Android基础UI开发

主讲人：字节跳动Android工程师-马梓轩

通过本次课程你能掌握如下技能

- Activity基础
- 了解如何编写View布局
- 掌握高级UI组件ListView的使用
- intent相关

1.Activity 基础

- activity概念
- 创建注册
- 生命周期
- 使用场景

1.1 Activity 概念

❑ 什么是Activity?

Activity是Android的四大组件之一， 是用户操作的可视化界面。 在Android App 中只要能看见的几乎都要依托于Activity，所以Activity是在开发中使用最频繁的一种组件。 一个应用通常是由 Activity 组成。 一般会指定应用中的某个 Activity 为“主”Activity，即首次启动应用时呈现给用户的那个 Activity,而且每个 Activity 均可启动另一个 Activity，以便执行不同的操作。

- 每次新 Activity 启动时，前一 Activity 便会停止，但系统会在堆栈（“返回栈”）中保留该 Activity。

❑ 创建Activity的两种方式

https://developer.android.com/guide/components/activities/?hl=zh-CN

1.1 生命周期

```java
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

1.1.2 生命周期图解

- 为什么要有生命周期?

1.弹窗（处于Partially Visible） 暂停视频播放

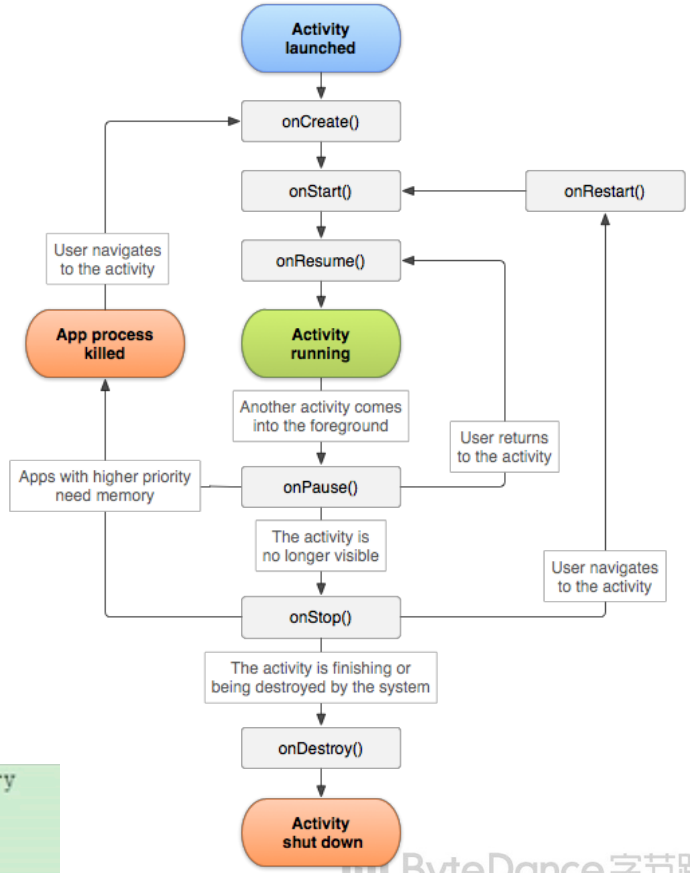
2.home退出抖音，或者从抖音进入了其他APP(比如微博授权，QQ登录)

3.刷抖音 突然来了一个电话，结束后又回到抖音

很多情况下，这些时候，我们自己的界面不处于active状态，那么应该让手机的资源即使释放或者降低优先级，**以确保提供一个流畅的用户体验**

4.释放资源

