

Android/개발

## [CustomListView] 안드로이드 커스텀 리스트뷰

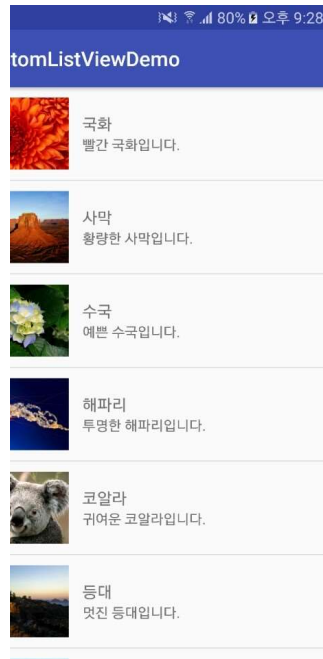
2017.02.07 22:15

## 개발자 키우기



이매촌

## 안드로이드 간단하게 Custom ListView 만들기



RecyclerView는 안드로이드 개발을 하면서 정말 많이 쓰이는 view중 하나입니다.

하지만 RecyclerView를 많이 사용하지만 때에 저 같은 초보자는 때에따라 ListView를 사용해야 할 때가 있습니다.

이 글에서는 ListView에 대해 정리하는 차원에서 Custom ListView를 간단하게 만들어봤습니다.

이 글에서는 Custom ListView란 여러개의 뷰를 ListView에 보여주는 ListView를 말합니다!

이 글에서는 Custom ListView를 만들기 위해서는 4가지만 준비하시면 됩니다.

## 1. activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.imaec.customlistviewdemo.MainActivity">

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"

```

```

        android:layout_height="match_parent" />
    </RelativeLayout>

```

개의 View들을 담은 ListView를 만들어 줍니다.

## 개발자 키우기



이매촌

### m - item\_custom.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="100dp"
        android:layout_height="70dp"
        android:layout_marginRight="10dp"
        android:src="@mipmap/ic_launcher" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center_vertical">

        <TextView
            android:id="@+id/text_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="title"
            android:textSize="16sp" />

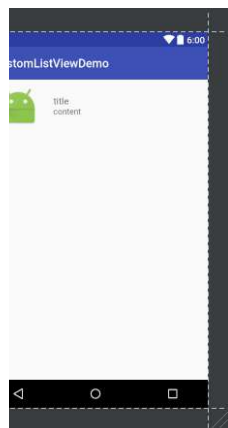
        <TextView
            android:id="@+id/text_content"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="content" />

    </LinearLayout>
</RelativeLayout>

```

은 ListView안에 나올 View들을 배치해줍니다. 모양을 잡으시고 사용할 View들의 id를 만들어주세요.

iew



## O - CustomDTO.java

```

<age> com.imaec.customlistviewdemo;

lic class CustomDTO {
    private int resId;
    private String title;
    private String content;

    public int getResId() {
        return resId;
    }

    public void setResId(int resId) {
        this.resId = resId;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }
}

```

## 개발자 키우기



이매촌

**data transfer object**)는 계층간 데이터교환을 위한 것으로 class이름은 편하신대로 하셔도 무방합니다.

## apter - CustomAdapter.java

```

<age> com.imaec.customlistviewdemo;

ort android.view.LayoutInflater;
ort android.view.View;
ort android.view.ViewGroup;
ort android.widget.BaseAdapter;
ort android.widget.ImageView;
ort android.widget.TextView;

ort java.util.ArrayList;

lic class CustomAdapter extends BaseAdapter {

    private ArrayList<CustomDTO> listCustom = new ArrayList<>();

    // ListView에 보여질 Item 수
    @Override
    public int getCount() {
        return listCustom.size();
    }

    // 하나의 Item(ImageView 1, TextView 2)
    @Override
    public Object getItem(int position) {
        return listCustom.get(position);
    }

    // Item의 id : Item을 구별하기 위한 것으로 position 사용
}

```

## 개발자 키우기



이매촌

```

@Override
public long getItemId(int position) {
    return position;
}

// 실제로 Item이 보여지는 부분
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    CustomViewHolder holder;
    if (convertView == null) {
        convertView = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_custom, null, false);

        holder = new CustomViewHolder();
        holder.imageView = (ImageView) convertView.findViewById(R.id.imageView);
        holder.textTitle = (TextView) convertView.findViewById(R.id.text_title);
        holder.textContent = (TextView) convertView.findViewById(R.id.text_content);

        convertView.setTag(holder);
    } else {
        holder = (CustomViewHolder) convertView.getTag();
    }

    CustomDTO dto = listCustom.get(position);

    holder.imageView.setImageResource(dto.getResId());
    holder.textTitle.setText(dto.getTitle());
    holder.textContent.setText(dto.getContent());

    return convertView;
}

class CustomViewHolder {
    ImageView imageView;
    TextView textTitle;
    TextView textContent;
}

// MainActivity에서 Adapter에있는 ArrayList에 data를 추가시켜주는 함수
public void addItem(CustomDTO dto) {
    listCustom.add(dto);
}

```

너 사용된 ViewHolder 안드로이드의 퍼포먼스를 높이기 위한 것으로 사용하지 않아도 기능상의 문제는 없습니다.

