

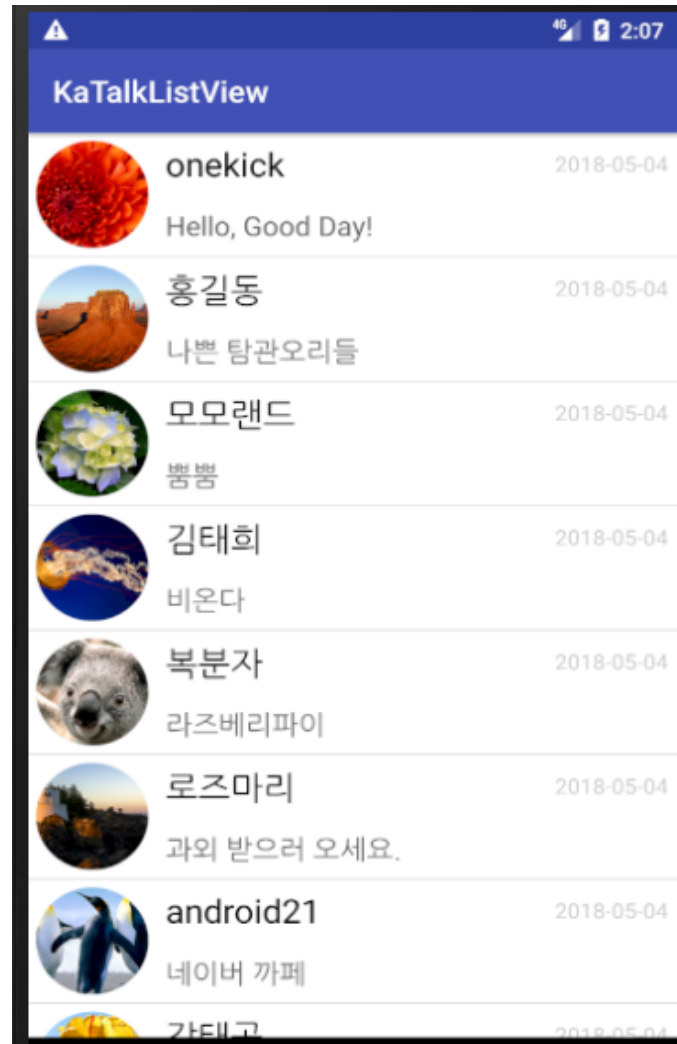
안드로이드 SDK 개발 플랫폼 활용하기

커스텀 리스트뷰 만들기

- Custom ListView
 - ListView의 각Item 들에 대한 디자인을 직접 정의해서 사용
 - ListView의 Adapter도 extend 해서 추가기능을 구현
 - 주로 getView()에서 Item의 View를 생성하거나 설정

- 모든 정보는 Adapter가 관리
 - getView()로 Item의 View 생성
 - ListView의 터치, 스크롤 처리
 - ListView는 가로 스크롤 미지원
 - 부모 클래스인 ScrollView 사용
 - 커스텀 리스트뷰의 아이템 만들기

