

使用 ExpandableListView 实现可折叠的二级列表

一、ExpandableListView 介绍

ExpandableListView 是一种用于垂直展示两级列表的视图，与 ListView 不同的地方在于它可以展示两级列表，而这些数据是通过 ExpandableListAdapter 关联的，通过修改该 ExpandableListAdapter，可以实现展示并根据用户搜索的内容更改列表显示的内容。

在本文中，额外实现了省份图片展示、动态背景变换以及实时搜索功能。

二、实现效果

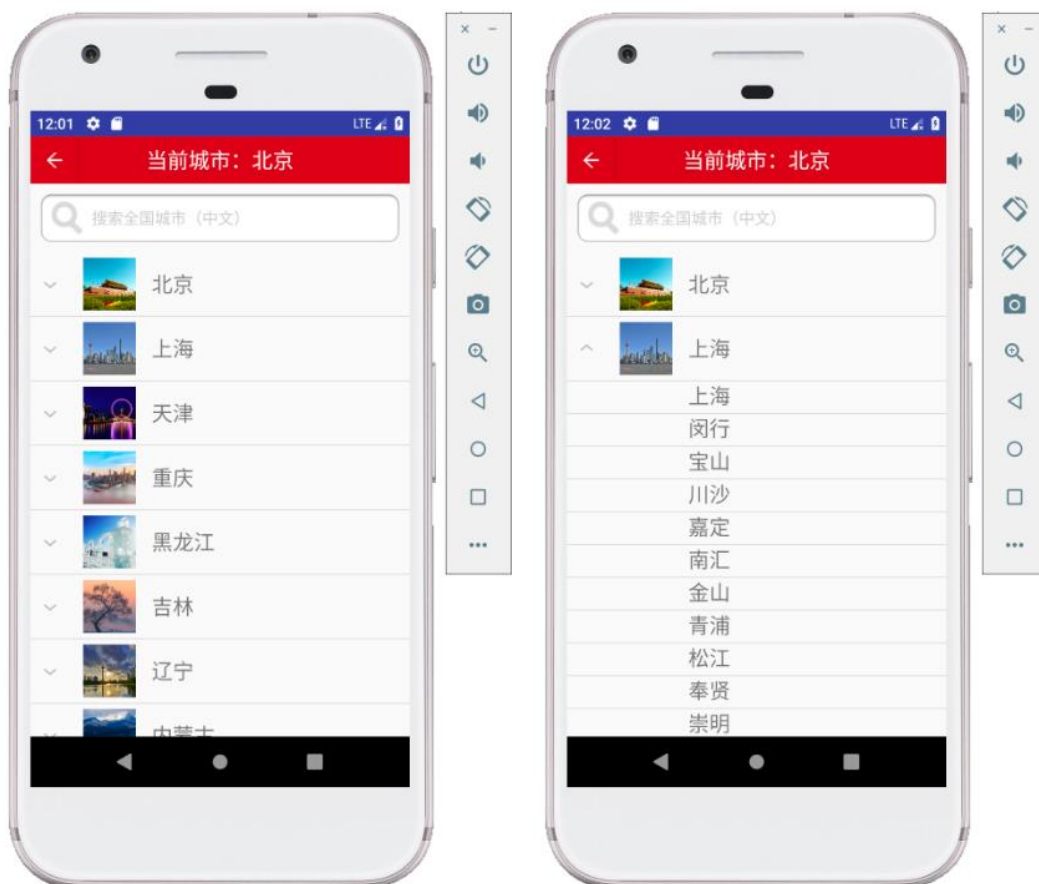


图 1 图形化的选择界面



图 2 根据用户选择的的城市所在省份的不同切换背景

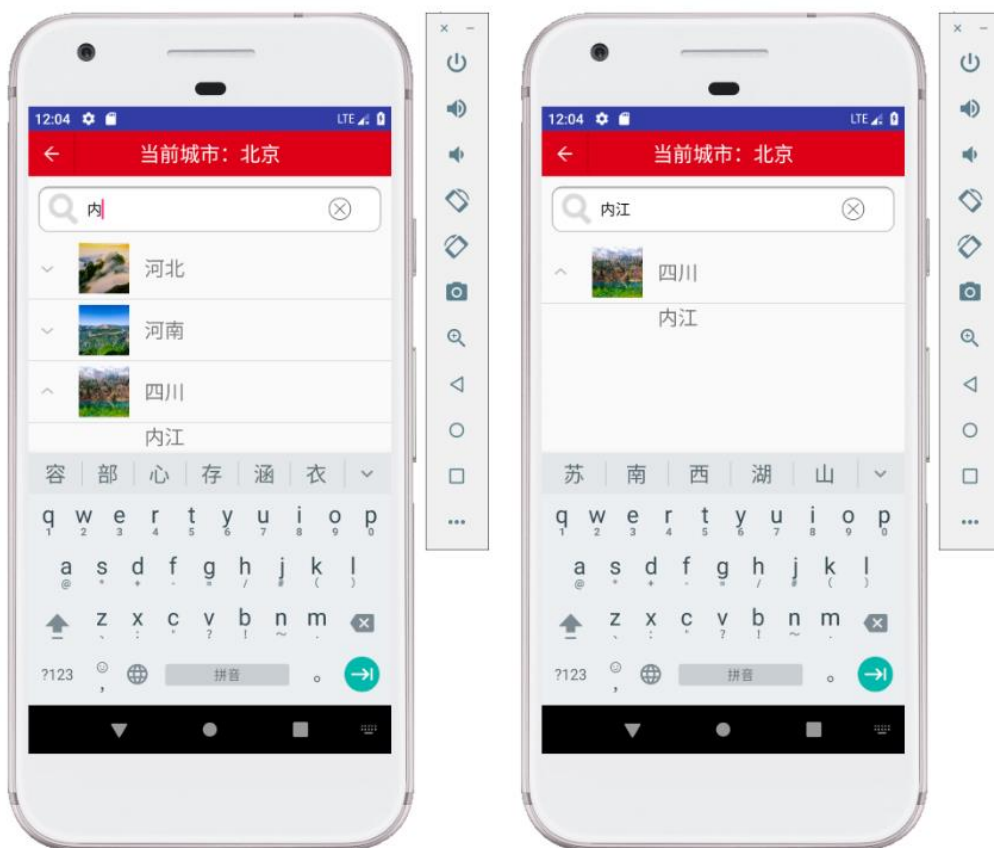


图 3 实时预览搜索结果

三、具体实现

由于改动的地方比较多，如果有疑惑，请参见：

https://github.com/RobinChen95/Android_Project

1. XML 布局文件

创建 item_first.xml 与 item_second.xml 文件，分别对应一级菜单（省份）与二级菜单（城市）的布局文件：

item_first.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:layout_width="50dip"
        android:layout_height="50dip"
        android:layout_marginLeft="50dip"
        android:layout_gravity="center_vertical"
        android:id="@+id/province_icon"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="60dip"
        android:layout_marginLeft="15dip"
        android:id="@+id/first_group"
        android:gravity="center"
        android:text="first group"
        android:textSize="20dip"
    />
</LinearLayout>
```

item_second.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="30dip"
        android:id="@+id/second_group"
        android:paddingLeft="115dip"
        android:text="second group"
        android:textSize="20dip"
    />

</LinearLayout>
```

2. 修改 SelectCity.java

在 SelectCity 实现一级的列表的基础上，将 SelectCity 对应的 ListView 列表、对象以及监听器修改为 ExpandableListView 类的列表、对象以及监听器：

