

Android基础UI开发

主讲人：字节跳动Android工程师-王燚



通过本次课程你能掌握如下技能

- ❑ Activity基础
- ❑ 页面编写View布局
- ❑ 常用UI组件的使用
- ❑ 高级UI组件RecyclerView



1.Activity

- ❑ 生命周期
- ❑ Intent
- ❑ 了解XML
- ❑ Manifest

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；它为用户提供了一个完成操作指令的窗口。当我们创建完毕Activity之后，需要调用setContentView()方法来完成界面的显示；以此来为用户提供交互的入口。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

❑ 为什么要有生命周期?

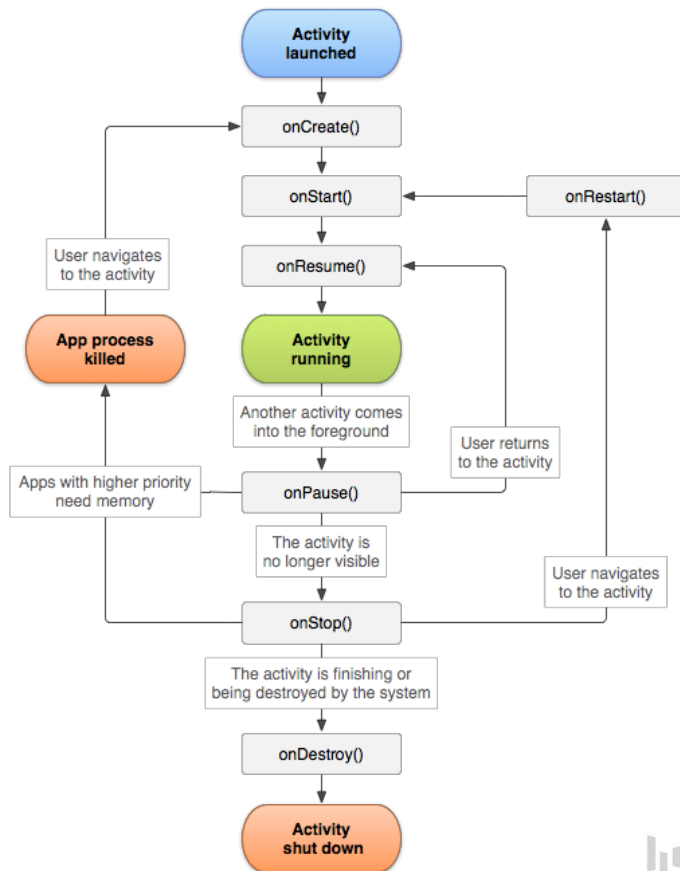
移动端的设备硬件条件有限，内存也比较吃紧，为了流畅的用户体验，于是生命周期诞生了，按场景举例子

- 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

1.1.2 生命周期图解

- ❑ 官方图解
- ❑ 代码演示生命周期何时触发



1.1.3 屏幕旋转&数据保存

❑ 屏幕旋转&数据保存

- 1.生命周期发生了什么样的变化?
- 2.出现了什么问题?

❑ onSaveInstanceState

- 1.触发时机
- 2.如何使用?



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

❑ 几个重要的属性：

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>`

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



1.3XML

- ❑ 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

❑ 用途: XML设计用来传送及携带数据信息

❑ 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>
```



1.3.2XML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)
- ❑ 主要代码举例PULL解析方式，其他2种，有兴趣的童鞋自行Google或参考下面链接

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>
```



