<https://blog.csdn.net/carrey1989/article/details/10227165#>

**Element.java:**

package com.example.androidtreeviewdemo.treeview;

/\*\*

\* Element类

\* @author carrey

\*

\*/

public class Element {

/\*\* 文字内容 \*/

private String contentText;

/\*\* 在tree中的层级 \*/

private int level;

/\*\* 元素的id \*/

private int id;

/\*\* 父元素的id \*/

private int parendId;

/\*\* 是否有子元素 \*/

private boolean hasChildren;

/\*\* item是否展开 \*/

private boolean isExpanded;

/\*\* 表示该节点没有父元素，也就是level为0的节点 \*/

public static final int NO\_PARENT = -1;

/\*\* 表示该元素位于最顶层的层级 \*/

public static final int TOP\_LEVEL = 0;

public Element(String contentText, int level, int id, int parendId,

boolean hasChildren, boolean isExpanded) {

super();

this.contentText = contentText;

this.level = level;

this.id = id;

this.parendId = parendId;

this.hasChildren = hasChildren;

this.isExpanded = isExpanded;

}

public boolean isExpanded() {

return isExpanded;

}

public void setExpanded(boolean isExpanded) {

this.isExpanded = isExpanded;

}

public String getContentText() {

return contentText;

}

public void setContentText(String contentText) {

this.contentText = contentText;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getParendId() {

return parendId;

}

public void setParendId(int parendId) {

this.parendId = parendId;

}

public boolean isHasChildren() {

return hasChildren;

}

public void setHasChildren(boolean hasChildren) {

this.hasChildren = hasChildren;

}

}

**TreeViewAdapter.java:**

package com.example.androidtreeviewdemo.treeview;

import java.util.ArrayList;

import com.example.androidtreeviewdemo.R;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.ImageView;

import android.widget.TextView;

/\*\*

\* TreeViewAdapter

\* @author carrey

\*

\*/

public class TreeViewAdapter extends BaseAdapter {

/\*\* 元素数据源 \*/

private ArrayList<Element> elementsData;

/\*\* 树中元素 \*/

private ArrayList<Element> elements;

/\*\* LayoutInflater \*/

private LayoutInflater inflater;

/\*\* item的行首缩进基数 \*/

private int indentionBase;

public TreeViewAdapter(ArrayList<Element> elements, ArrayList<Element> elementsData, LayoutInflater inflater) {

this.elements = elements;

this.elementsData = elementsData;

this.inflater = inflater;

indentionBase = 50;

}

public ArrayList<Element> getElements() {

return elements;

}

public ArrayList<Element> getElementsData() {

return elementsData;

}

@Override

public int getCount() {

return elements.size();

}

@Override

public Object getItem(int position) {

return elements.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

ViewHolder holder = null;

if (convertView == null) {

holder = new ViewHolder();

convertView = inflater.inflate(R.layout.treeview\_item, null);

holder.disclosureImg = (ImageView) convertView.findViewById(R.id.disclosureImg);

holder.contentText = (TextView) convertView.findViewById(R.id.contentText);

convertView.setTag(holder);

} else {

holder = (ViewHolder) convertView.getTag();

}

Element element = elements.get(position);

int level = element.getLevel();

holder.disclosureImg.setPadding(

indentionBase \* (level + 1),

holder.disclosureImg.getPaddingTop(),

holder.disclosureImg.getPaddingRight(),

holder.disclosureImg.getPaddingBottom());

holder.contentText.setText(element.getContentText());

if (element.isHasChildren() && !element.isExpanded()) {

holder.disclosureImg.setImageResource(R.drawable.close);

//这里要主动设置一下icon可见，因为convertView有可能是重用了"设置了不可见"的view，下同。

holder.disclosureImg.setVisibility(View.VISIBLE);

} else if (element.isHasChildren() && element.isExpanded()) {

holder.disclosureImg.setImageResource(R.drawable.open);

holder.disclosureImg.setVisibility(View.VISIBLE);

} else if (!element.isHasChildren()) {

holder.disclosureImg.setImageResource(R.drawable.close);

holder.disclosureImg.setVisibility(View.INVISIBLE);

}

return convertView;

}

/\*\*

\* 优化Holder

\* @author carrey

\*

\*/

static class ViewHolder{

ImageView disclosureImg;

TextView contentText;

}

}

**TreeViewItemClickListener.java:**

package com.example.androidtreeviewdemo.treeview;

import java.util.ArrayList;

import android.view.View;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

/\*\*

\* TreeView item点击事件

\* @author carrey

\*

\*/

public class TreeViewItemClickListener implements OnItemClickListener {

/\*\* adapter \*/

private TreeViewAdapter treeViewAdapter;

public TreeViewItemClickListener(TreeViewAdapter treeViewAdapter) {

this.treeViewAdapter = treeViewAdapter;

}

@Override

public void onItemClick(AdapterView<?> parent, View view, int position,

long id) {

//点击的item代表的元素

Element element = (Element) treeViewAdapter.getItem(position);

//树中的元素

ArrayList<Element> elements = treeViewAdapter.getElements();

//元素的数据源

ArrayList<Element> elementsData = treeViewAdapter.getElementsData();

//点击没有子项的item直接返回

if (!element.isHasChildren()) {

return;

}

if (element.isExpanded()) {

element.setExpanded(false);

//删除节点内部对应子节点数据，包括子节点的子节点...