- 카톡 친구 리스트 만들기

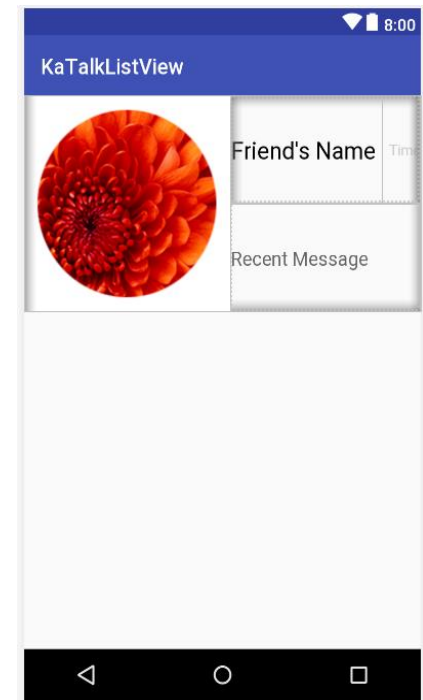
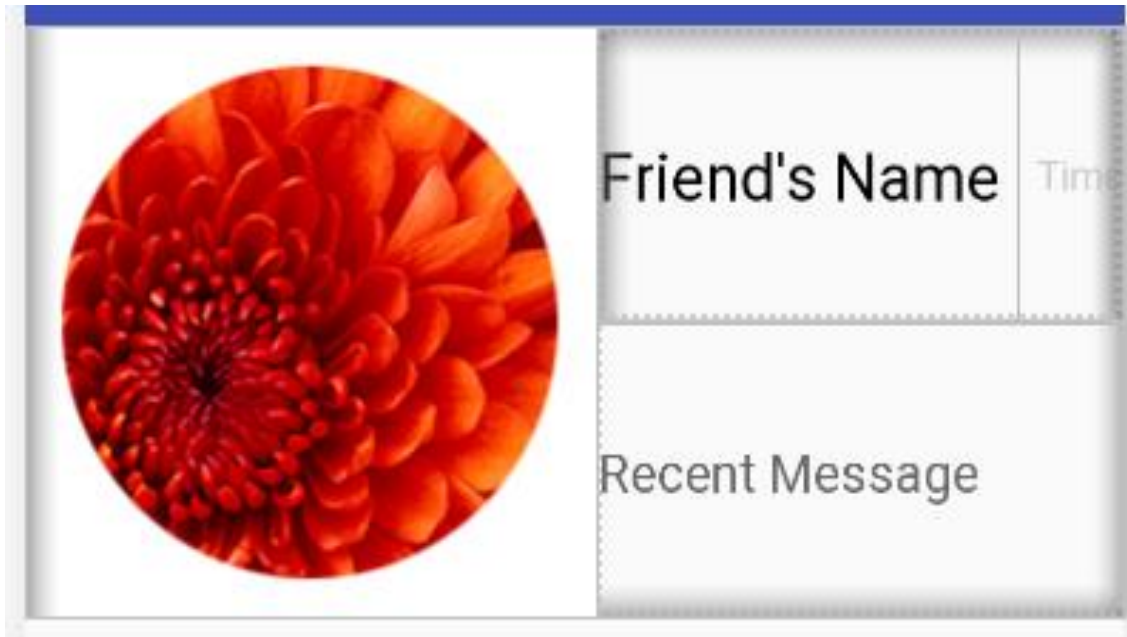


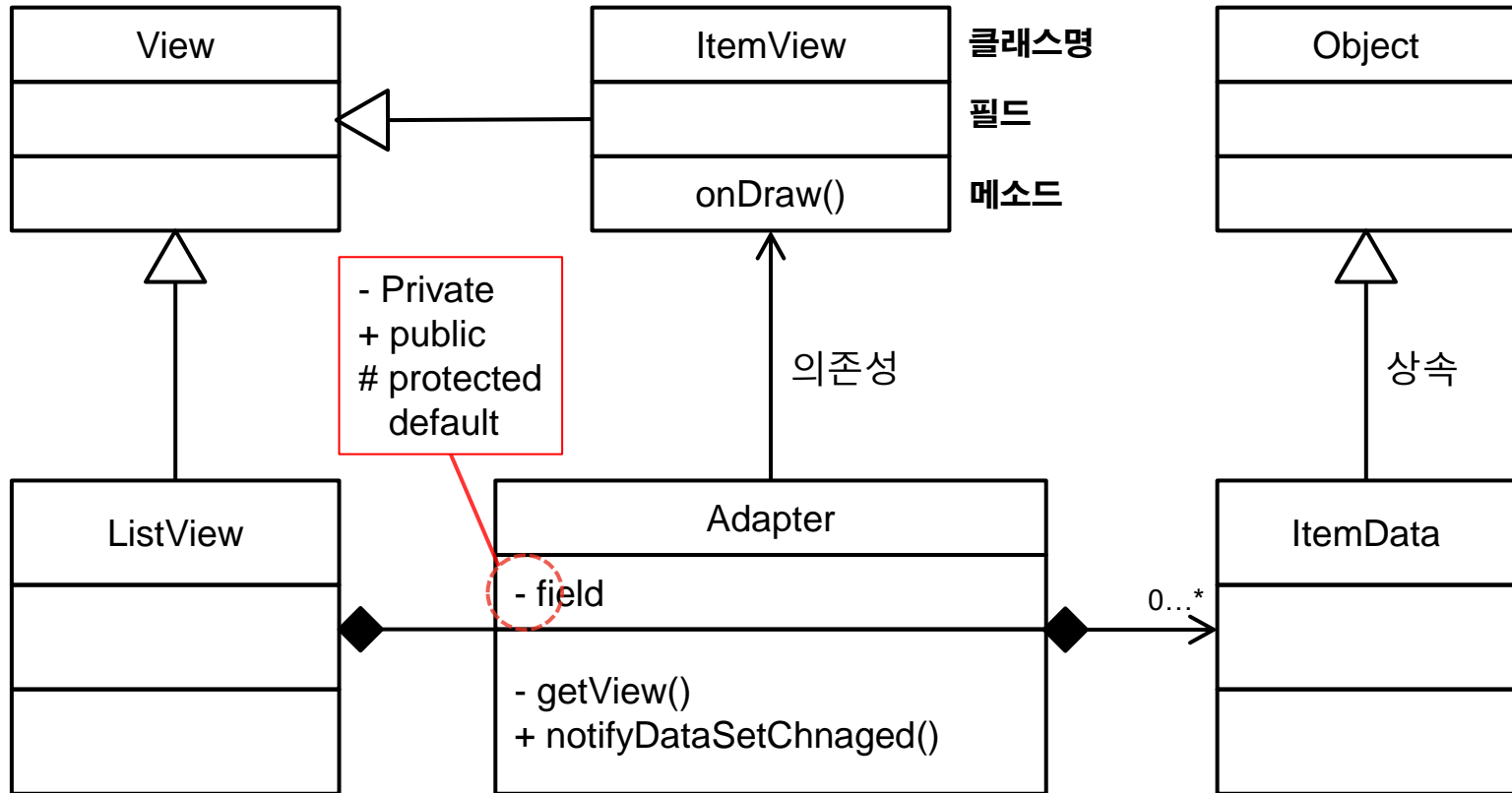
- 패키지명: com.iot.katalklistview
 - 기본클래스: MainActivity