```
完整生命周期  oncreate--》onstart--》onresume--》onpause--》onstop--》ondestory
可视生命周期  onstart--》onresume--》onpause--》onstop
前台生命周期  onresume--》onpause
```



1.1.3 (了解)屏幕旋转&数据保存

- 屏幕旋转&数据保存
- onSaveInstance的触发时机
- 启动模式/任务栈

https://developer.android.com/guide/components/tasks-and-back-stack.html?hl=zh-CN

android:screenOrientation="portrait"
一般app都默认写死竖直

相关资料：https://www.ktanx.com/blog/p/1283

2 View的入门

- Xml介绍
- 常用控件使用
- 容器 ViewGroup 使用介绍

2.1 XML

- 什么是XML
- 格式是什么样的?
- Android中XML应用场景?
- （了解）Android系统以何种方式解析Manifest?

2.1.1什么是XML

❑ 定义:Extensible Markup Language 可扩展标记语言

W3C组织发布，目前遵循的是W3C组织的2000年发布的XML 1.0规范

现实生活中存在着大量的数据，在这些数据存在一定的关系，我们希望能在计算机中保持和处理这些数据的同时能够保存和处理他们之间的关系。

XML 就是为了解决这样的需求而产生数据存储格式

2.1.2 XML格式

- 一个标签分为开始标签和结束标签，必须成对出现
- 子元素. 父元素 利用嵌套关系表示上下级关系
- 属性

```
<?xml version = “1.0” encoding= “utf-8”?>
<北京  区号=010>
        <朝阳 >
            <潘家园>
            </潘家园>
            <国贸>
            </国贸>
            …
        </朝阳>
        <海淀>
            <知春路>
            </知春路>
            <知春里>
           </知春里>
                                    …
        </海淀>
          ….
</北京>
```

2.1.3 Android中XML应用场景

❑ 传输数据 （网络）

跨平台 XML支持跨平台 跨平台跨跨开发语言 本质字符串
xml使用网络传输使用的不多，现在主流主要是使用json

```
{
  "id": 123,
  "title": "Object Thinking",
  "author": "David West",
  "published": {
    "by": "Microsoft Press",
    "year": 2004
  }
}
```

❑ 配置文件

1.AndroidManifest App的配置文件

2.布局描述

…