ArrayList<Element> elementsToDel = new ArrayList<Element>();

for (int i = position + 1; i < elements.size(); i++) {

if (element.getLevel() >= elements.get(i).getLevel())

break;

elementsToDel.add(elements.get(i));

}

elements.removeAll(elementsToDel);

treeViewAdapter.notifyDataSetChanged();

} else {

element.setExpanded(true);

//从数据源中提取子节点数据添加进树，注意这里只是添加了下一级子节点，为了简化逻辑

int i = 1;//注意这里的计数器放在for外面才能保证计数有效

for (Element e : elementsData) {

if (e.getParendId() == element.getId()) {

e.setExpanded(false);

elements.add(position + i, e);

i ++;

}

}

treeViewAdapter.notifyDataSetChanged();

}

}

}

**MainActivity.java:**

package com.example.androidtreeviewdemo;

import java.util.ArrayList;

import com.example.androidtreeviewdemo.treeview.Element;

import com.example.androidtreeviewdemo.treeview.TreeViewAdapter;

import com.example.androidtreeviewdemo.treeview.TreeViewItemClickListener;

import android.os.Bundle;

import android.app.Activity;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.Menu;

import android.widget.ListView;

public class MainActivity extends Activity {

/\*\* 树中的元素集合 \*/

private ArrayList<Element> elements;

/\*\* 数据源元素集合 \*/

private ArrayList<Element> elementsData;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

LayoutInflater inflater = (LayoutInflater) getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

init();

ListView treeview = (ListView) findViewById(R.id.treeview);

TreeViewAdapter treeViewAdapter = new TreeViewAdapter(

elements, elementsData, inflater);

TreeViewItemClickListener treeViewItemClickListener = new TreeViewItemClickListener(treeViewAdapter);

treeview.setAdapter(treeViewAdapter);

treeview.setOnItemClickListener(treeViewItemClickListener);

}

private void init() {

elements = new ArrayList<Element>();

elementsData = new ArrayList<Element>();

//添加节点 -- 节点名称，节点level，节点id，父节点id，是否有子节点，是否展开

//添加最外层节点

Element e1 = new Element("山东省", Element.TOP\_LEVEL, 0, Element.NO\_PARENT, true, false);

//添加第一层节点

Element e2 = new Element("青岛市", Element.TOP\_LEVEL + 1, 1, e1.getId(), true, false);

//添加第二层节点

Element e3 = new Element("市南区", Element.TOP\_LEVEL + 2, 2, e2.getId(), true, false);

//添加第三层节点

Element e4 = new Element("香港中路", Element.TOP\_LEVEL + 3, 3, e3.getId(), false, false);

//添加第一层节点

Element e5 = new Element("烟台市", Element.TOP\_LEVEL + 1, 4, e1.getId(), true, false);

//添加第二层节点

Element e6 = new Element("芝罘区", Element.TOP\_LEVEL + 2, 5, e5.getId(), true, false);

//添加第三层节点

Element e7 = new Element("凤凰台街道", Element.TOP\_LEVEL + 3, 6, e6.getId(), false, false);

//添加第一层节点

Element e8 = new Element("威海市", Element.TOP\_LEVEL + 1, 7, e1.getId(), false, false);

//添加最外层节点

Element e9 = new Element("广东省", Element.TOP\_LEVEL, 8, Element.NO\_PARENT, true, false);

//添加第一层节点

Element e10 = new Element("深圳市", Element.TOP\_LEVEL + 1, 9, e9.getId(), true, false);

//添加第二层节点

Element e11 = new Element("南山区", Element.TOP\_LEVEL + 2, 10, e10.getId(), true, false);

//添加第三层节点

Element e12 = new Element("深南大道", Element.TOP\_LEVEL + 3, 11, e11.getId(), true, false);

//添加第四层节点

Element e13 = new Element("10000号", Element.TOP\_LEVEL + 4, 12, e12.getId(), false, false);

//添加初始树元素

elements.add(e1);

elements.add(e9);

//创建数据源

elementsData.add(e1);

elementsData.add(e2);

elementsData.add(e3);

elementsData.add(e4);

elementsData.add(e5);

elementsData.add(e6);

elementsData.add(e7);

elementsData.add(e8);

elementsData.add(e9);

elementsData.add(e10);

elementsData.add(e11);

elementsData.add(e12);

elementsData.add(e13);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

}

**treeview\_item.xml:**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<ImageView

android:id="@+id/disclosureImg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_alignParentLeft="true"/>

<TextView

android:id="@+id/contentText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_toRightOf="@id/disclosureImg"/>

</RelativeLayout>

**activity\_main.xml:**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<ListView

android:id="@+id/treeview"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

</RelativeLayout>