

Android基础UI开发

主讲人：字节跳动Android工程师-王瀚林



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RecyclerView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

```
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?

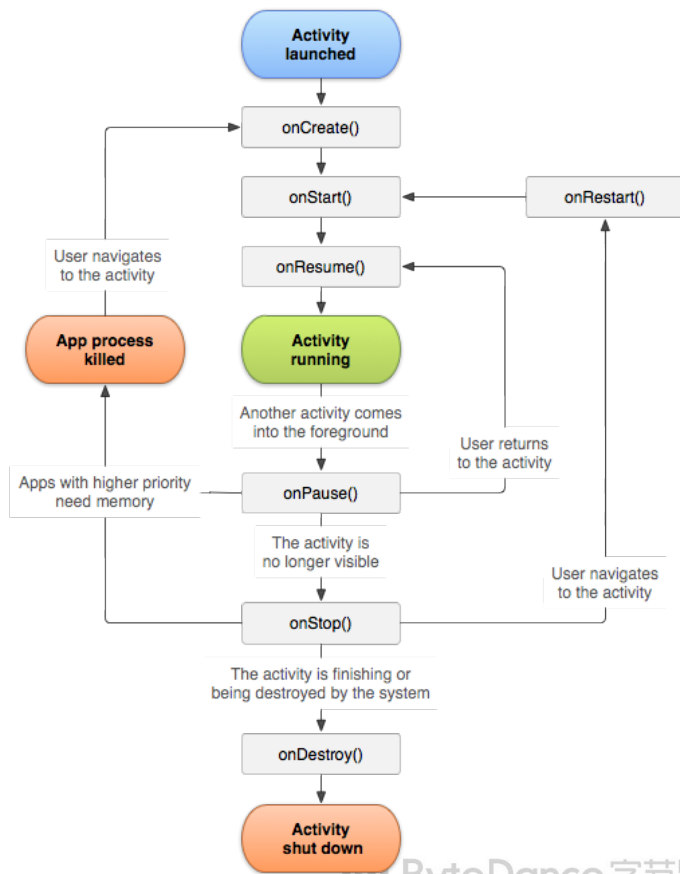
1.弹窗（处于Partially Visible）

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





1.1.3 屏幕旋转&数据保存

- ❑ 屏幕旋转&数据保存
- ❑ onSaveInstanceState的触发时机



2 View的入门

- ❑ Xml介绍
- ❑ View和ViewGroup 布局控件
- ❑ 常用控件使用
- ❑ 使用 Layout Editor 构建 UI



1.3 XML

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

1.3.1什么是XML

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

❑ 格式举例:

```
<messages>
  <message>
    <title isOfficial="true">抖音小助手</title>
    <hashtag>#收下我的双下巴祝福</hashtag>
    <time>昨天</time>
    <icon>小助手.jpg</icon>
  </message>
  ...
</messages>
```



❑ 作用: 1.跨平台数据传输 2.配置文件

1.3.2 XML解析方式

- ❑ DOM解析、SAX解析、PULL解析 (Android系统内部采用的就是这种方式)
- ❑ Android的应用：Manifest清单文件配置,界面布局编写，各种resource的编写都是用得上的 (后期课程中会讲到)

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>

1.4 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>
```

