# 実践

# ANDROIDアプリ開発

@J×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



北が韓国に砲撃、韓国側も反撃…住民に退避命令 2015年08 月20日 23時10分

読売新聞 - 1時間前

■ ヘッドライン



年金情報流出まで13日間対応 怠る P C 感染で調査報告

朝日新聞 - 42分前



中露の合同演習、日本海で開 始…初の上陸演習も

読売新聞 - 2時間前



【米大統領選】共和・トランプ 氏の旋風やむかVS民主・下降 気味のクリントン氏

産経ニュース - 1時間前



首相事務所、「吐血」報道に抗 議文

日本経済新聞 - 1時間前

# 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 (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 → (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 → (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 → (this, android.R.layout simple_list_item_1
                                                                               strings);
       listView.setAdapter(adapter);
```

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



#### android.R.layout.simple\_list\_item\_l

```
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



```
<?xml version="1.0" encoding="utf-8"?>
       <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
                      xmlns:tools="http://schemas.android.com/tools"
                      android:orientation="horizontal"
                      android: layout_width="match_parent"
                      android: layout_height="match_parent">
            <ImageView
                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
            <LinearLayout</pre>
15
                android:id="@+id/layout_article_right"
16
                android:orientation="vertical"
17
                android: layout_weight="1"
                android: layout_width="match_parent"
                android: layout_height="match_parent">
20
21
                <TextView
22
                    android:id="@+id/text_title"
23
                    tools:text="sample text"
24
                    android:textSize="22sp"
                    android:textStyle="bold"
26
                    android: layout_gravity="center_vertical"
27
                    android: layout width="match parent"
                    android: layout_height="wrap_content"/>
29
                <TextView
30
                    android:id="@+id/text_body"
31
                    tools:text="sample text"
32
                    android: layout_gravity="center_vertical"
33
                    android:layout_width="match_parent"
                    android: layout_height="wrap_content"/>
                <ImageView
36
                    android:id="@+id/image_share"
37
                    tools:src="@mipmap/ic_launcher"
                    android: layout_gravity="bottom|right"
                    android: layout_width="wrap_content"
40
                    android:layout_height="wrap_content"/>
41
42
            </LinearLayout>
43
       </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;
    }
```

## OVERRIDE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/Z

### 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) {
        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 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;
```



#### MainListActivity

#### 子 View の onClick で ListView の onItemClick を コールする。



ListView で各 item 自体の選択やイベントは起こしたく無いけど、その中にあるボタンや画像をタップした際に onItemClickで処理したい場合。

<div class=



#### RecyclerView.smoothScrol IToPositionの停止位置を変 更する



RecyclerViewのsmoothScrollToPositionが思ったところに止まらないので修正した時のメモ。<br/>
らないので修正した

LinearLayoutManagerを使っています。



手動でJavaライブラリ(jar/









#### MainListActivity

#### 子 View の onClick で ListView の onItemClick を コールする。



p>ListView で各 item 自体の選択やイベン 、は起こしたく無いけど、その中にあるボ アンや画像をタップした際に onItemClick で処理したい場合。

<div class=



#### RecyclerView.smoothScrol IToPositionの停止位置を変 更する



p>RecyclerViewのsmoothScrollToPositionが思ったところに止まらないので修正した のメモ。<br/>
ちのメモ。<br/>
のメモ。<br/>
のメート<br/>
のより<br/>

LinearLayoutManagerを使っています。</



手動でJavaライブラリ(jar/

◁





# 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(mlmageView);

# 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

setOnltemClickListener

• 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);
        Teast makeText(MainListActivity this, article getTitle(), Teast .LENGTH_SHORT).show();
    }
});
```





#### MainListActivity



#### [Lollipop]UsageStatsMana gerの権限状態の確認方法

UsageStatsManagerの権限確認

UsageStatsManagerは、Lollipop以降で ActivityManagerのgetRunningTasksなどの 各種メソッドの代替と

### AndroidでIPv6を無効化する



2015年8月上旬頃よりWi-Fi環境で(公式アプリ、非公式アプリに関係なく)Twitter画像が表示されないという事象が発生しています。どうやらauひかり回線で特定のプロバイダ(または特定の端末)の

子 View の onClick で ListView の onItemClick を コールする。

[Android][Lollipop]UsageStatsManagerの 権限状態の確認方法

XX 07C 0 -80 C 0

YourAdapter



◁

