Android Study Plan

deepwaterooo

2021 年 12 月 19 日

目录

1 View 相关 **1**
1.1 View 工作流程 . . . 1
1.2 事件分发 . . . 2
1.3 自定义 View!! . . . 2
1.4 自定义 View . . . 2
1.4.1 一、View 中关于四个构造函数参数 . . . 2
1.5 二、.xml 中的自定义属性 . . . 3
1.5.1 color：引用颜色 . . . 3
1.5.2 dimension: 引用字体大小 . . . 3
1.5.3 enum：枚举值 . . . 3
1.5.4 flags: 标志（位或运行）主要作用 = 可以多个值 . . . 3
1.5.5 raction: 百分数： . . . 4
1.5.6 reference: 参考/引用某一资源 ID . . . 4
1.5.7 混合类型：属性定义时指定多种类型值 . . . 4

1 View 相关

1.1 View 工作流程

- 通过 SetContentView()，调用到 PhoneWindow，后实例 DecorView，通过 LoadXmlResourceParser() 进行 IO 操作解析 xml 文件通过反射创建出 View，并将 View 绘制在 DecorView 上，这里的绘制则交给了 ViewRootImpl 来完成，通过 performTraversals() 触发绘制流程，performMeasure 方法获取 View 的尺寸，performLayout 方法获取 View 的位置，然后通过 performDraw 方法遍历 View 进行绘制。

1.2 事件分发

- 一个 MotionEvent 产生后，按 Activity -> Window -> DecorView（ViewGroup）-> View 顺序传递，View 传递过程就是事件分发，因为开发过程中存在事件冲突，所以需要熟悉流程：
  - dispatchTouchEvent：用于分发事件，只要接受到点击事件就会被调用，返回结果表示是否消耗了当前事件
  - onInterceptTouchEvent：用于判断是否拦截事件（只有 ViewGroup 中存在），当 ViewGroup 确定要拦截事件后，该事件序列都不会再触发调用此 ViewGroup 的 onIntercept
  - onTouchEvent：用于处理事件，返回结果表示是否处理了当前事件，未处理则传递给父容器处理。（事件顺序是：OnTouchListener -> OnTouchEvent -> OnClick）

1.3 自定义 View!!

准备自定义 View 方面的面试最简单的方法：

就是自己动手实现几个 View（由简单到复杂）；分析一些热门 App 中的自定义 View 的效果是怎么实现的；阿里面试官：自定义 View 跟绘制流程相关知识点？（标准参考解答，值得收藏）

- https://www.cnblogs.com/Android-Alvin/p/12297933.html

1.4 自定义 View

1.4.1 一、View 中关于四个构造函数参数

- 自定义 View 中 View 的构造函数有四个

```
// 主要是在 java 代码中生命一个 View 时所调用，没有任何参数，一个空的 View 对象
public ChildrenView(Context context) {
    super(context);
}
// 在布局文件中使用该自定义 view 的时候会调用到，一般会调用到该方法
public ChildrenView(Context context, AttributeSet attrs) { // AttributeSet from .xml 设置
    this(context, attrs, 0);
}

// 如果你不需要 View 随着主题变化而变化，则上面两个构造函数就可以了
// 下面两个是与主题相关的构造函数
public ChildrenView(Context context, AttributeSet attrs, int defStyleAttr) {
    this(context, attrs, defStyleAttr, 0);
}
public ChildrenView(Context context, AttributeSet attrs, int defStyleAttr, int defStyleRes) {
    super(context, attrs, defStyleAttr, defStyleRes);
}
```

- 构造函数的传入参数说明
  - context: 上下文
  - AttributeSet attrs：从 xml 中定义的参数
  - intdefStyleAttr：主题中优先级最高的属性
  - intdefStyleRes：优先级次之的内置于 View 的 style(这里就是自定义 View 设置样式的地方)
- 多个地方定义属性,优先级排序 Xml 直接定义 > xml 中 style 引用 > defStyleAttr>defStyleRes > theme 直接定义

1.5 二、.xml 中的自定义属性

- 基本类型包括：integer, boolean, color, string, float, dimension, enum, flags, fraction, reference
- 基本类型略过，其它相对重要一点儿的

1.5.1 color ：引用颜色

1.5.2 dimension: 引用字体大小

- 定义

```
<attr name = "text_size" format = "dimension" />
```

- 使用：

```
<app:text_size = "28sp"  />
<app:text_size = "@android:dimen/app_icon_size" />
```

1.5.3 enum：枚举值

- 定义

```
<attr name="orientation">
  <enum name="horizontal" value="0" />
  <enum name="vertical" value="1" />
</attr>
```

- 使用：

```
<app:orientation = "vertical" />
```

1.5.4 flags: 标志（位或运行）主要作用 = 可以多个值

- 定义

```
<attr name="gravity">
  <flag name="top" value="0x01" />
  <flag name="bottom" value="0x02" />
  <flag name="left" value="0x04" />
  <flag name="right" value="0x08" />
  <flag name="center_vertical" value="0x16" />
</attr>
```

- 使用

```
<app:gravity = Top|leftf />
```

1.5.5 raction: 百分数：

- 定义：

```
<attr name = "transparency" format = "fraction" />
```

- 使用：

```
<app:transparency = "80%"  />
```

1.5.6 reference: 参考/引用某一资源 ID

- 定义：

```
<attr name="leftIcon" format="reference" />
```

- 使用：

```
<app:leftIcon = "@drawable/图片 ID" />
```

1.5.7 混合类型：属性定义时指定多种类型值

- 属性定义

```
<attr name = "background" format = "reference|color" />
```

- 使用

```
<android:background = "@drawable/图片 ID"  />
<android:background = "#FFFFFF"  />
```