```
<?xml version="1.0"?>
<book id="123">
  <title>Object Thinking</title>
  <author>David West</author>
  <published>
    <by>Microsoft Press</by>
    <year>2004</year>
  </published>
</book>
```

2.1.3 Android的Manifest

- 作用:

AndroidManifest.xml清单文件是每个Android程序中必须的文件，它是整个Android程序的全局描述文件，除了能声明程序中的Activities，Content Providers，Services，和Intent Receivers，还能指定应用的名称、使用的图标、包含的组件以及permissions和instrumentation（权限和测试）

- 举例子:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.helloworld">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity"
android:launchMode="standard">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

2.1.3 Manifest常用属性介绍(了解)

各节点解释（顺序从上到下）：

- **xmlns:android**：定义android命名空间，这样使得Android中各种标准属性能在文件中使用，提供了大部分元素中的数据。
- **package**：指定本应用内java主程序包的包名，它也是一个应用进程的默认名称。
- **application**：一个AndroidManifest.xml中必须含有一个Application标签，这个标签声明了每一个应用程序的组件及其属性(如icon、label、permission等)。
- **icon**：这个很简单，就是声明整个APP的图标，图片一般都放在drawable文件夹下。
- **label**：声明整个APP的名字，字符串常量一般都放在values文件夹下的strings.xml里。
- **theme**：是一个资源的风格，它定义了一个默认的主题风格给所有的activity，当然也可以在自己的theme里面去设置它，有点类似style。
- **activity**：定义APP中的一个组件Activity。
- **launchMode**: Activity的启动模式，默认即standard代表标准的
- **name**：该Activity的名字。
- **intent-filter**：意图过滤器，后续会讲到。
- **<action android:name**：指定程序入口Activity，在这里是MainActivity。
- **<category android:name**：指定当前动作（Action）被执行的环境。这里的CATEGORY_LAUNCHER决定应用程序是否显示在程序列表里。

更多的属性参考(不用翻墙)：https://developer.android.com/guide/topics/manifest/manifest-intro?hl=zh-cn

2.1.4 XML解析方式（了解）

- DOM解析 、SAX解析 、PULL解析（Android系统内部采用的就是这种方式）

拓展参考：

1.xml的三种解析方式:https://www.jianshu.com/p/4e6eeec47b27

2.resource in android: https://developer.android.com/guide/topics/resources/providing-resources?hl=zh-cn

2.2 常用控件

- 通用属性
- TextView
- Button
- Imageview

Android 控件介绍：https://developer.android.com/reference/android/widget/package-summary

或者直接看 design

2.2.1 通用属性 (View)

❑ 控件的宽高

match_parent(fill_parent 已经废弃）填充满父布局

wrap_parent

固定宽高 具体大小dp（UI） sp（文字） px（基本不用）

❑ 背景

- android:background=“#ccc”

❑ 间距 margin padding

- android:layout_marigin=“20dp” 不扩大背景
- android:padding=“20dp”。扩大背景
- Top Bottom Left Right 上下左右

❑ ID 用来查找控件

- Android:id= “@+id/txt_id”

❑ 布局是否可见

Visible 可见

inVisible 不显示但是占空间

Gone 隐藏

2.2.1 Dp&Sp

知识点：

- **px**： pixel，像素Android原生API，UI设计计量单位，如获取屏幕宽高。
- **屏幕分辨率**：指在纵向和横向方向上的像素点数，单位是px，一般显示方式是纵向像素数量*横向像素数量，如1920*1080。
- **屏幕尺寸**：一般是屏幕对角线长度，单位是英寸，常见尺寸有3.5，4.0，4.3，4.7，5.0，6.0等。
- **dpi屏幕像素密度**：ppi pixel per inch的缩写，意思是每英寸屏幕上的像素数，因为屏幕尺寸是商家生产时就规定好的，屏幕尺寸一样的手机，屏幕宽高却不一定一样，所以通常取屏幕对角线像素数量和屏幕尺寸（屏幕对角线长度）来计算屏幕像素密度，计算公式就是通过勾股定理和分辨率计算得到屏幕对角线像素数量，再除以屏幕尺寸。手机参数上也会有这个数值。

  屏幕密度计算 以**6.0**英寸 **1920*1080**分辨率为例：

  1920和1080的平方和开根号=对角线像素（勾股定理）

  √（1920^2+1080^2）=2202.9071

  屏幕密度= 对角线像素/对角线长度

  367.1511=2202.9071/6
- **dp /dip**： 一个基于屏幕密度的抽象单位，Android规定160dpi为baseline，其他均以此为基准，如右图1
- **sp**： 同dp相似，但还会根据用户的字体大小偏好来缩放(建议使用sp作为文本的单位，其它用dp)

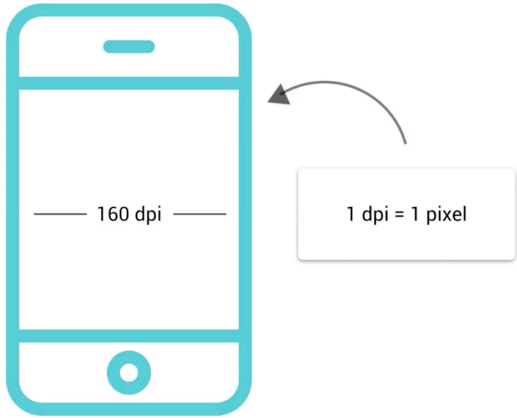


图1

Conversion

px = dp * (dpi / 160)

图2

Tips: 目前市面上大部分常见的手机都是480dpi，所以在这些手机上 1dp = 3px ，转换公式如右图2

2.2.2 TextView

TextView是Android中最常用的控件，主要承担文本显示的任务，任何APP都不可避免的会用到它。

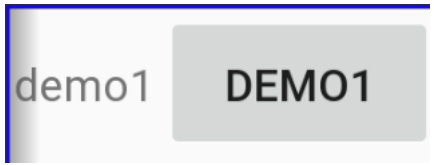
各节点解释（顺序从上到下）：

- **android:text=“今日头条”**
- **android:textColor=“#000” 文字颜色**
- **android:textSize=“18sp” 文字大小**
- **Android:gravity=“center” 空间内内容显示位置**
- **android:singleLine=“true” 文字显示在一行**
- **android:lines="2" 不管多大都显示两行**
- **android:maxLines="2" 超过两行只显示两行**
- **android:ellipsize=“end" 文字省略方式**

更多的属性参考：http://wiki.jikexueyuan.com/project/twenty-four-Scriptures/fun-text-view.html

2.2.2 Button

Button按钮是Android应用中最常用到的控件，说它为交互之王一点都不过分，每个应用中都包含了多个Button响应和解决用户各种点击交互事件



- 继承TextView所以拥有所有TextView的属性

```
public class  Button  extends TextView
```

- 点击事件的设置 (所有控件都具有的属性)

1.布局设置点击事件