SelectCity.java:

```
package com.example.chenguoqiang.android_project;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.text.Editable;
import android.text.TextUtils;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.ExpandableListView;
import android.widget.ImageView;
import android.widget.TextView;

import com.pku.edu.ChenGuoqiang.app.MyApplication;
```

```
import com.pku.edu.ChenGuoqiang.bean.City;
import com.pku.edu.ChenGuoqiang.util.ClearEditText;
import com.pku.edu.ChenGuoqiang.util.MyExpandableListView;

import java.util.ArrayList;
import java.util.List;

public class SelectCity extends Activity implements View.OnClickListener {

    private ImageView mBackBtn;
    // 声明 ExpandableListView 控件
    private ExpandableListView mList;
    private List<City> cityList;
    // 搜索框过滤之后的数据
    private ArrayList<City> filteredDataList;
    private String citycode;
    private TextView select_city_Tv;
    // 声明 ExpandableListView 控件的适配器
    private MyExpandableListView myadapter;
    private ClearEditText mClearEditText;

    // 将存有 city 的列表整理为按省份存储的二级列表
    private ArrayList<ArrayList<City>> formattedDataList;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.select_city);
        select_city_Tv = findViewById(R.id.city_name);
        mBackBtn = (ImageView) findViewById(R.id.title_back);
        mBackBtn.setOnClickListener(this);
        initView();

        //设置标题栏的城市，此处是 MyApplication 类的静态变量
        select_city_Tv.setText("当前城市: " + MyApplication.cityname_global);
    }

    private void initView() {
        mList = (ExpandableListView) findViewById(R.id.expandableListView);
        MyApplication myApplication = (MyApplication) getApplication();
        cityList = myApplication.getmCityList();
        filteredDataList = new ArrayList<>(cityList);
        mClearEditText = (ClearEditText) findViewById(R.id.search_bar);
        myadapter = new MyExpandableListView(SelectCity.this,cityList);
    }
}
```

```

mList.setAdapter(myadapter);

// 对二级列表设置点击子类对象 Child_Click 的监听器
mList.setOnChildClickListener(new ExpandableListView.OnChildClickListener() {

    @Override

    public boolean onChildClick(ExpandableListView parent, View v, int groupPosition, int childPosition, long id) {

        formattedDataList = myadapter.formatData(filtedDataList);

        citycode = formattedDataList.get(groupPosition).get(childPosition).getNumber();

        Intent i = new Intent();

        i.putExtra("citycode", citycode);

        setResult(RESULT_OK, i);

        finish();

        return true;

    }

});

mClearEditText.addTextChangedListener(new TextWatcher() {

    @Override

    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }

    @Override

    public void onTextChanged(CharSequence s, int start, int before, int count) {

    }

    @Override

    public void afterTextChanged(Editable s) {

        filterData(s);

    }

});

}

private void filterData(Editable filterStr) {

    // 搜索之后过滤数据

    filtetedDataList = new ArrayList<>();

    Log.d("Filter", filterStr.toString());

    if (TextUtils.isEmpty(filterStr)) {

        for (City city : cityList) {

            filtetedDataList.add(city);

        }

    } else {

```

```

        filteredDataList.clear();
        for (City city : cityList)
            if (city.getCity().indexOf(filterStr.toString()) != -1) {
                filteredDataList.add(city);
            }
        }
        myadapter.updateListView(filteredDataList);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.title_back:
                finish();
                break;
            default:
                break;
        }
    }
}

```

3. 在 util 工具包中新建 MyExpandableListView.java

MyExpandableListView 适配器需要实现：数据更新后的视图更新、将含有 ArrayList<City>的数据格式化成 ArrayList<ArrayList<City>>分省存放、一级列表以及二级列表的展示逻辑，注意图中涉及到的各省图片自己制作或者到：

https://github.com/RobinChen95/Android_Project/tree/master/app/src/main/res/drawable-xhdpi 下载我制作好的图片。

MyExpandableListView.java

```

package com.pku.edu.ChenGuoqiang.util;

import android.content.Context;
import android.util.Log;