1.4.1 Manifest常用属性介绍

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 View&ViewGroup

□ View介绍

是所有控件的基类

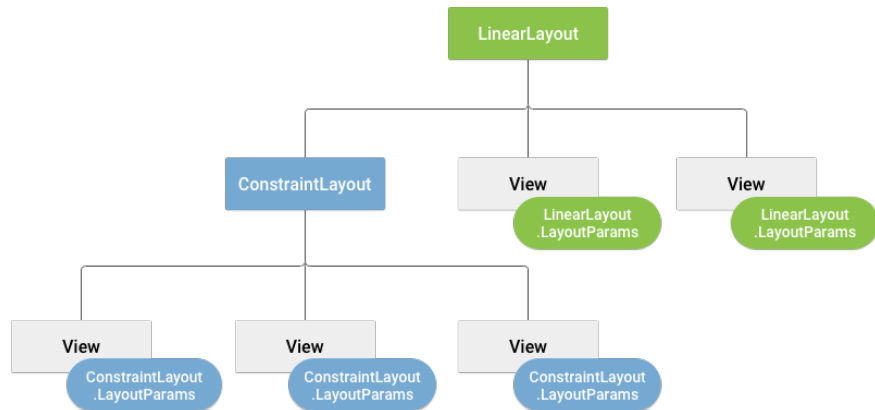
□ ViewGroup介绍

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>
```



2.2 常用控件

❑ UI布局

LinearLayout 、 RelativeLayout、 FrameLayout

❑ UI控件

TextView、 ImageView、 Button、 ProgressBar、 ScrollView、 Toast

想看一共有哪些控件可以看design

参考: <http://hukai.me/android-training-course-in-chinese/basics/firstapp/building-ui.html>

2.2 宽高

- ❑ `match_parent`
- ❑ `wrap_parent`
- ❑ 具体大小 `dp` `sp` `px`

参考: <http://hukai.me/android-training-course-in-chinese/basics/firstapp/building-ui.html>

2.3 Dp&Sp px

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）

$$\sqrt{(1920^2 + 1080^2)} = 2202.9071$$

屏幕密度= 对角线像素/对角线长度

$$2202.9071 / 6 = 367.1511 \approx 367 \text{ dpi}$$

- ❑ **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)

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

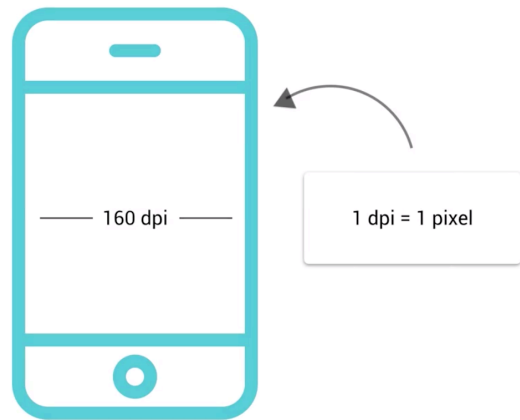


图1

Conversion

$$\text{px} = \text{dp} * (\text{dpi} / 160)$$

图2



2.4 使用 Layout Editor 构建 UI

□ 写界面的两种方式



3 高级组件RecyclerView

- ❑ 实现一个“抖音消息页面为什么要使用RecyclerView?”
- ❑ RecyclerView工作原理
- ❑ RecyclerView的基本使用
- ❑ ViewHolder和Adapter的作用
- ❑ LayoutManager布局管理器
- ❑ 对item进行添加点击响应
- ❑ 拓展

3.1 Why RecyclerView ?

❑ 之前学过的知识想做出一个抖音的消息页面可以吗？

答：可以，如何做？（ScrollView+N个LinearLayout。。。）

❑ 上面的做法有何不妥？

答：数据过多时，内存有限，性能欠佳，直观感受->滑动很卡,严重丢帧！！，
甚至崩溃

❑ 更优解？

RecyclerView 应运而生

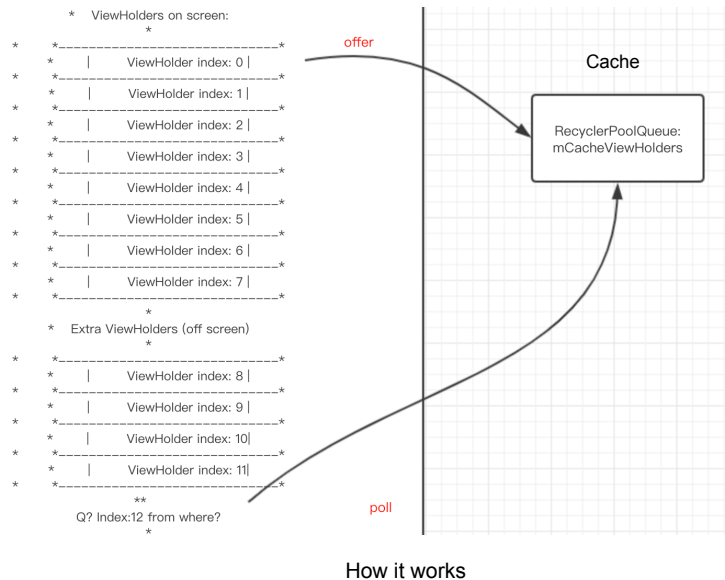
3.2 RecyclerView 定义&工作原理

❑ 什么是 RecyclerView?

1. ViewGroup to display large set of data
2. For each item in large dataset, it display a view
3. Efficient way to implements Scrollable List

❑ 工作原理?

1. 直接上图



3.3 RecyclerView 基本使用

❑ Steps (参考官方的步骤)

1. Add the support library to gradle build file
2. Add recycle-view in xml
3. Specify the LayoutManager type
4. Initialize it inside the Activity or Fragment
5. Create a layout file for a single item inside the RecyclerView
6. Create an adapter that describes how to display the data for each item
 1. DataSource 【db、xml、json、memory】
 2. Number of items
 3. Use the onCreateViewHolder to link the Adapter with layout from 4
 4. Use the onBindViewHolder to show data inside the Adapter for each position
7. RecyclerView setAdapter

3.4 ViewHolder&Adapter 的作用

- ❑ 为什么ViewHolder 必须继承至RecyclerView.ViewHolder?

原因是因为RecyclerView 内部的缓存结是直接缓存一个 ViewHolder

- ❑ ViewHolder作用：减少findViewById的操作，该操作相当昂贵，或者当视图结构特别复杂的时候更加有效
- ❑ Adapter的作用：这是一个对象适配器模式，可以用依赖反转的思想来理解

1、RecyclerView本身

2、自己编写的itemView

3、自己定义的Adapter：

如何理解上述三者关系呢？

直观设计的话RecyclerView本身是和itemView是一个耦合(依赖)的关系，且编写view的类型也不确定，那么如何解决呢？可以通过一个抽象类来反转，变成来依赖adapter，最终两者完全解耦，也方便了itemView的可变性，如右图代码描述：