CustomListView를 만들 준비가 끝났습니다!

만든 4개를 MainActivity.java에서 선언하고 연결시켜주면 CustomListView가 완성됩니다!

## Activity.java

```

<age com.imaec.customlistviewdemo;

<rt android.content.res.TypedArray;
<rt android.support.v7.app.AppCompatActivity;
<rt android.os.Bundle;
<rt android.widget.ListView;

<ic class MainActivity extends AppCompatActivity {

    private CustomAdapter adapter;
    private ListView listView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

```

adapter = new CustomAdapter();
listView = (ListView) findViewById(R.id.listView);

setData();

listView.setAdapter(adapter);
}

// 보통 ListView는 통신을 통해 가져온 데이터를 보여줍니다.
// arrResId, titles, contents를 서버에서 가져온 데이터라고 생각하시면 됩니다.
private void setData() {
    TypedArray arrResId = getResources().obtainTypedArray(R.array.resId);
    String[] titles = getResources().getStringArray(R.array.title);
    String[] contents = getResources().getStringArray(R.array.content);

    for (int i = 0; i < arrResId.length(); i++) {
        CustomDTO dto = new CustomDTO();
        dto.setResId(arrResId.getResourceId(i, 0));
        dto.setTitle(titles[i]);
        dto.setContent(contents[i]);

        adapter.addItem(dto);
    }
}

```

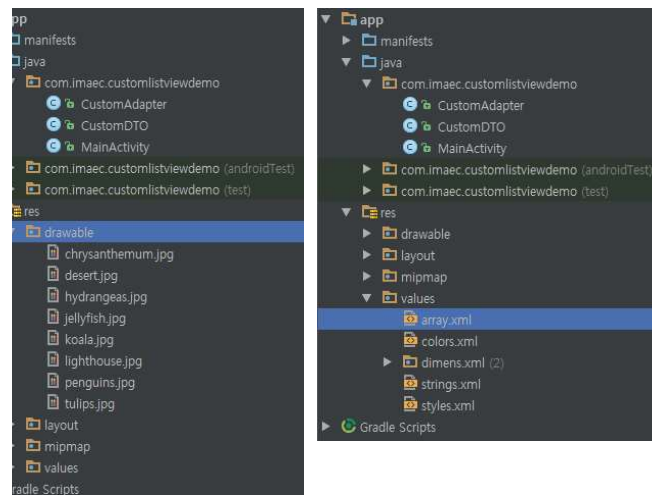
## 개발자 키우기



이매촌

이는 drawable폴더에 저장하여 사용했습니다.

drawable폴더와 array.xml 위치



resId, titles, contents는 **array.xml**에 저장된 배열을 사용합니다.

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <integer-array name="resId">
        <item>@drawable/chrysanthemum</item>
        <item>@drawable/desert</item>
        <item>@drawable/hydrangeas</item>
        <item>@drawable/jellyfish</item>
        <item>@drawable/koala</item>
        <item>@drawable/lighthouse</item>
        <item>@drawable/penguins</item>
        <item>@drawable/tulips</item>
    </integer-array>

    <string-array name="title">
        <item>국화</item>
        <item>사막</item>
        <item>수국</item>
        <item>해파리</item>
        <item>코알라</item>
        <item>등대</item>
        <item>펭귄</item>
        <item>튤립</item>
    </string-array>

```

# 개발자 키우기



이매촌

```
<string-array name="content">
    <item>빨간 국화입니다.</item>
    <item>황량한 사막입니다.</item>
    <item>예쁜 수국입니다.</item>
    <item>투명한 해파리입니다.</item>
    <item>귀여운 코알라입니다.</item>
    <item>멋진 등대입니다.</item>
    <item>사이좋은 펭귄입니다.</item>
    <item>노란 튜립입니다.</item>
</string-array>
</sources>
```

커스텀 리스트뷰 만드는 방법이었습니다.  
경험치 열심히 올려서 안드로이드 고수가 됩시다!

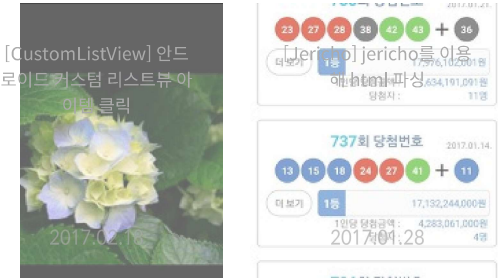
공감

roid> 개발' 카테고리의 다른 글

tomListView] 안드로이드 커스텀 리스트뷰 아이템 클릭 (0)	2017.0
<b>tomListView] 안드로이드 커스텀 리스트뷰</b> (0)	2017.0
chol jericho를 이용해 html 파싱 (0)	2017.0

android custom listview, custom listview, 안드로이드 Custom ListView, 안드로이드 커스텀 리스트뷰, 커스텀 리스트뷰

## roid/개발' Related Articles



## omments

Name

Password

여러분의 소중한 댓글을 입력해주세요

## 개발자 키우기

Blog is powered by kakao / Designed by Tistory



이매촌