
ListView 实现过程和常见问题

一、 ListView 简介

ListView 是 android 中常用的一个控件，通常是以列表的形式展示数据，并且当加载的数据过多的时候可以分页加载，动态显示数据，通常是横向显示数据。

二、 ListView 实现数据加载的步骤:

ListView 适合“自底向上”的开发模式，即从每个条目的显示组件，到对其进行控制的数据结构，最后通过 Activity 等进行使用。

- a) 首先是 item 组件，即用于每项布局输出的 xml 文件。Android SDK 中有 simple_list_item_1、simple_list_item_2 可用，当需要比较丰富的显示效果时，一般通过自定义 xml 实现。
- b) 其次是父对象 layout 文件，即用于 Activity 或者 Fragment 的布局输出文件，一般在此输出文件中包含 ListView。当然，如果采用 ListFragment 或 ListActivity，并不需要再显示的定义 ListView 组件
- c) 定义数据结构（容器），即用于持有单个 Item 的数据，可以是简单的 String，也可以通过抽象 Items 所需字段组成一个类，抽象的原则是与 Item 中的组件对应。本文中上图涉及多个字段，因此通过抽象组件形成 BBSTopicItem 类。
- d) 列表适配器。决定每行 Item 中具体显示什么内容，以怎样的样式显示等，通常通过继承 ArrayAdapter、SimpleAdapter 等实现。本文定义 BBSTopicAdapter，继承于 ArrayAdapter<BBSTopicItem>。
- e) 最后，需要定义一个 Activity 或 Fragment 来使用上述模块。需要说明的是，ListView 可以直接被 ListActivity 或者 ListFragment 使用。

三、 ListView 在开放中常见问题

1. ListView 滑动停止下载：

- 1) 滑动时停止下载也是提高用户体验的方式之一，因为如果在 ListView 滑动过程中执行下载任务，将会使得 ListView 出现卡顿。监听滑动状态改变的方法是 `onScrollStateChanged(AbsListView view, int scrollState)`，该方法在 `OnScrollListener` 接口中定义的，而 `OnScrollListener` 是 `AbsListView` 中为了在列表或网格滚动时执行回调函数而定义的接口。
- 2) 为了实现下载任务与滑动状态的关联，在自定义列表适配器中实现了 `OnScrollListener` 接口，在 `onScrollStateChanged` 方法中根据 `scrollState` 执行相应的下载任务操作。代码如下所示：

```
public void onScrollStateChanged(AbsListView view, int scrollState) {
    this.scrollState = scrollState;
    if (scrollState == AbsListView.OnScrollListener.SCROLL_STATE_IDLE)
    {
        showImage(mFirstVisibleItem, mVisibleItemCount);
    } else {
        cancelTask();
    }
}
```

2. notifyDataSetChanged 与局部更新：

`notifyDataSetChanged` 就是通过操作 `mDataSetObservable` 实现的，`DataSetObservable` 是观察者模式的一个实现。`notifyDataSetChanged` 调用了 `notifyChanged` 方法，回到 `DataSetObservable` 中：

3. listview 的 item 内部组件的事件响应

在具体的工程中，item 组件的响应会根据对其使用的 Activity (Fragment) 的不同而变化，因此，不宜在其部设定响应事件的具体实现。推荐在 adapter 中定义接口，将接口暴露给具体的 Activity (Fragment)，Activity (Fragment) 根据具

体的业务逻辑进行配置。

4. ScrollView 与 ListView 的冲突

如果在 ScrollView 中嵌套了 ListView(原则上应尽量避免这种情况),那么很不幸,可能会遇到以下问题。

ListView 只显示一行;页面默认不从顶端开始显示;ListView 滑动事件无法监听,这几个问题都是比较常见的问题了,这里不再赘述其原理,给出比较通用的解决方案:

- i. listview 需要手动设置高度;
listview 需要设置 `listview.setFocusable(false);`
- ii. 重载 listview 的 `onInterceptTouchEvent` 方法,在 `ACTION_DOWN` 时通过 ScrollView 的 `requestDisallowInterceptTouchEvent` 方法设置交出 `ontouch` 权限, `ACTION_CANCEL` 时再恢复 `ontouch` 权限。
- iii. 再次强调,应尽量避免 ScrollView 中嵌套了 ListView。

5. ListView 的分页加载:

- a) ListView 分页加载功能:

ListView 的分页功能主要是用在以下的场景,当我们要加载的数据很多,我们不可能将某些数据全部的展示在界面上,通常情况是当我们在手指滑动的时候会有“查看更多”,当我们点击或者下拉时就是开始加载更多的信息,这样的功能我们一般叫做数据下拉刷新功能,也是分页加载功能。

- b) 分页加载实现的原理:

首先用户确认默认展示在 ListView 上的数据,比如 ListView 每页默认显示多少条数据将数据传递给自定义适配器上,然后加载到 listview 中,当用户将数据下拉到最

后一条数据的时候，就要开始加载新的数据了，通过监听 listView 的滑动事件，判断是否是最后一条，如果达到最后一条则开始刷新。