```

```
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseExpandableListAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import com.example.chenguoqiang.android_project.R;
import com.pku.edu.ChenGuoqiang.app.MyApplication;
import com.pku.edu.ChenGuoqiang.bean.City;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class MyExpandableListView extends BaseExpandableListAdapter {

    private LayoutInflater mInflater;
    private List<City> mDatas;

    //使用哈希映射存储省份,每个省份下又包含各自城市
    private HashMap<Integer,ArrayList<City>> provinces ;
    //使用哈希映射存储省份对应的数字
    private HashMap<String,Integer> province_integer_map;
    //最终存储的省份列表
    private ArrayList<String> provinces_result ;
    //最终存储的城市列表
    private ArrayList<ArrayList<City>> city_result;
    public MyExpandableListView(Context context, List<City> datas){
        mInflater = LayoutInflater.from(context);
        formatData(datas);
    }

    //自定义的格式化数据的函数
    public ArrayList<ArrayList<City>> formatData(List<City> datas){
        mDatas = datas;
        provinces = new HashMap<>();
        province_integer_map = new HashMap<>();
        provinces_result = new ArrayList<>();
        city_result = new ArrayList<>();
        int provinces_count = 0;
        //将省份与数字建立映射关系
        for (City city: mDatas) {
            if (!province_integer_map.containsKey(city.getProvince())){
```

```

        province_integer_map.put(city.getProvince(), provinces_count);
        provinces_count++;
    }
}

for (City city : mDatas){
    if (!provinces.containsKey(province_integer_map.get(city.getProvince()))){
        //没有该省份时就添加省份的城市列表，可能处理的比较繁琐
        provinces.put(province_integer_map.get(city.getProvince()), new ArrayList<City>());
        provinces_result.add(city.getProvince());
        ArrayList<City> temp = provinces.get(province_integer_map.get(city.getProvince()));
        temp.add(city);
        provinces.remove(province_integer_map.get(city.getProvince()));
        provinces.put(province_integer_map.get(city.getProvince()), temp);
        Log.d("添加了省份", city.getProvince() + "\t 编号: " + province_integer_map.get(city.getProvince()));
    }
    else {
        //如果有了该城市所在的省份，就在对应的省份列表中添加该城市，可能处理的比较繁琐
        ArrayList<City> temp = provinces.get(province_integer_map.get(city.getProvince()));
        temp.add(city);
        provinces.remove(province_integer_map.get(city.getProvince()));
        provinces.put(province_integer_map.get(city.getProvince()), temp);
    }
}

//将结果添加进结果集
for (int i = 0; i < provinces.size(); i++) {
    city_result.add(provinces.get(i));
}

return city_result;
}

public void updateListView(List<City> datas){
    formatData(datas);
    notifyDataSetChanged();
}

/**
 * Gets the number of groups.
 *
 * @return the number of groups
 */
@Override
public int getGroupCount() {
    return provinces_result.size();
}

```

```

/**
 * Gets the number of children in a specified group.
 *
 * @param groupPosition the position of the group for which the children
 *                       count should be returned
 * @return the children count in the specified group
 */
@Override
public int getChildrenCount(int groupPosition) {
    return city_result.get(groupPosition).size();
}

/**
 * Gets the data associated with the given group.
 *
 * @param groupPosition the position of the group
 * @return the data child for the specified group
 */
@Override
public Object getGroup(int groupPosition) {
    return provinces_result.get(groupPosition);
}

/**
 * Gets the data associated with the given child within the given group.
 *
 * @param groupPosition the position of the group that the child resides in
 * @param childPosition the position of the child with respect to other
 *                       children in the group
 * @return the data of the child
 */
@Override
public Object getChild(int groupPosition, int childPosition) {
    return city_result.get(groupPosition).get(childPosition);
}

/**
 * Gets the ID for the group at the given position. This group ID must be
 * unique across groups. The combined ID (see
 * {@link #getCombinedGroupID(long)}) must be unique across ALL items
 * (groups and all children).
 *
 * @param groupPosition the position of the group for which the ID is wanted
 * @return the ID associated with the group

