

android
developers

Android开发教程&笔记



android

Tab 与 TabHost

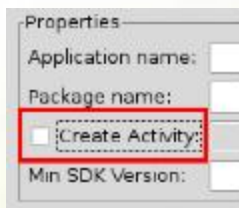


这就是 Tab，而盛放 Tab 的容器就是 TabHost

如何实现？？

每一个 Tab 还对应了一个布局，这个就有点好玩了。一个 Activity，对应了多个功能布局。

① 新建一个 Tab 项目，注意，不要生成 main Activity



这里不要选

② 在包里面新建一个类 MyTab，继承于 TabActivity

其实，TabActivity 是 Activity 的子类

```
package zyf.tab.test;
```

```
import android.app.TabActivity;
```

```
public class MyTab extends TabActivity {
```

```
}
```

③ 从父类继承 onCreate()入口方法

```
package zyf.tab.test;
```

```
import android.app.TabActivity;
```

```
import android.os.Bundle;
```

```
public class MyTab extends TabActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        // TODO Auto-generated method stub
```

```
        super.onCreate(savedInstanceState);
```

```
    }
```

```
}
```

④ 在 Manifest.xml 文件中注册一下 MyTab 类 (Activity)

```
<activity android:name=".MyTab">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"></action>
        <category android:name="android.intent.category.LAUNCHER"></category>
    </intent-filter>
</activity>
```

⑤ 这时候, 需要设计一下标签页对应的布局, 一般采用 FrameLayout 作为根布局, 每个标签页对应一个子节点的 Layout

```
<?xml version="1.0" encoding="utf-8"?>
<!-- 这里是根节点布局 -->
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent">

<!-- 第一个 Tab 对应的布局 -->
    <LinearLayout android:id="@+id/widget_layout_Blue"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:orientation="vertical" >
        <EditText android:id="@+id/widget34" android:layout_width="fill_parent"
            android:layout_height="wrap_content" android:text="EditText"
            android:textSize="18sp">
        </EditText>
        <Button android:id="@+id/widget30" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:text="Button">
        </Button>
    </LinearLayout>
<!-- 第二个 Tab 对应的布局 -->
    <LinearLayout android:id="@+id/widget_layout_red"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:orientation="vertical" >
        <AnalogClock android:id="@+id/widget36"
            android:layout_width="wrap_content" android:layout_height="wrap_content">
        </AnalogClock>
    </LinearLayout>
<!-- 第三个 Tab 对应的布局 -->
    <LinearLayout android:id="@+id/widget_layout_green"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:orientation="vertical">
        <RadioGroup android:id="@+id/widget43"
            android:layout_width="166px" android:layout_height="98px"
            android:orientation="vertical">
            <RadioButton android:id="@+id/widget44"
                android:layout_width="wrap_content" android:layout_height="wrap_content"
```

```
        android:text="RadioButton">
    </RadioButton>
    <RadioButton android:id="@+id/widget45"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="RadioButton">
    </RadioButton>
</RadioGroup>
</LinearLayout>
</FrameLayout>
```

⑥ 首先, 应该声明 TabHost, 然后用 LayoutInflater 过滤出布局来, 给 TabHost 加上含有 Tab 页面的 FrameLayout

```
private TabHost myTabhost;
myTabhost=this.getTabHost();//从 TabActivity 上面获取放置 Tab 的 TabHost
LayoutInflater.from(this).inflate(R.layout.main, myTabhost.getTabContentView(), true);
//from(this)从这个 TabActivity 获取 LayoutInflater
//R.layout.main 存放 Tab 布局
//通过 TabHost 获得存放 Tab 标签页内容的 FrameLayout
//是否将 inflate 拴系到根布局元素上
myTabhost.setBackgroundColor(Color.argb(150, 22, 70, 150));
//设置一下 TabHost 的颜色
```

⑦ 接着, 在 TabHost 创建一个标签, 然后设置一下标题 / 图标 / 标签页布局

```
myTabhost
    .addTab(myTabhost.newTabSpec("TT")// 制造一个新的标签 TT
        .setIndicator("KK",
            getResources().getDrawable(R.drawable.ajjc))
        // 设置一下显示的标题为 KK, 设置一下标签图标为 ajjc
        .setContent(R.id.widget_layout_red));
//设置一下该标签页的布局内容为 R.id.widget_layout_red, 这是 FrameLayout 中的
一个子 Layout
```

⑧ 标签切换事件处理, setOnTabChangeListener

```
myTabhost.setOnTabChangeListener(new OnTabChangeListener(){
    @Override
    public void onTabChanged(String tabId) {
        // TODO Auto-generated method stub
    }
});
```

⑨ 各个标签页的动态 MENU

先把在 XML 中设计好的 MENU 放到一个 int 数组里

```
private static final int myMenuResources[] = { R.menu.phonebook_menu,
        R.menu.addphone_menu, R.menu.chatting_menu, R.menu.userapp_menu };
```

在 setOnTabChangeListener() 方法中根据标签的切换情况来设置 myMenuSettingTag

```
@Override
public void onTabChanged(String tagString) {
    // TODO Auto-generated method stub
    if (tagString.equals("One")) {
        myMenuSettingTag = 1;
    }
    if (tagString.equals("Two")) {
        myMenuSettingTag = 2;
    }
    if (tagString.equals("Three")) {
        myMenuSettingTag = 3;
    }
    if (tagString.equals("Four")) {
        myMenuSettingTag = 4;
    }
    if (myMenu != null) {
        onCreateOptionsMenu(myMenu);
    }
}
```

然后 onCreateOptionsMenu(Menu menu) 方法中通过 MenuInflater 过滤器动态加入 MENU

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    // Hold on to this
    myMenu = menu;
    myMenu.clear();//清空 MENU 菜单
    // Inflate the currently selected menu XML resource.
    MenuInflater inflater = getMenuInflater();
    //从 TabActivity 这里获取一个 MENU 过滤器
    switch (myMenuSettingTag) {
        case 1:
            inflater.inflate(myMenuResources[0], menu);
            //动态加入数组中对应的 XML MENU 菜单
            break;
        case 2:
            inflater.inflate(myMenuResources[1], menu);
            break;
```



```
case 3:
    inflater.inflate(myMenuResources[2], menu);
    break;
case 4:
    inflater.inflate(myMenuResources[3], menu);
    break;
default:
    break;
}
return super.onCreateOptionsMenu(menu);
}
```



⑩ 运行效果



撰写: 地狱怒兽

联系: zyf19870302@126.com

אנדרואיד
developers



אנדרואיד