```
android:onClick="onClick"
```

2.代码设置点击事件

```
button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
          Toast.makeText(this,"Toast",Toast.LENGTH_SHORT).show();
            }
        });
```

更多的属性参考：https://wiki.jikexueyuan.com/project/twenty-four-Scriptures/button-controls.html

Toast介绍：https://www.twle.cn/l/yufei/android/android-basic-toast.html

2.2.2 ImageView

ImageView介绍：

ImageView展示图片的放大、缩小、旋转等操作。还有一个就是显示网络中下发的图片展示都是靠ImageView的，具体资料可以看参考。

属性介绍：
布局设置：android:src：设置ImageView所显示的Drawable对象的ID。
代码设置：setImageResource

更多的属性参考：https://www.cnblogs.com/plokmju/p/android__ImageView.html

2.3 布局 ViewGroup 使用介绍

- LinearLayout
- RelativeLayout
- View和Viewgroup区别

Android常用布局介绍：https://www.jianshu.com/p/2598626b1b04

2.3.1 LinearLayout

LinearLayout 线性布局

各节点解释（顺序从上到下）：

- **android:orientation="horizontal"**
  指定线性布局的排列方式　水平 **horizontal**　垂直 **vertical**
- **android:layout_gravity="center"** 当前控件在父元素的位置
- **android:layout_weight="1"** 权重 按比例划分 和**"0"**一起使用
  其实在这里linearlayout会优先使用我们声明的值进行分配，然后再将剩下的尺寸按照weight进行分配
- **android:layout_weightsum="2"**

更多的属性参考：https://www.runoob.com/w3cnote/android-tutorial-linearlayout.html

https://www.jianshu.com/p/002e69dfcedd

2.3.2 RelativeLayout相对布局

但是使用LinearLayout的时候也有一个问题，就是当界面比较复杂的时候，需要嵌套多层的 LinearLayout,另外太多层LinearLayout嵌套会占用更多的系统资源;但是如果我们使用RelativeLayout的话,可能仅仅需要一层就可以完成了。

- **android:layout_toRightOf 在指定控件的右边 Left**
- **android:layout_above 在指定控件的上边 below**
- **android:layout_alignBaseLine 跟指定控件水平对齐**
- **Android:layout_alignLeft 跟指定控件左对齐 Right Top Bottom**
- **android:layout_toleftof toRightof**
- **android:layout_alignParentLeft 是否跟父布局左对齐 Top Right Bottom**
- **android:layout_centerVertical 在父布局中垂直居中**
- **android:layout_centerHorizontal 在父布局中水平居中**
- **android:layout_centerParent 在父布局中居中**

更多的属性参考：http://wiki.jikexueyuan.com/project/twenty-four-Scriptures/fun-text-view.html

2.3.3 View&ViewGroup

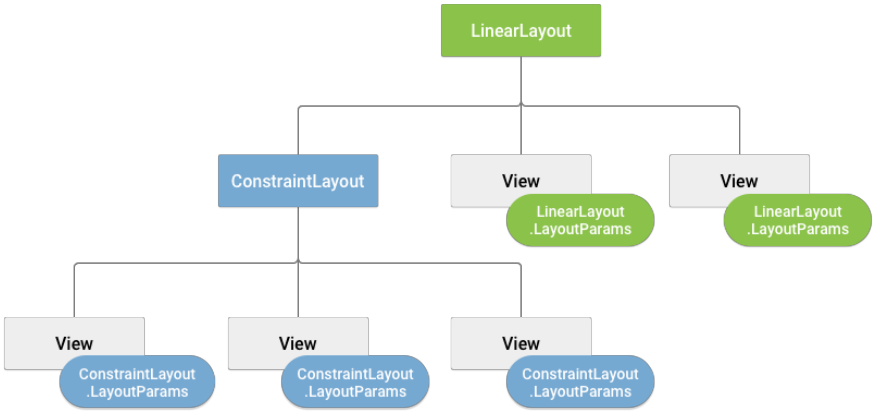
- View介绍

是所有控件的基类

- ViewGroup介绍

View的容器， ViewGroup extends View

- Attribute(属性)&LayoutParams(布局参数)
- 在UI上两者的关系：呈现一个树性结构