```

```

    */

    @Override
    public long getGroupId(int groupPosition) {
        return groupPosition;
    }

    /**
     * Gets the ID for the given child within the given group. This ID must be
     * unique across all children within the group. The combined ID (see
     * {@link #getCombinedChildId(long, long)}) must be unique across ALL items
     * (groups and all children).
     *
     * @param groupPosition the position of the group that contains the child
     * @param childPosition the position of the child within the group for which
     *                        the ID is wanted
     * @return the ID associated with the child
     */
    @Override
    public long getChildId(int groupPosition, int childPosition) {
        return childPosition;
    }

    /**
     * Indicates whether the child and group IDs are stable across changes to the
     * underlying data.
     *
     * @return whether or not the same ID always refers to the same object
     */
    @Override
    public boolean hasStableIds() {
        return true;
    }

    /**
     * Gets a View that displays the given group. This View is only for the
     * group--the Views for the group's children will be fetched using
     * {@link #getChildView(int, int, boolean, View, ViewGroup)}.
     *
     * @param groupPosition the position of the group for which the View is
     *                        returned
     * @param isExpanded    whether the group is expanded or collapsed
     * @param convertView    the old view to reuse, if possible. You should check
     *                        that this view is non-null and of an appropriate type before
     *                        using. If it is not possible to convert this view to display

```

```

*           the correct data, this method can create a new view. It is not
*
*           guaranteed that the convertView will have been previously
*           created by
*
*           {@link #getGroupView(int, boolean, View, ViewGroup)}.
* @param parent      the parent that this view will eventually be attached to
* @return the View corresponding to the group at the specified position
*/

@Override
public View getGroupView(int groupPosition, boolean isExpanded, View convertView, ViewGroup parent) {
    if (null == convertView) {
        convertView = inflater.inflate(R.layout.item_first, parent, false);
    }
    TextView tvGroup = (TextView) convertView.findViewById(R.id.first_group);
    ImageView imgGroup = (ImageView) convertView.findViewById(R.id.province_icon);
    // 设置一级列表的分组组名
    tvGroup.setText(provinces_result.get(groupPosition));
    MyApplication.province_global = provinces_result.get(groupPosition);
    // 设置图片
    switch (provinces_result.get(groupPosition)){
        case "北京":
            imgGroup.setImageResource(R.drawable.bj);
            break;
        case "上海":
            imgGroup.setImageResource(R.drawable.sh);
            break;
        case "天津":
            imgGroup.setImageResource(R.drawable.tj);
            break;
        case "重庆":
            imgGroup.setImageResource(R.drawable.cq);
            break;
        case "黑龙江":
            imgGroup.setImageResource(R.drawable.hl);
            break;
        case "吉林":
            imgGroup.setImageResource(R.drawable.jl);
            break;
        case "辽宁":
            imgGroup.setImageResource(R.drawable.ln);
            break;
        case "内蒙古":
            imgGroup.setImageResource(R.drawable.nmg);
            break;
        case "河北":