- 소스코드 구성
 1. MainActivity.java
 - 기본화면을 구현
 2. 리스트 어댑터 클래스
 - KatalkAdapter.java
 3. 리스트의 각 아이템에 대한 데이터
 - ItemData.java
 4. 기본 화면의 activity_main.xml
 - 리스트뷰를 포함하도록 구현
 5. 리스트 뷰의 각 아이템에 대한 디자인
 - layout_item.xml

layout_item.xml 구조

- 리스트 아이템의 복잡한 구조
 - LinearLayout : 화면의 비율 차지, 영역 배치
 - FrameLayout : 꽃 그림 위에 마스킹 이미지 겹침
- 사진에 동그라미 마스킹 기능을 이용
 - ImageView 2장의 그림을 겹쳐서 표시
 - 아래쪽에는 사용자의 사진을 출력
 - 전면에는 동그란 구멍이 나 있는 마스킹용 이미지를 표시





```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="안드로이드 APK
파일의 스키마 경로는 생략합니다."
    xmlns:app="안드로이드 APK 파일의 스키마 경로는 생략"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.iot.katalklistview.KaTalkListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="안드로이드 APK 파일의 스키마 경로는 생략합니다."
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:padding="0dp"
    android:weightSum="10"
    android:background="#FFFFFF"
    >
    <!-- Image Layout -->
    <FrameLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:layout_margin="0dp"
        >
        <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:src="@drawable/image_1"
            android:scaleType="fitXY"
        />
```

```
<!-- Mask -->
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/image_mask"
    android:scaleType="fitXY"
/>
</FrameLayout>
<!-- Right Layout -->
<LinearLayout
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="8"
    android:orientation="vertical"
    android:weightSum="2"
>
<!-- Top Layout -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:weightSum="5"
>
```

```
<!-- Name -->
```

```
<TextView
```

```
    android:id="@+id/textViewName"
```

```
    android:layout_width="0dp"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_weight="3"
```

```
    android:text="Friend's Name"
```

```
    android:textColor="#000000"
```

```
    android:textSize="22dp"
```

```
    android:gravity="left|center"
```

```
    android:padding="5dp"
```

```
<!-- Time -->
```

```
<TextView
```

```
    android:id="@+id/textViewTime"
```

```
    android:layout_width="0dp"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_weight="2"
```

```
    android:text="Time"
```

```
    android:textColor="#cccccc"
```

```
    android:textSize="14dp"
```

```
    android:gravity="right|center"
```

```
    android:padding="5dp"
```

```
</LinearLayout>
<!-- Message -->
<TextView
    android:id="@+id/textViewMessage"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:text="Recent Message"
    android:textSize="18dp"
    android:textColor="#666666"
    android:gravity="left|center"
    android:padding="5dp"
/>
</LinearLayout>
</LinearLayout>
```