```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/
res/android"
            android:layout_width=“match_parent"
            android:layout_height=“match_parent”
            android:orientation="vertical" >
   <TextView android:id="@+id/text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="I am a TextView" />
   <Button android:id="@+id/button"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="I am a Button" />
</LinearLayout>
```

课后题1

2.3.4 使用 Layout Editor 构建 UI

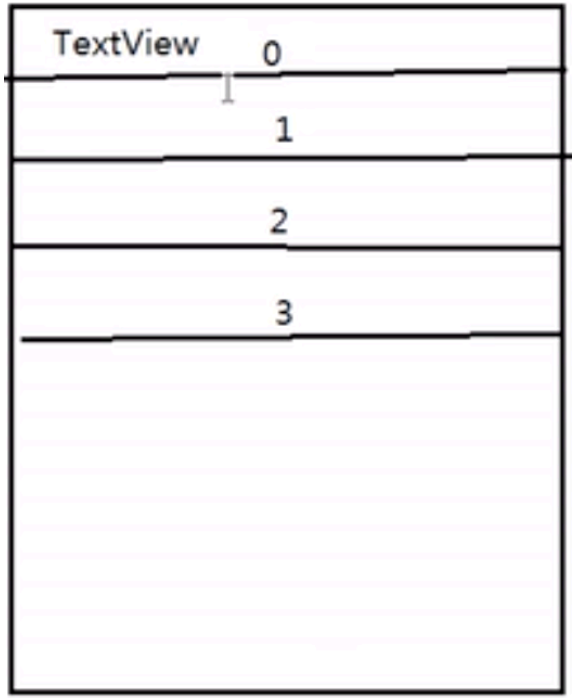
- 使用布局来写
- 直接在Layout Editor 拖拽
- 使用代码直接写

3 如何实现一个页面显示0-100

- Linearlayout?
- Relativelayout?

如果是服务端下发的呢？列表控件

- ListView 初级
- RecyclerView 高级



3.1.1 利用ListView实现上面的页面

- listview
- 数据
- Adapter 数据和listview连接器，里面决定显示多少个，显示什么内容

Public Int getCount（） 主要是返回有多少个Item

public View getView(int position, View convertView, ViewGroup container) {
}
实现主要目的是返回listview所要显示的item, 参数：position

https://developer.android.com/reference/android/widget/ListView

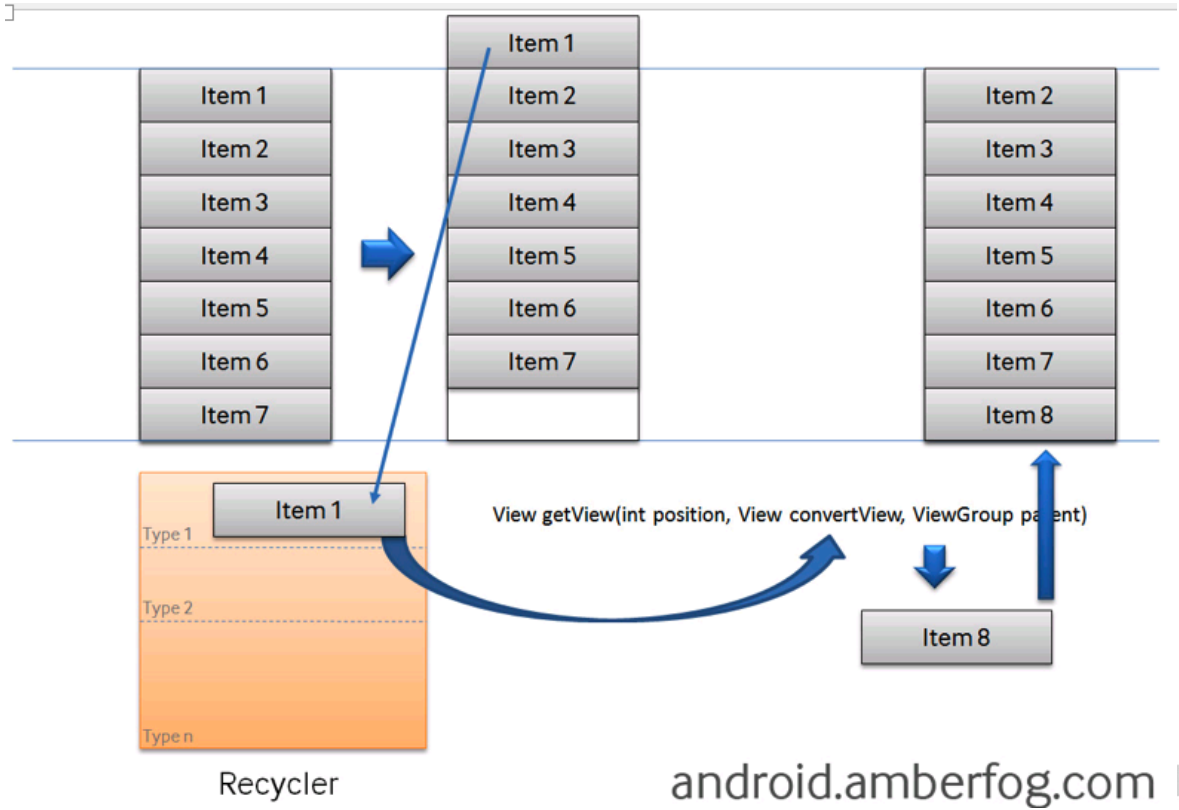
3.1.2 Listview中如何加载布局xml?

什么是LayoutInflater
LayoutInflater是一个用于将xml布局文件加载为View或者ViewGroup对象的工具，我们可以称之为布局加载器。

