

- 1、创建ListView组件及xml布局文件
- 2、创Item的xml布局文件
- 3、创建JAVABean并自定义类用来保存获取数据
- 4、创建集合将保存在自定义类中的数据添加到集合中 javabean集合
- 5、代码获取ListView的ID (ListView) findViewById(R.id.lv_select_safenum);
- 6、代码先获取集合对象加载数据前必须对集合初始化赋值

```
private List<AppMessage> mAppMessage = new ArrayList<AppMessage>();
```

- 7、代码通过对象点方法返回集合对象添加数据

```
mAppMessage = AppInitUtils.getAppMessage(getApplicationContext());
```

- 8、创建数据适配器对象 继承extends BaseAdapter

```
class SelectSafeNumAdapter extends BaseAdapter
```

实现四个方法

getCount() 获取一共有多少item

getItem(int position) 获取某个item

getItemId(int position) 获取item的ID

getView(int position, View convertView, ViewGroup parent)

1、创建袋子对象 ViewHolder holder = null;

2、绑定袋子

3、找到item的组件

4、获取绑定有item

5、为item中的组件设置数据

6、返回视图

```
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder = null;
    if (convertView == null) {
        // 1 创建袋子
        holder = new ViewHolder();
        convertView = View.inflate(getApplicationContext(), R.layout.view_lv_select_safenum_item);
        // 2 绑定
        convertView.setTag(holder);
        // 3 findViewById, 放进袋子
        holder.ivSelectSafeNum = (ImageView) convertView.findViewById(R.id.iv_select_safenum);
        holder.tvSelectName = (TextView) convertView.findViewById(R.id.tv_select_name);
        holder.tvSelectPhone = (TextView) convertView.findViewById(R.id.tv_select_phone);
    } else {
        holder = (ViewHolder) convertView.getTag();
    } // 设置数据到item组件
    ContactInfo info = (ContactInfo) getItem(position);
    holder.ivSelectSafeNum
        .setImageBitmap(ContactsUtils.getContactPhotoFromContactId(getApplicationContext(),
        holder.tvSelectName.setText(info.name);
        holder.tvSelectPhone.setText(info.number);
    // 返回视图
    return convertView;
}
```

- 9、提前创建好装item布局的袋子

```
class ViewHolder {
    public ImageView ivSelectSafeNum;
    public TextView tvSelectName;
    public TextView tvSelectPhone;
}
```

10、将数据适配器对象设置给ListView `ListView.setAdapter(adapter);`

11、数据适配器类中的加载数据方法设置

1、用一个类保存ITEM中所有的UI组件

存: `public ImageView ico;` UI组件的用法: `houderView.ico`

2、将保存UI的类与convertView绑定

3、通过保存UI的类找到ITEM中的所有组件

找item布局

`convertView = View.inflate(getApplicationContext(), R.layout.app_data_item, null);`

找UI组件 `houderView.ico = (ImageView) convertView.findViewById(R.id.app_ico);`

4、通过集合的条目获取自定义对象

`AppMessage message = mAppMessage.get(position);`

5、通过自定义对象添加数据到ListView的每一个UI组件

`houderView.ico.setImageDrawable(message.appico);`

6、返回convertView

ListView列表更新 数据适配器对象.`notifyDataSetChanged();`

应用图片 JAVA并自定义类保存图片数据的方式 `public Drawable appico;`

设置ImageView图片的方式 `setImageDrawable(message.appico);`

自带上图拉加载更多下拉刷新的第三方Listview CommonPullToRefresh:

下拉刷新支持大部分view: ListView、ScrollView、WebView等, 甚至一个单独的TextView

加载更多目前支持ListView、RecyclerView、GridView、SwipeRefreshLayout

支持自定义header以及footer

增加SwipeRefreshLayout刷新方式, 同样支持加载更多

依赖库: `compile 'com.chanven.lib:cptr:1.1.0'`

基本用法: 在布局文件中添加PtrClassicFrameLayout

```
<com.chanven.lib.cptr.PtrClassicFrameLayout
    xmlns:cube_ptr="http://schemas.android.com/apk/res-auto"
    android:id="@+id/test_list_view_frame"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f0f0f0"
    cube_ptr:ptr_duration_to_close="200"
    cube_ptr:ptr_duration_to_close_header="700"
    cube_ptr:ptr_keep_header_when_refresh="true"
    cube_ptr:ptr_pull_to_fresh="false"
    cube_ptr:ptr_ratio_of_header_height_to_refresh="1.2"
```

```
cube_ptr:ptr_resistance="1.8" >
```

```
<ListView
    android:id="@+id/test_list_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:dividerHeight="0.1dp"
    android:background="@android:color/white"
    android:divider="#b0b0b0"
    android:choiceMode="singleChoice"
    android:fadingEdge="none"
    android:scrollbarStyle="outsideOverlay" />
</nven.lib.cptr.PtrClassicFrameLayout>
```

SpringView :

SpringView 是一个提供上下拖放组件的功能，可以高度定制，实现各种下拉\拉绘画效果，完全兼容源控件如ListView, RecyclerView, ScrollView, WebView等，使用简单，轻松自定义风格的拖放页面 SpringView个人独立将头/几乎实现任何你想要的效果，只需要继承BaseHeader（或Footer）

依赖: compile 'com.liaoinstan.springview:library:1.2.6'

基本用法:

在布局文件中添加Spring View，注意Spring View和ScrollView有相同的限制：只有一个子元素：

```
<com.liaoinstan.springview.widget.SpringView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:header="@layout/myheader"
    app:footer="@layout/myfooter">

    <listView
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</com.liaoinstan.springview.widget.SpringView>
```

滚屏控件ScrollView和HorizontalScrollView

ScrollView和HorizontalScrollView这两个View的定义。ScrollView和HorizontalScrollView

都是一个布局容器，里面可以放入child View控件，我们通过其继承关系看到，ScrollView和HorizontalScrollView这两个类是ViewGroup的一个间接子类。
ScrollView和HorizontalScrollView只是两种滚动方向不同的View而已

类试于ListView的控件：GridView