

创建活动

2018年3月25日 17:30

活动Activity

活动是一种可以包含用户界面的组件，主要用于与用户进行交互。一个程序可以包括零个、一个或多个活动。零个的很少。

创建活动Create Activities

准备活动：

如果当前有项目File → close project

新建一个项目 包名默认值com.example.activitytest 不再选择Empty Activity选项，选择Add No Activity（选择活动界面第一个什么都没有的那个），然后Finish

手动创建活动：

找到app/src/main/java/com.example.activitytest（应该是空目录）

（as会使用默认的Android结构，需要手动改成Project结构，以后都这么做）右击com.example.activitytest包，New → Activity → Empty Activity →（对话框），命名FirstActivity，并且不要勾选Generate Layout File 和 Launcher Activity

tip：勾选Generate Layout File作用是自动为FirstActivity创建布局文件。

勾选Launcher Activity作用是把FirstActivity作为当前项目的主活动。

勾选Backwards Compatibility作用是为项目启用向下兼容的模式（要勾上）
其他没有勾上的为手动操作的项目。

注意：任何项目都要重写Activity的onCreate方法，as自动帮我们完成了。

代码如下：

```
public class FirstActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
  
}
```

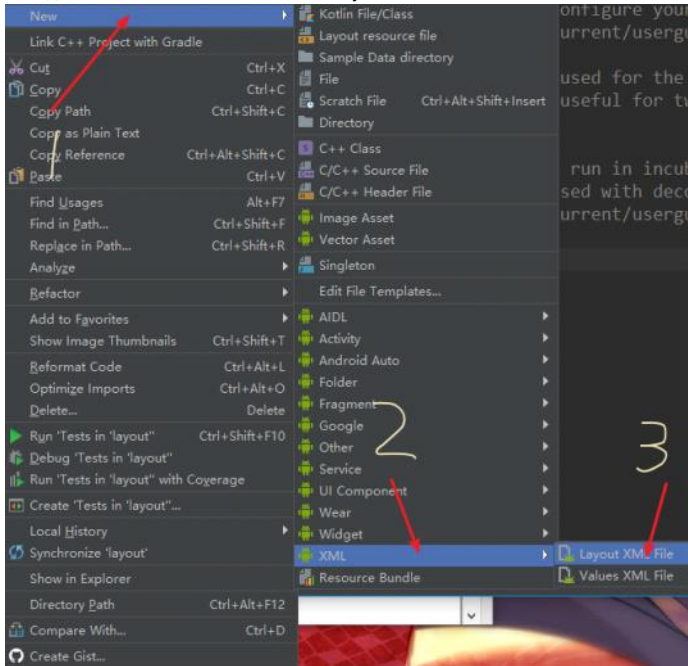
分析这个onCreate方法可以发现，这个方法很简单，只是调用了父类的onCreate方法。这是默认的实现，后面我们可能会加入自己的逻辑。

创建和加载布局

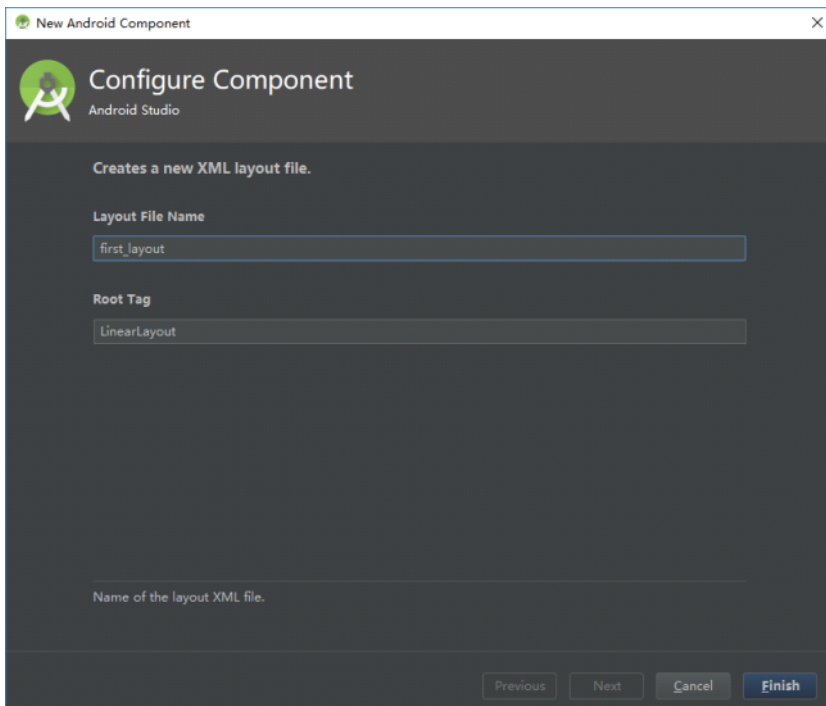
Android的设计讲究逻辑和视图分离，最好每一个活动对应一个布局。布局就是用来显示界面内容的。

