

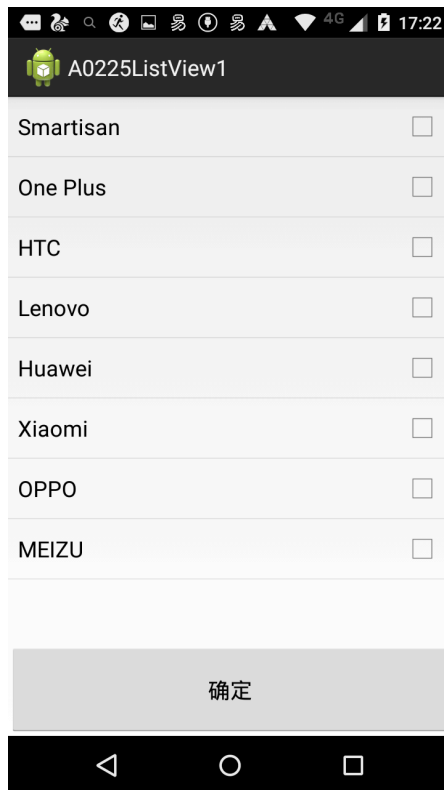
处理 ListView 多选列表的选项

知识解析

可以使用 ListView 上的 `getCheckedItemPositions()` 来取得选中的选项，它返回一个 `SparseBooleanArray` 对象，`SparseBooleanArray` 是一个保存 key-value 的映射对象，返回的结果对象里，保存了所有选项的状态，key 为 ListView 重的选项的位置，value 为是否选中的状态。因此，可以通过它来处理多选列表的选项：

```
SparseBooleanArray array =  
lv.getCheckedItemPositions();  
  
    int size = array.size();  
    for(int i = 0;i<size;i++){  
        if(array.valueAt(i)){  
  
            Log.i("Selected",""+array.keyAt(i));  
  
        }  
    }  
}
```

功能演示



实战操作

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        ArrayAdapter<CharSequence> adapter =  
        ArrayAdapter.createFromResource(this, R.array.mobiles,  
            android.R.layout.simple_list_item_multiple_choice);  
  
        adapter.setDropDownViewResource(android.R.layout.simple_dropdown  
            _item_1line);  
    }  
}
```

```
        final ListView lv = (ListView)
this.findViewById(R.id.listView1);

        lv.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
        lv.setAdapter(adapter);

        Button btn = (Button) this.findViewById(R.id.button1);
        btn.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View v) {

                SparseBooleanArray array =
lv.getCheckedItemPositions();

                int size = array.size();

                for (int i = 0; i < size; i++) {

                    if (array.valueAt(i)) {

                        Log.i("Selected", "" + array.keyAt(i));

                    }
                }
            }
        });
    }
}
```

职业素质

根据开发中遇到的场景选择相应的多选模式。他能更直观的为我们现显示是险要的数据，也可以更直接的为用户提供选择。实现过程简单，如果需要美观的话需要自定义样式。