```
public class CustomListView extends ViewGroup {

    public CustomListView(Context context) {
        super(context);
    }

    @Override
    protected void onLayout(boolean changed, int l, int t, int r, int b) {
    }

    /**
     * 方式1: 之间设置
     * 面临的问题, itemView和 CustomListView 严重耦合, 且数据类型也必须明确,
     */
    /**
     * @param messageList
     */
    public void setDataSource(List<Message> messageList) {
        for (int i = 0; i < messageList.size(); i++) {
            LayoutInflater mInflater = LayoutInflater.from(getContext());
            View itemView = mInflater.inflate(R.layout.support_simple_spinner_dropdown_item, root: this, attachToRoot: false);
            addView(itemView);
        }
    }

    /**
     * 方式2, 通过抽象的方式, 依赖倒置, itemView和CustomListView 彻底解耦
     */
    abstract class AbstractAdapter {

        abstract View getView(int position);

        abstract int getViewCounts();
    }

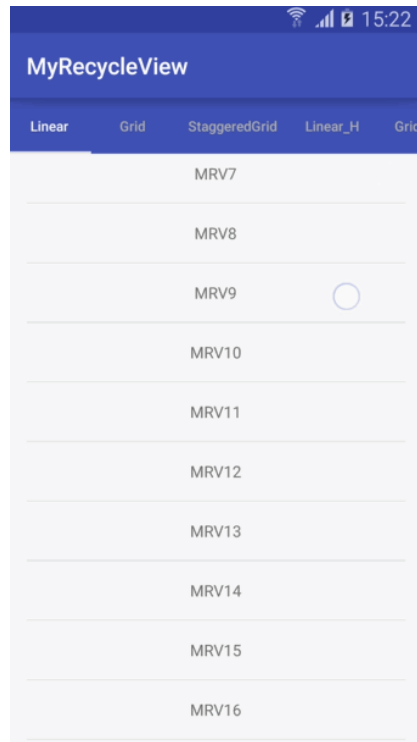
    private AbstractAdapter mAdapter;

    public void setAdapter(AbstractAdapter adapter) {
        mAdapter = adapter;

        //add views
        for (int i = 0; i < adapter.getViewCounts(); i++) {
            View child = mAdapter.getView(i);
            addView(child);
        }
    }
}
```

3.5 LayoutManager 布局管理器

- ❑ 已有类别：LinearLayoutManager、
GridLayoutManager、StaggeredGridLayoutManager
- ❑ 优势：将布局onMeasure onLayout这块暴露出来，由
LayoutManager来处理，设计比较模块化
与ListView相比，可以轻松切换布局，实现各种效果，
右gif所示：



3.6 为item添加点击事件

❑ RecyclerView 没有现成的setOnItemClickListener方法, 我们需要为其添加

1. 定义一个

```
public interface ListItemClickListener {  
    void onItemClick(int position);  
}
```

2. 在adapter中对外暴露set ListItemClickListener的方法

3. ViewHolder中

```
itemView.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        //通过代理  
        mListener.onListItemClick(getAdapterPosition())  
    }  
});
```

4. 在外部持有adapter的地方, mAdapter.setListItemClickListener(new ListItemClickListener(){

```
    public void onItemClick(int position) {  
        //do your want  
    }  
})
```

3.6 拓展

- ❑ ItemAnimator 动画
- ❑ ItemDecoration 分割线
- ❑ SnapHelper 滑动效果

参考: <https://cloud.tencent.com/developer/article/1147120>



1.2 Intent

- ❑ 作用&基础属性
- ❑ 类型： 显式&隐式
- ❑ startActivityForResult

1.2.1 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）

3. Category（为Action提供额外的附加类别信息，两者通常结合使用）

4. Data&Type(MIME类型)

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>

1.2.2 Intent的类型

❑ 显示intent(明确要指定启动的Component)

1.启动Activity、启动Service、传递Broadcast

举例: `startActivity(A.class,B.class)`

❑ 隐式intent(没有明确的指定要启动哪个Component)

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-filter>`

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

1.2.3 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

4 课后作业

- ❑ Exercises1: Logcat在屏幕旋转的时候 #onStop() #onDestroy()会展示出来, 但UI界面我们看不到, 在今天课程基础上想办法补全它, 让其跟logcat的展示一样
- ❑ 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recycleview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



THANKS

■



联系方式

Email: wanghanlin.gundam@bytedance.com

WeChat: hano65535