下面手动创建一个布局文件。

右击app/src/main/res 目录 → New → XML → layout XML file ，

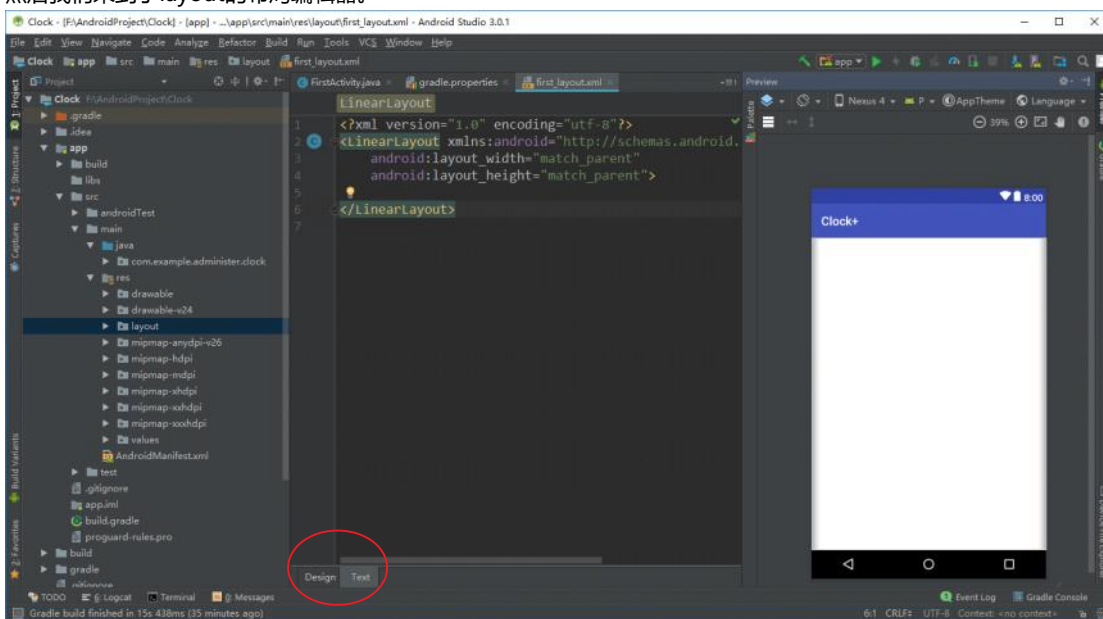


弹出一个新建文件的对话框：



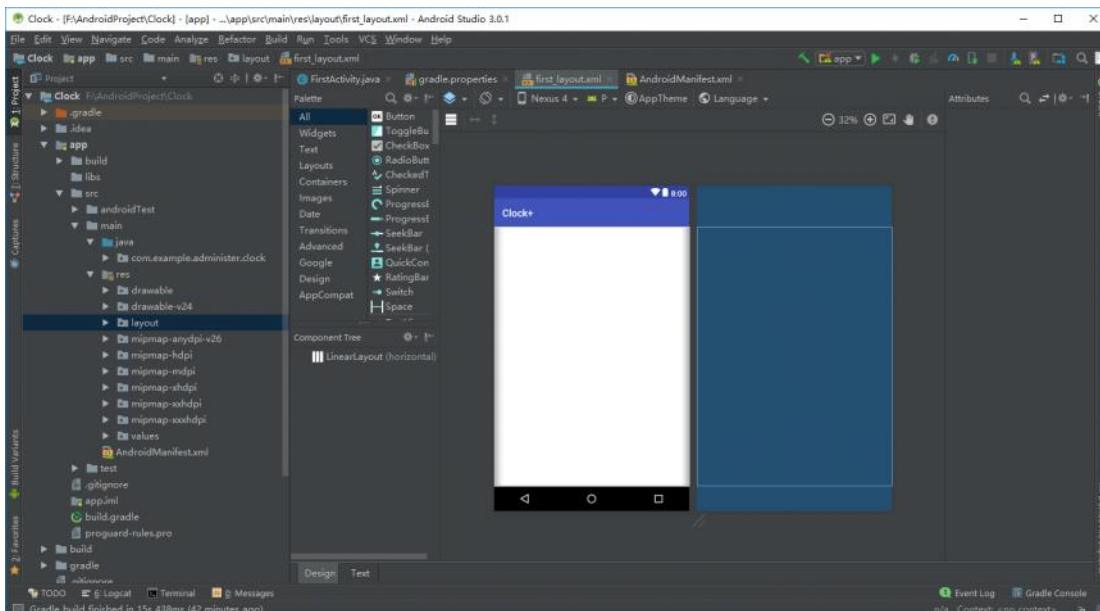
我们建一个名字为first_layout的文件，默认Root tag为Linear Layout类型，注意文件名必须用小写字母加合法的符号，大写字母也不可以。点Finish完成创建。as 会自动帮我们新建一个Layout目录并把我们的layout文件放进去。在以前的版本需要手动创建layout目录。另外layout目录也分为很多种，可以手动创建。

