

DialogFragment

知识解析

在 API 13 之后，Activity 中的 onCreateDialog()/onPrepareDialog()等方法被 deprecated，推荐使用 DialogFragment 来实现：

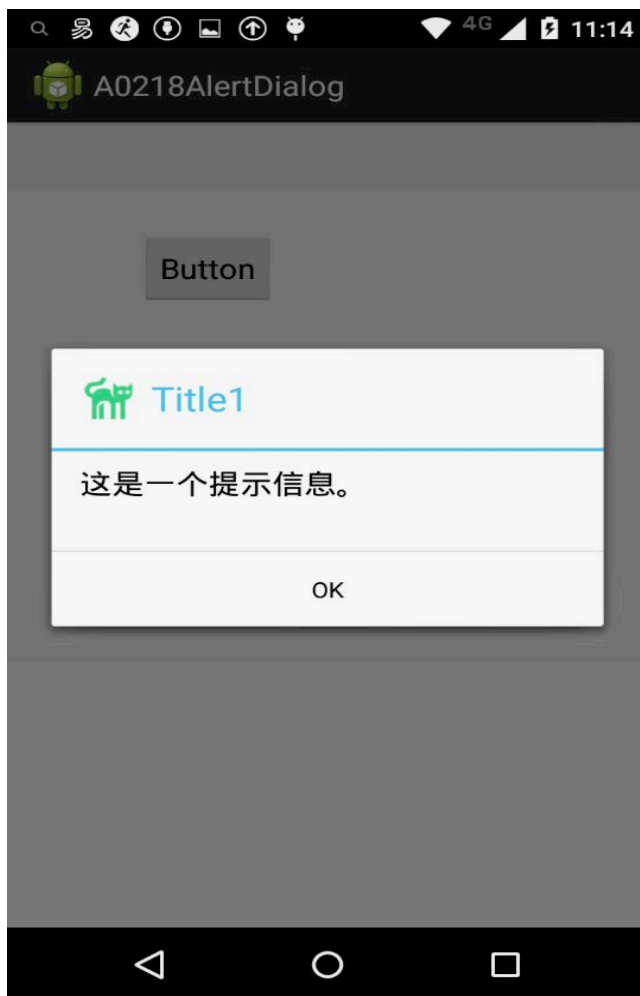
创建 DialogFragment 的子类，覆盖 onCreateDialog(Bundle bundle)方法，创建并返回 Dialog。例如：

```
public class MyDialogFragment extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        Log.i("Msg", "onCreateDialog invoked");
        Dialog dialog = null;
        AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
        builder.setTitle("title");
        builder.setIcon(R.drawable.ic_launcher);
        builder.setMessage("提示信息提示信息");
        builder.setCancelable(false);
        builder.setPositiveButton("OK", null);
        dialog = builder.create();
        return dialog;
    }
}
```

每次需要 Dialog 的时候,调用 DialogFragment 上的 show(FragmentManager mgr,String tag)方法来显示:

```
MyDialogFragment fragment = new MyDialogFragment();  
Bundle bundle = new Bundle();  
bundle.putString("tag", "title");  
fragment.setArguments(bundle);  
fragment.show(MainActivity.this.getSupportFragmentManager(), "test");
```

功能演示



实战操作

第一步

```
public class MyDialogFragment extends DialogFragment {  
    @Override  
    public Dialog onCreateDialog(Bundle savedInstanceState) {  
        Dialog dialog = null;  
        AlertDialog.Builder builder = new  
AlertDialog.Builder(getActivity());  
        builder.setTitle("title");  
        builder.setIcon(R.drawable.ic_launcher);  
        builder.setMessage("提示信息提示信息");  
        builder.setCancelable(false);  
        builder.setPositiveButton("OK", null);  
        dialog = builder.create();  
        return dialog;  
    }  
}
```

第二步

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Button btn = (Button) this.findViewById(R.id.button1);  
        btn.setOnClickListener(new OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                MyDialogFragment fragment = new MyDialogFragment();  
                Bundle bundle = new Bundle();  
                bundle.putString("tag", "title");  
            }  
        });  
    }  
}
```

```
        fragment.setArguments(bundle);  
        fragment.show(MainActivity.this.getFragmentManager(),  
"test");  
    }  
  
    });  
}  
}
```

职业素质

DialogFragment 在 android 3.0 时被引入。是一种特殊的 Fragment，用于在 Activity 的内容之上展示一个模态的对话框。典型的用于：展示警告框，输入框，确认框等等。

在 DialogFragment 产生之前，我们创建对话框：一般采用 AlertDialog 和 Dialog。注：官方不推荐直接使用 Dialog 创建对话框。。

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。