2 View的入门

- ❑ View和ViewGroup
- ❑ 常用控件
- ❑ DP,SP

2.1 View&ViewGroup

- View介绍

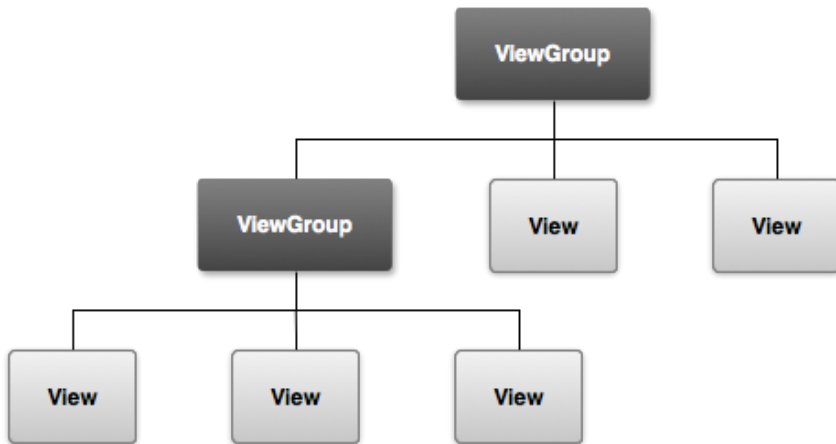
是所有控件的基类

- 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



2.1 View&ViewGroup

□ 代码展示

```
<?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

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

2.3 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**: 一英寸长的直线上的像素点的数量, 即像素密度, 不同的设备, dpi值不同
- ❑ **dp**: 一个基于屏幕密度的抽象单位, 与屏幕密度无关, 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

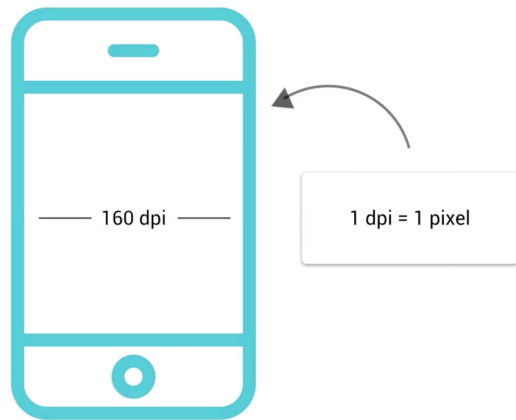


图1

Conversion

$$px = dp * (dpi / 160)$$

图2



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（可选）

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?

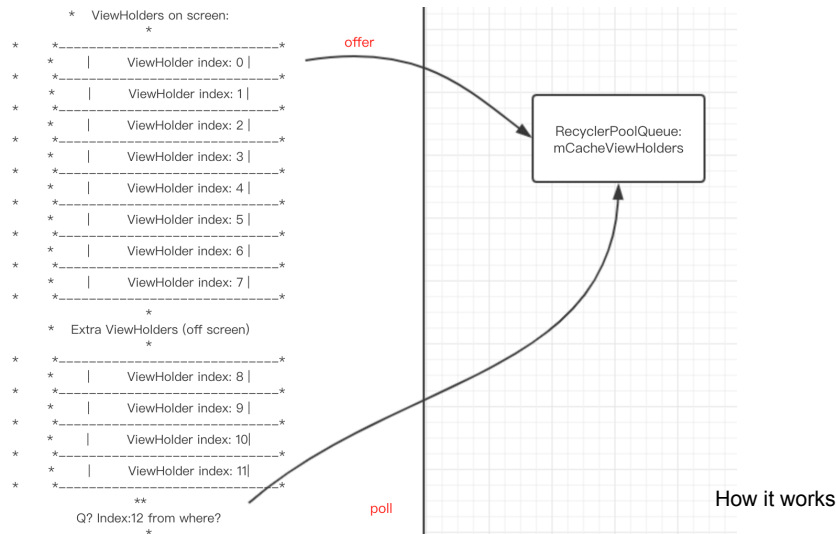
RecyclerView 用于显示大量数据

每个 item 都是一个 view 用于展示数据源

高效展示可滑动的列表控件

工作原理?

右图->



Cache

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();

    }

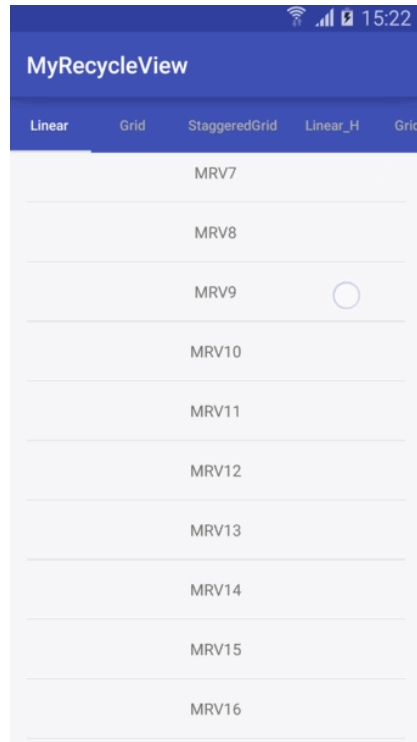
    private AbstractAdapter mAdapter;

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

        //add views
        for (int i = 0; i < mAdapter.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setItemClickListener()的方法

3. ViewHolder中

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

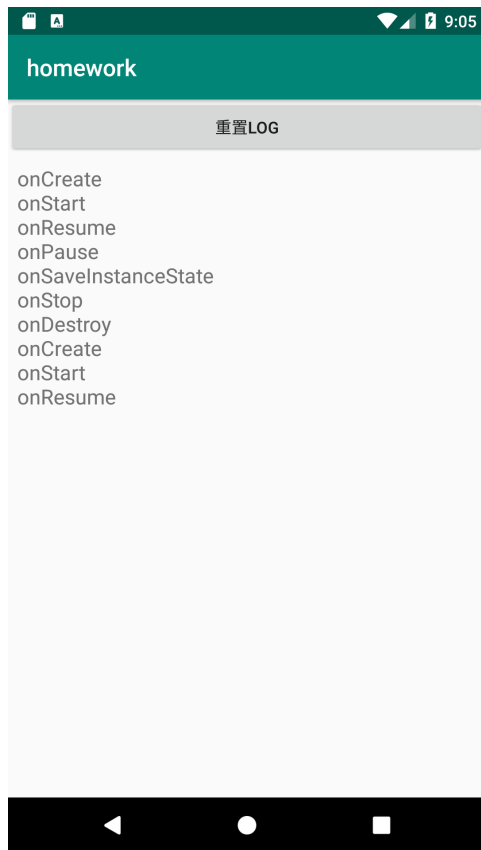
4. 在外部持有adapter的地方, mAdapter.setOnItemClickListener(new ListItemClickListener){

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

4.1 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`SaveInstanceStateActivity`基础上想办法补全它，让其跟Logcat的展示一样,最终效果如图：



4.2 课后作业

- ❑ Exercises2: 一个抖音笔试题: 统计页面所有view的个数

效果如图:



4.3 课后作业

□ Exercises3: 实现一个抖音消息页面, 效果如图:



THANKS



<https://github.com/bytedance-android/chapter1>

联系方式

Email: wangyi.huohuo@bytedance.com (发作业到这里)

WeChat: huohuo397717749