```
LayoutInflater inflater = LayoutInflater.from(mContext);
View view = inflater.inflate(R.layout.activity_main, null);
```

https://www.jianshu.com/p/81a698aaf05c

3.1.3 ListView缓存复用机制



3.1.3 ListView缓存复用机制

```
@Override public View getView(int position, View convertView, ViewGroup parent) {
    TextView textView;
    if (convertView == null) {
        textView = new TextView(mContext);
    } else {
        textView = (TextView) convertView;
    }

    textView.setText(" " + args[position]);
    textView.setTextSize(18);
    return textView;
}
```

3.1.4 点击事件

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText( context: MainActivity.this,  text: "点击了第: " + position, Toast.LENGTH_LONG).show()
    }
});
```

View: 被点击的item对象

Position: 点击的是第几个item

Id: getItemId 返回的id

4.1 Intent

- 作用&基础属性
- 类型：显式&隐式
- startActivityForResult (了解)

4.1.1 Intent的作用和基础属性

- 作用：解决Android的组件之间的通讯(应用内和应用外)

1.启动Activity、启动Service、传递Broadcast

- 七大属性：

1. ComponentName 标识唯一性 （包名+类名全称）
2. Action （一个普通字符串，代表intent要完成的一个动作，最后由intent-filter来筛选）自己的意图
3. Category （为Action提供额外的附加类别信息，两者通常结合使用）用来表现动作的类别
4. Data 它们构成了隐式启动的匹配项，通过不同的配置完成不同的路由跳转。表示与动作要操纵的数据
5. Extra(一个bundle对象，数据存储的拓展)

参考：https://developer.android.com/guide/components/intents-filters?hl=zh-cn

https://www.cnblogs.com/smyhvae/p/3959204.html
https://blog.csdn.net/franky814/article/details/81868387#data%E6%A0%87%E7%AD%BE

4.1.3 显示Intent

显示intent(明确了要指定启动的Component)
1.启动Activity、启动Service、传递Broadcast
举例：startActivity(A.class,B.class)

```
Intent intent = new Intent(IntentActivity.this, OpenedIntentActivity.class);
//传入context, 要打开的Activity
intent.putExtra("data", “我是IntentActivity");
//要传输的数据 的”代号“ 和数据本身
startActivity(intent);
```

被打开的Activity

```
String data = getIntent().getStringExtra("data");
```

常见系统Action请参考：https://developer.android.com/reference/android/content/Intent

4.1.2 隐式intent

- 隐式intent(没有明确的指定要启动哪个Component)
- 多用在多APP之间调用

主要是通过比较Intent对象内容与Intent-filter过滤器来实现，主要匹配三个属性：action、category、data

举例:

```
<intent-filter>
    <action android:name="com.wy.demo.TEST_ACTION"/>  //自定义的Action
    <category android:name="com.intent.category.DEFAULT"/> //系统的category
    <data android:mimeType="video/mpeg" android:schema="http://xxx.xxx."/>
```

```
Intent it = new Intent();
it.setAction(Intent.ACTION_VIEW);
it.setData(Uri.parse("http://www.baidu.com"));
startActivity(it);
Case 2:
Intent intent = new Intent(Intent.ACTION_DIAL);
```

