PICASSO

# PICASSO

- 画像まわりのライブラリ
- 非同期で取得や、キャッシュもしてくれる

# BUILD.GRADLE

- compile 'com.squareup.picasso:picasso:2.5.2'



# 一行

- Picasso.with(mContext).load(Uri.parse(url)).into(mImageView);

# 詳細

- <http://qiita.com/hotchemi/items/33ebd5faa42d2d05c2b6>

# クリックイベント

- ListViewに対して、  
Buttonと同じように実装する

- setOnItemClickListener
  - setOnClickListenerではない
- ListViewに対してのリスナー

```
mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        Article article = (Article)parent.getItemAtPosition(position);  
        Toast.makeText(MainListActivity.this, article.getTitle(), Toast.LENGTH_SHORT).show();  
    }  
});
```



- AdapterからDataを取得する

```
mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        Article article = (Article)parent.getItemAtPosition(position);  
        Toast.makeText(MainActivity.this, article.getTitle(), Toast.LENGTH_SHORT).show();  
    }  
});
```

- タイトルが表示される！

