

通过 DialogFragment 自定义对话框界面

知识解析

可以通过 DialogFragment 来自定义一个对话框的界面,而不是用 AlertDialog 默认界面。

步骤如下:

1. 定义自定义的 Dialog 的 layout

```
<?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:gravity="center"
    android:padding="10dp"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/image"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />
```

2.

```

<TextView>
    android:id="@+id/title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Lorem ipsum dolor sit amet..."
    android:textSize="20dp" />

<Button
    android:id="@+id/dismiss"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Dismiss" />

</LinearLayout>

```

3. 继承 DialogFragment，覆盖 onCreateView()方法，使用 inflater 来 inflate 自定义的 Dialog 的 layout 即可。
4. 可以通过 DialogFragment 的 getDialog()来得到 Dialog 对象，可以对它进行设置 Title 等操作
5. 通过 inflater inflate 的 View 对象来获得自定义 layout 中的组件，例如上面的 Button 等，然后加上监听器

```

class MyDialogFragment extends DialogFragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_sample_dialog,
container, false);

```

6.

```

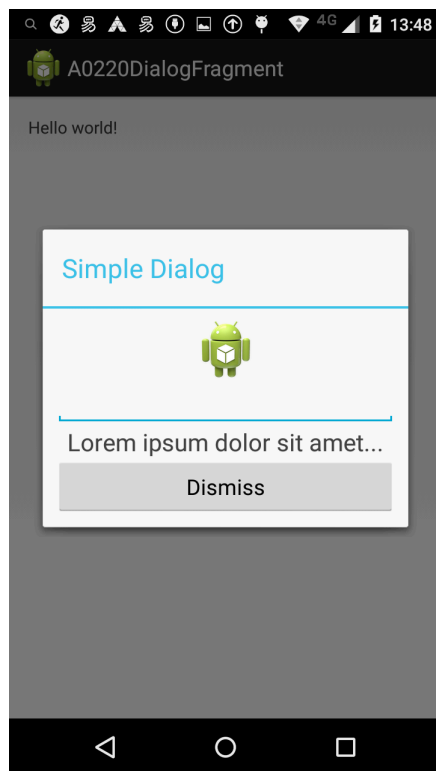
getDialog().setTitle("Simple Dialog");
Button btn = (Button) rootView.findViewById(R.id.dismiss);
    btn.setOnClickListener(new OnClickListener(){
        @Override

```

```
        public void onClick(View v) {  
            // TODO Auto-generated method stub  
            getDialog().dismiss();  
        }  
    }  
}
```

这样，一个完全自定义的对话框就完成了。

功能演示



职业素质

使用DialogFragment来管理对话框，当旋转屏幕和按下后退键时可以更好的管理其声明周期，它和Fragment有着基本一致的声明周期。且DialogFragment也允许开发者把Dialog

作为内嵌的组件进行重用，类似Fragment（可以在大屏幕和小屏幕显示出不同的效果）。上面会通过例子展示这些好处~

使用 DialogFragment 至少需要实现 onCreateView 或者 onCreateDialog 方法。onCreateView 即使用定义的 xml 布局文件展示 Dialog。onCreateDialog 即利用 AlertDialog 或者 Dialog 创建出 Dialog。