

ListView 动态增、删数据

知识解析

当 Adapter 里封装的数据改变时，需要调用 BaseAdapter 的 `notifyDataSetChanged()` 方法来通知 ListView，让 ListView 对数据变化做出响应。

如果使用的是 `ArrayAdapter`，可以采用方法 2：使用 `ArrayAdapter` 的 `add()/remove()/clear()` 方法来操作数据，配合 `ArrayAdapter` 的 `setNotifyDataSetChanged(true)` 使用。如：

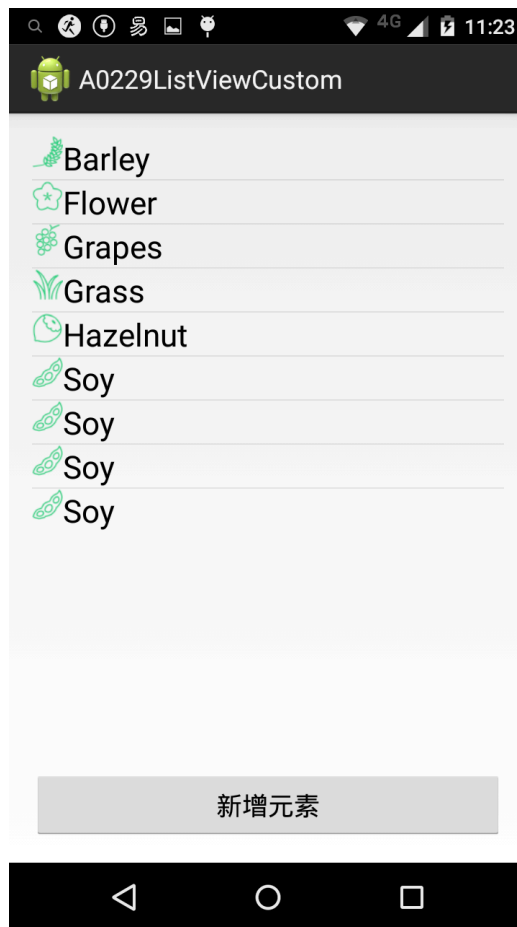
```
MyAdapter adapter = new MyAdapter(this, 0, list);
    adapter.setNotifyOnChange(true);
    ListView lv = (ListView)
this.findViewById(R.id.listView1);
    lv.setAdapter(adapter);
    Button btn = (Button)
this.findViewById(R.id.button1);
    btn.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // list.add(new
ImageText(R.drawable.strawberry, "StrawBerry"));
            // adapter.notifyDataSetChanged();
            adapter.add(new ImageText(R.drawable.soy,
"Soy"));
        }
    });
```

这里当点击按钮后，将会调用 Adapter 上的 `add()` 方法往里加入一个数据，而 adapter 上调用了 `setNotifyOnChange(true)`，因此，当往 adapter 中放入数据后，将会通知到对应的

ListView 更新视图。

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);  
        int[] imgs = { R.drawable.barley, R.drawable.flower,
```

```
R.drawable.grapes, R.drawable.grass, R.drawable.hazelnut };

    String[] names = { "Barley", "Flower", "Grapes", "Grass",
"Hazelnut" };

    final ArrayList<ImageText> list = new ArrayList<ImageText>();
    for (int i = 0; i < imgs.length; i++) {
        ImageText it = new ImageText(imgs[i], names[i]);
        list.add(it);
    }

    final MyAdapter adapter = new MyAdapter(this, 0, list);
    ListView lv = (ListView) this.findViewById(R.id.listView1);
    lv.setAdapter(adapter);

    Button btn = (Button) this.findViewById(R.id.button1);
    adapter.setNotifyOnChange(true);
    btn.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            adapter.add(new ImageText(R.drawable.soy, "Soy"));
        }
    });
}

class ImageText {
    private int imageId;
    private String text;

    public ImageText(int imageId, String text) {
        this.setImageId(imageId);
        this.setText(text);
    }

    public int getImageId() {
        return imageId;
    }

    public void setImageId(int imageId) {
        this.imageId = imageId;
    }
}
```

```
}

public String getText() {
    return text;
}

public void setText(String text) {
    this.text = text;
}
}

class MyAdapter extends ArrayAdapter<ImageText> {
    List<ImageText> list;

    public MyAdapter(Context context, int resource, List<ImageText>
objects) {
        super(context, resource, objects);
        this.list = objects;
    }

    @Override
    public int getCount() {
        return list.size();
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        View view = null;
        ViewHolder vh = null;
        if (convertView == null) {
            // 自己创建一个Item View
```

```
        LayoutInflater inflater = (LayoutInflater)
MainActivity.this.getSystemService(LAYOUT_INFLATER_SERVICE);
        view = inflater.inflate(R.layout.item, parent, false);
        ImageView iv = (ImageView)
view.findViewById(R.id.imageView1);
        TextView tv = (TextView)
view.findViewById(R.id.textView1);
        vh = new ViewHolder();
        vh.vh_iv = iv;
        vh.vh_tv = tv;
        view.setTag(vh); } else {
        view = convertView;
        vh = (ViewHolder) view.getTag();
        ImageText it = list.get(position);
        vh.vh_iv.setImageResource(it.getImageId());
        vh.vh_tv.setText(it.getText());
        return view;
    }
    class ViewHolder {
        TextView vh_tv;
        ImageView vh_iv;
    }
}
```

职业素质

ListView 是 Android 开发中最常用到的控件之一，所以学习 ListView 显得非常重要，继承 BaseAdapter 实现自己的 Adapter，可以在其中实现相应的方法，包括添加，删除，清空 list。