- 각 아이템의 데이터를 보관할 컨테이너 클래스를 구현

```
package com.iot.katalklistview;

public class ItemData {
    private int image = 0;
    private String name = null;
    private long time = 0;
    private String message = null;

    public ItemData(int imageRid, String name, long time, String message) {
        this.image = imageRid;
        this.name = name;
        this.time = time;
        this.message = message;
    }

    public int getImage() {
        return image;
    }

    public void setImage(int imageRid) {
        this.image = imageRid;
    }
}
```

```
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public long getTime() {  
    return time;  
}  
public void setTime(long time) {  
    this.time = time;  
}  
public String getMessage() {  
    return message;  
}  
public void setMessage(String message) {  
    this.message = message;  
}  
}
```

- 어댑터 클래스
 - 리스트뷰에 표시할 데이터를 직접 가지고 있으며 스크롤에 따라서 필요한 아이템 뷰 생성

```
package com.iot.katalklistview;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.graphics.Point;
import android.graphics.drawable.Drawable;
import android.os.Build;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.annotation.RequiresApi;
import android.util.Log; // 로그 출력 목적
import android.view.Display;
import android.view.LayoutInflater;
import android.view.View; // 뷰계열 최상위 클래스
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import java.text.SimpleDateFormat;
```

```
import java.util.ArrayList;
import java.util.Date;

public class KaTalkAdapter extends BaseAdapter {

    private static final String TAG = "KaTalkAdapter";
    private final Activity activity;
    private ArrayList<ItemData> arrayList = new ArrayList<>();

    public KaTalkAdapter(Activity activity) {
        this.activity = activity;
    }
    @Override
    public int getCount() {
        return arrayList.size();
    }
    @Override
    public Object getItem(int position) {
        return arrayList.get(position);
    }
}
```



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

```
@SuppressWarnings("SimpleDateFormat")
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
@NonNull
```

```
@Override
public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
    Log.i(TAG,"getView called");
    if(convertView==null) {
        Log.i(TAG,"getView: convertView==null");
        LayoutInflater inflater = activity.getLayoutInflater();
        convertView = inflater.inflate(R.layout.layout_item, parent,false);
    }
}
```

```
// convertView is a LinearLayout who root node of xml.
LinearLayout linearLayout = (LinearLayout)convertView;

// resize height of linear layout
Display display = activity.getWindowManager().getDefaultDisplay();
Point size = new Point();
display.getSize(size);

int itemHeight = size.x/5;
ViewGroup.LayoutParams param = linearLayout.getLayoutParams();
param.height = itemHeight;
linearLayout.setLayoutParams(param);

// set item data on the view
ItemData itemData = arrayList.get(position);
ImageView imageView = linearLayout.findViewById(R.id.imageView);
TextView textViewName = linearLayout.findViewById(R.id.textViewName);
TextView textViewTime = linearLayout.findViewById(R.id.textViewTime);
TextView textViewMessage = linearLayout.findViewById(R.id.textViewMessage);
```

```
imageView.setImageResource(itemData.getImage());  
textViewName.setText(itemData.getName());  
Date date = new Date(itemData.getTime());
```

```
SimpleDateFormat simpleFormat = new SimpleDateFormat("yyyy-MM-dd");  
textViewTime.setText("" + simpleFormat.format(date));  
textViewMessage.setText(itemData.getMessage());
```

```
return linearLayout;
```

```
}
```

```
public void addItem(int rid, String name, String message) {  
    ItemData item = new ItemData(rid, name, System.currentTimeMillis(),  
    message);  
    arrayList.add(item);  
}
```

```
}
```

- 리스트뷰 클래스를 구현
 - 딱히 하는 것은 없음

```
package com.iot.katalklistview;

import android.content.Context;
import android.util.AttributeSet;
import android.widget.ListView;

public class KaTalkListView extends ListView {

    public KaTalkListView(Context context, AttributeSet attrs) {

        super(context, attrs);
    }
}
```

```
package com.iot.katalklistview;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private KaTalkListView listView;
    private KaTalkAdapter adapter;

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

```
listView = (KaTalkListView)findViewById(R.id.listView);