常见系统Action请参考：https://developer.android.com/reference/android/content/Intent

4.1.3 Intent 之startActivityForResult

- 作用：当启动一个新的Activity之后需要有返回值

举例(使用场景)：

1.选择联系人

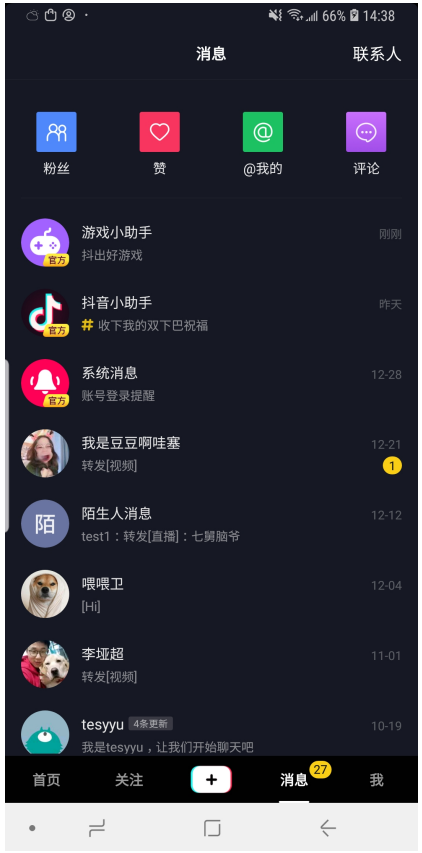
2.仿微信授权

常见系统Action请参考：https://developer.android.com/reference/android/content/Intent

https://developer.android.com/training/basics/intents/result#java

4 课后作业

- Exercises1：打印出Activity屏幕切换 Activity会执行什么生命周期?
- Exercises2：一个抖音笔试题：统计页面所有view的个数　ViewGroup中的API：　getChildCount()　getChildAt()
- Exercises3:实现一个抖音消息页面，如右图-> ,并且点击每个listview的item,能够跳转到一个新的界面，并且在新的页面显示出他是第几个item (也可以用recyclerview) UI样式类似即可，第一行平均分居中，底下的文字左对齐 文字图片间距都是5dp，文字16sp. 14sp　一行 60dp ,上面四栏，点击进入新页面，中间显示点击按钮，点击弹出toast显示他前一个页面点击按钮上的文字
- （选择完成）Exercises4:ABCD是四个Activity A的启动模式是Standard B是SingleTop C是 SingleTask D是SingleInstance ，用户的启动顺序是ABCD ABCD AB,问1.最后启动到B的时候，当前Activity栈的情况，2.如果用户一致点返回键，退栈顺序，最好能够写上原因

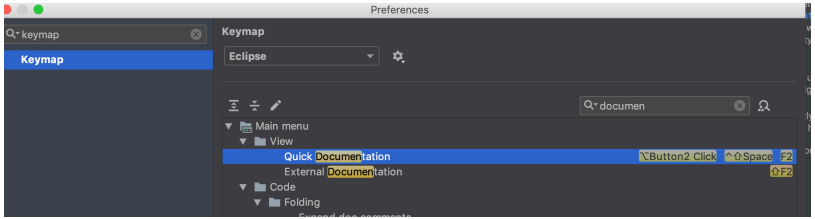


4 课后作业

匿名内部类成员变量https://www.zhihu.com/question/21395848

Listview recyclerview 缓存对比：https://segmentfault.com/a/1190000007331249

小技巧：https://blog.csdn.net/qq137722697/article/details/74085789



快捷键

打开documentation ctrl+Q获取文档

搜索文件 ctrl+shift+a

前进后退 alt 左右

补全alt + 回车

https://blog.csdn.net/s1674521/article/details/78769936

THANKS

.

联系方式

Email: mazixuan@bytedance.com

WeChat: ma2650118