然后我们来到了layout的布局编辑器。



这是Text编辑视图，就是用代码写界面

注意红色圆圈里面的两个选项，点Design可以切换到可视化布局编辑器。如下

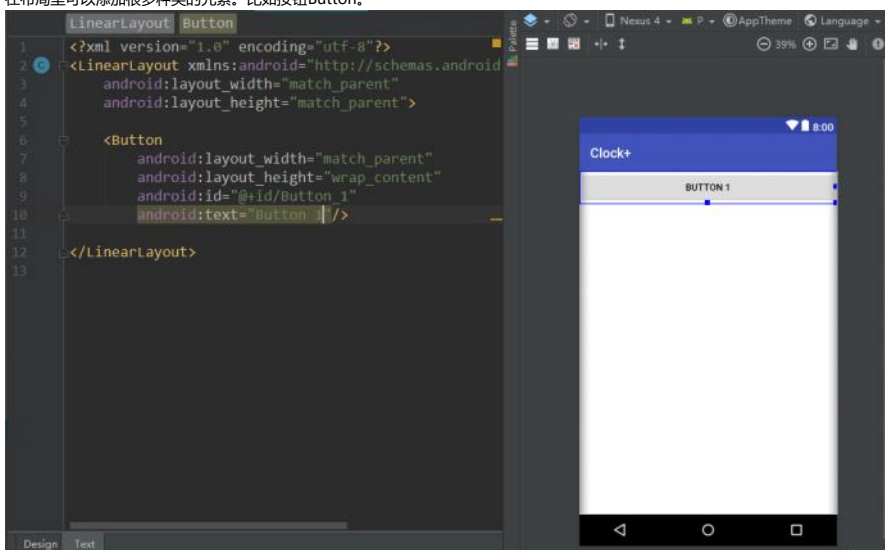


回到Text看代码

```
<?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">

</LinearLayout>
```

在创建布局的时候我们选择了LinearLayout作为根元素，所以这里已经有了一个布局元素了。从英文里应该可以猜到每一项的意思。在布局里可以添加很多种类的元素。比如按钮Button。



上面的代码添加了一个Button元素，并设置了属性：

注意有关的语法

android: id 是给当前元素一个唯一的标识符，以便今后在代码中对这个元素进行操作。相应的他的引号里面的是@+id/Button_1（联系字符串的引用）

我们后面的引用是用@id/(name)

android: layout_width 是设置宽度 相应的height是设置高度

这里的match parent 是指让元素的宽度或者高度和父元素一样。 wrap content指的是让元素的长或者宽正好可以包含里面的内容就行。

回到FirstActivity添加如下代码

```
public class FirstActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.first_layout);
    }
}
```

setContentView（）方法表示为当前的活动加载一个布局，在这个方法一般传入的是一个布局文件的id。任何添加到项目里的资源都会在R反面中生成相应的id。刚才创建的first_layout.XML的id已经添加进去了。只需要用R.layout.first_layout.xml进行调用就可以了。

在AndroidManifest文件里注册

所有的活动都需要在AndroidManifest文件里注册才能生效。（查错位置之一）

FirstActivity已经在这个文件里注册过了，这是as自己自动完成的。as基本会完成所有的组件的注册。看一下这个文件：

```

manifest
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.administer.clock">
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/ic_launcher"
8          android:label="Clock+"
9          android:roundIcon="@mipmap/ic_launcher_round"
10         android:supportsRtl="true"
11         android:theme="@style/AppTheme">
12         <activity android:name=".FirstActivity"></activity>
13     </application>
14
15 </manifest>

```

活动的声明要放在<application>标签内，这是通过<activity>标签来对活动进行注册的。

看到android:name这一项，“.FirstActivity”是com.example.activitytest.FirstActivity的缩写。因为最外层的<manifest>标签就已经通过package限定了程序的包名是com.example.activitytest，因此在注册活动时就可以省略了。

但是仅仅注册了活动还是不可以运行的，因为还没有配置主活动，程序启动时不知道要首先启动哪一个活动。

配置主活动的方法：

在<Activity>标签的内部加入<intent - filter>标签，并在这个标签里添加

```
<action android:name="android.intent.action.MAIN"/>
```

```
<category android:name="android.intent.category.LAUNCHER"/>
```

两句声明即可。

除此之外，还可以用android:label 指定活动中标题栏的内容，标题栏是显示在活动的最顶部的，运行时可以看到。**注意：给主活动指定的label不仅会成为标题栏中的内容，而且会成为启动器（Launcher）中程序显示的名称。**

这样FirstActivity就成为我们应用的主活动了，就是点击桌面上的图标就会打开这个应用。

可以试一下编译和运行了。

最后的注意：你的程序没有设置主活动，也是可以安装的，只不过你看不到它，自然也没办法打开它，一般这样的都是作为第三方应用为别人提供调用的。