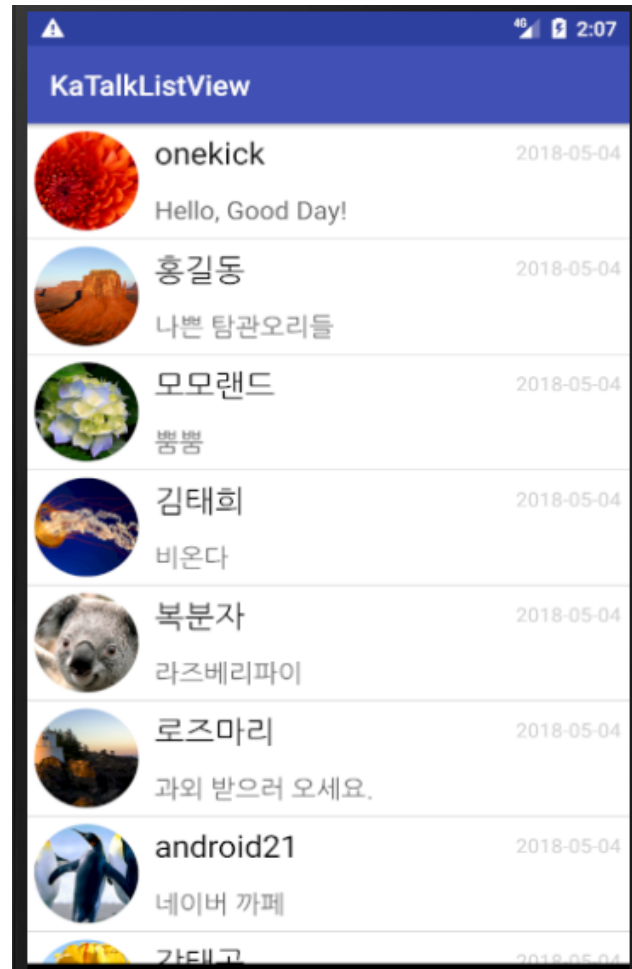
adapter = new KaTalkAdapter(this);

adapter.addItem(R.drawable.image_1, "onekick", "Hello, Good Day!");
adapter.addItem(R.drawable.image_2, "홍길동", "나쁜 탐관오리들");
adapter.addItem(R.drawable.image_3, "모모랜드", "뽐뽐");
adapter.addItem(R.drawable.image_4, "김태희", "비온다");
adapter.addItem(R.drawable.image_5, "복분자", "라즈베리파이");
adapter.addItem(R.drawable.image_6, "로즈마리", "과외 받으러 오세요.");
adapter.addItem(R.drawable.image_7, "android21", "네이버 카페");
adapter.addItem(R.drawable.image_8, "강태공", "낚시하러 갑시다.");

listView.setAdapter(adapter);

adapter.notifyDataSetChanged();           // adapter::getView() -> onDraw();
}
}
```

- 리스튜 뷰를 이용해서 상하로 스크롤해보자.
 - 데이터가 많아도 상당히 스크롤 속도가 빠름
 - 내부적으로 아이템의 메모리는 약 10개정도만 할당
 - 스크롤 할 때 객체를 새로 생성하는 것이 아니라 기존에 생성해 놓은 아이템뷰 재사용



- 오른쪽 그림과 같은 Custom ListView를 구현해보자
 - 이미지1, 텍스트 뷰1, 토글 버튼 1을 가지는 Item
 - ListView에 Item을 25개 등록
- 이렇게 구현하면 과연 아무 문제가 없을까요?
 - 귀신 들린듯이 동작하는 리스트뷰
 - 1번째 아이템의 Toggle버튼을 선택
 - 스크롤 해보면 다른 아이템의 Toggle버튼도 선택되어있다!!
 - 한번 해결해 봅시다.
- Adapter의 getView()는 아이템이 표시될 때마다 자동 호출
 - 이때 view를 재사용하게 될 경우 내용을 모두 새로 채워 넣어 주어야 함.



@Override

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

```
    Log.i(TAG,"getView called");
```

```
    if(convertView==null)  
        생성
```

// 재사용할 아이템뷰가 없는 경우만 새로

```
{
```

```
    Log.i(TAG,"getView: convertView==null");
```

```
    LayoutInflater inflater = activity.getLayoutInflater();
```

```
    convertView = inflater.inflate(R.layout.layout_item, parent,false);
```

```
}
```

```
// view에 모든 값을 새로 넣어 주어야 함.
```

```
}
```