```

```
        imgGroup.setImageResource(R.drawable.hebei);  
        break;  
    case "山西":  
        imgGroup.setImageResource(R.drawable.shanxi);  
        break;  
    case "陕西":  
        imgGroup.setImageResource(R.drawable.shaanxi);  
        break;  
    case "山东":  
        imgGroup.setImageResource(R.drawable.sd);  
        break;  
    case "新疆":  
        imgGroup.setImageResource(R.drawable.xj);  
        break;  
    case "西藏":  
        imgGroup.setImageResource(R.drawable.xz);  
        break;  
    case "青海":  
        imgGroup.setImageResource(R.drawable.qh);  
        break;  
    case "甘肃":  
        imgGroup.setImageResource(R.drawable.gs);  
        break;  
    case "宁夏":  
        imgGroup.setImageResource(R.drawable.nx);  
        break;  
    case "河南":  
        imgGroup.setImageResource(R.drawable.henan);  
        break;  
    case "江苏":  
        imgGroup.setImageResource(R.drawable.js);  
        break;  
    case "湖北":  
        imgGroup.setImageResource(R.drawable.hubei);  
        break;  
    case "浙江":  
        imgGroup.setImageResource(R.drawable.zj);  
        break;  
    case "安徽":  
        imgGroup.setImageResource(R.drawable.ah);  
        break;  
    case "福建":  
        imgGroup.setImageResource(R.drawable.ff);  
        break;
```

```

        case "江西":
            imgGroup.setImageResource(R.drawable.jx);
            break;
        case "湖南":
            imgGroup.setImageResource(R.drawable.hunan);
            break;
        case "贵州":
            imgGroup.setImageResource(R.drawable.gz);
            break;
        case "四川":
            imgGroup.setImageResource(R.drawable.sc);
            break;
        case "广东":
            imgGroup.setImageResource(R.drawable.gd);
            break;
        case "云南":
            imgGroup.setImageResource(R.drawable.yn);
            break;
        case "广西":
            imgGroup.setImageResource(R.drawable.gx);
            break;
        case "海南":
            imgGroup.setImageResource(R.drawable.hainan);
            break;
        case "香港":
            imgGroup.setImageResource(R.drawable.xg);
            break;
        case "澳门":
            imgGroup.setImageResource(R.drawable.am);
            break;
        case "台湾":
            imgGroup.setImageResource(R.drawable.tw);
            break;
        default:
            break;
    }

    return convertView;
}

```

/**

** Gets a View that displays the data for the given child within the given*

** group.*

** @param groupPosition the position of the group that contains the child*

```

* @param childPosition the position of the child (for which the View is
*
*                               returned) within the group
*
* @param isLastChild Whether the child is the last child within the group
*
* @param convertView the old view to reuse, if possible. You should check
*
*                               that this view is non-null and of an appropriate type before
*
*                               using. If it is not possible to convert this view to display
*
*                               the correct data, this method can create a new view. It is not
*
*                               guaranteed that the convertView will have been previously
*
*                               created by
*
*                               {@link #getChildView(int, int, boolean, View, ViewGroup)}.
*
* @param parent the parent that this view will eventually be attached to
*
* @return the View corresponding to the child at the specified position
*/
@Override
public View getChildView(int groupPosition, int childPosition, boolean isLastChild , View convertView, ViewGroup
parent) {
    if (null == convertView) {
        convertView = inflater.inflate(R.layout.item_second,parent,false);
    }
    TextView cityTV = (TextView)convertView.findViewById(R.id.second_group);
    // 设置二级列表的城市名字
    cityTV.setText(city_result.get(groupPosition).get(childPosition).getCity());
    return convertView;
}

/**
* Whether the child at the specified position is selectable.
*
*
* @param groupPosition the position of the group that contains the child
*
* @param childPosition the position of the child within the group
*
* @return whether the child is selectable.
*/
@Override
public boolean isChildSelectable(int groupPosition, int childPosition) {
    return true;
}
}

```

4. 背景图片的更换

背景图片的更换相对简单，在保存了省份名之后，在 MainActivity.java 找到背景图片对应的 id，使用 switch case 语句分别处理 34 个省份的图片将其替换即可，switch case 语句已经在上面的代码中实现，复制到 MainActivity.java 里粘贴即可。

四、一些不足

- 1、由于在选择城市界面中使用了高分辨率的图片，可能会导致点击时响应较慢。
- 2、应该使用 Intent 传递 cityname 与省份信息，但此处为了解安卓 Application 对象的生命周期，使用了 Application 类的静态变量，这是一个不好的开发风格，应该注意。