Step 6: 允许添加多个城市

目录

[目标 1](#_Toc448482725)

[步骤 1](#_Toc448482726)

[1、新建城市列表维护Activity 1](#_Toc448482727)

[2、在主界面上添加跳转到城市列表Activity的按钮 3](#_Toc448482728)

[3、修改CityListActivity的UI 4](#_Toc448482729)

[4、CityListActivity保存城市列表的方法 7](#_Toc448482730)

[5、CityListActivity添加城市方法 7](#_Toc448482731)

[6、CityListActivity删除城市方法 8](#_Toc448482732)

[7、在CityListActivity的Toolbar上添加按钮 9](#_Toc448482733)

## 目标

允许添加多个城市，设计添加、删除城市的Activity

知识点： ListView

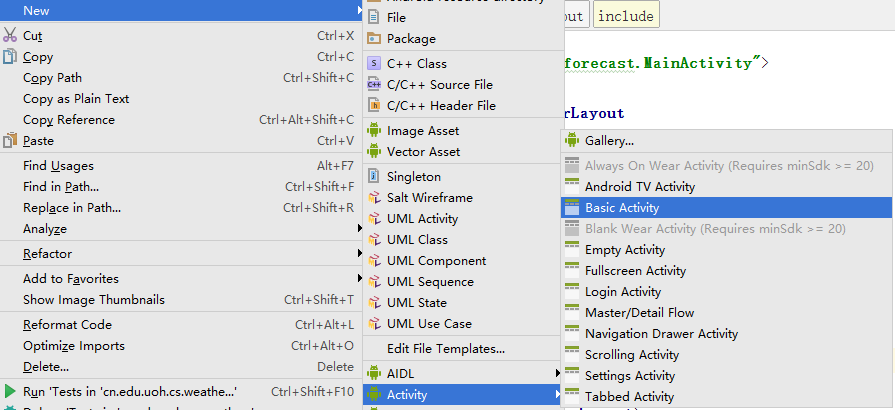
完成后的结果如下:



## 步骤

### 1、新建城市列表维护Activity

新建一个Basic Acivity，如下图



命名为CityListActivity，

删除activity\_city\_list.xml文件中的content\_city\_list和FloatingActionButton；

删除文件content\_city\_list.xml

删除CityListActivity类中FloatingActionButton的相关实现。

完成后的CityListActivity.java代码如下：

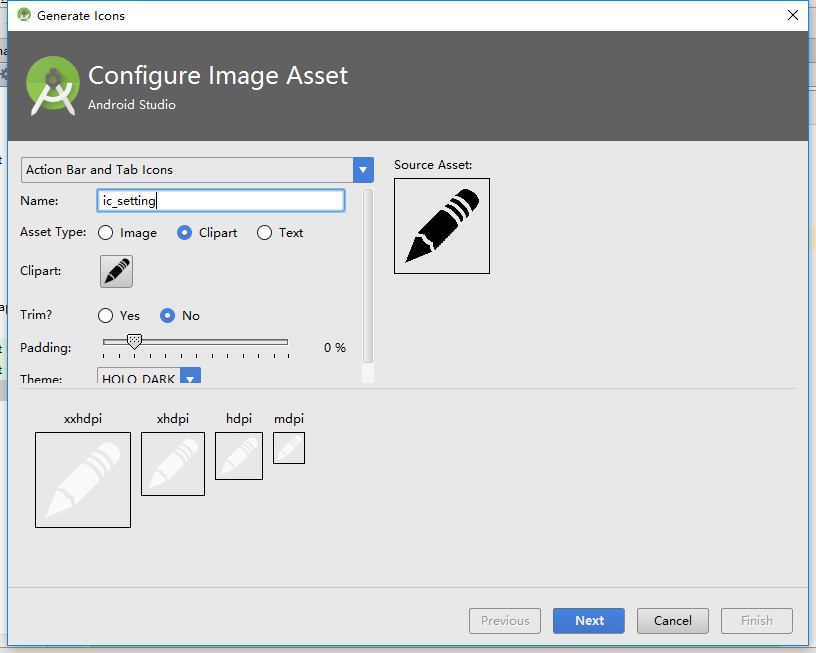
|  |
| --- |
| **public class** CityListActivity **extends** AppCompatActivity {  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_city\_list***);  Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  setSupportActionBar(toolbar);  getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  } } |

完成后的activity\_city\_list.xml如下

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:fitsSystemWindows="true"  tools:context="cn.edu.uoh.cs.weatherforecast.CityListActivity"**>   <**android.support.design.widget.AppBarLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:theme="@style/AppTheme.AppBarOverlay"**>   <**android.support.v7.widget.Toolbar  android:id="@+id/toolbar"  android:layout\_width="match\_parent"  android:layout\_height="?attr/actionBarSize"  android:background="?attr/colorPrimary"  app:popupTheme="@style/AppTheme.PopupOverlay"** />   </**android.support.design.widget.AppBarLayout**>  </**android.support.design.widget.CoordinatorLayout**> |

### 2、在主界面上添加跳转到城市列表Activity的按钮

1、添加按钮图标。



2、修改menu/menu\_main.xml文件，添加按钮代码如下:

|  |
| --- |
| <**menu xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  tools:context="cn.edu.uoh.cs.weatherforecast.MainActivity"**>   <**item  android:id="@+id/action\_settings"  android:icon="@drawable/ic\_setting"  android:title="设置"  app:showAsAction="ifRoom"**/> </**menu**> |

3、修改MainActivity中的onCreateOptionsMenu和onOptionsItemSelected方法，已加载菜单，响应按钮事件。

|  |
| --- |
| @Override **public boolean** onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.***menu\_main***, menu);  **return true**; }  @Override **public boolean** onOptionsItemSelected(MenuItem item) {  **int** id = item.getItemId();   *//noinspection SimplifiableIfStatement* **if** (id == R.id.***action\_settings***) {  Intent intent = **new** Intent(**this**, CityListActivity.**class**);  startActivity(intent);  **return true**;  }   **return super**.onOptionsItemSelected(item); } |

这时执行程序，已可以跳转到CityListActivity，并返回。

### 3、修改CityListActivity的UI

打开activity\_city\_list.xml，修改界面UI

1）用LinearLayout包裹所有元素；

2）添加城市输入框，和按钮，以输出城市和提交；

3）添加ListView，显示城市列表。

完成后代码如下:

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:fitsSystemWindows="true"  tools:context="cn.edu.uoh.cs.weatherforecast.CityListActivity"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**android.support.design.widget.AppBarLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:theme="@style/AppTheme.AppBarOverlay"**>   <**android.support.v7.widget.Toolbar  android:id="@+id/toolbar"  android:layout\_width="match\_parent"  android:layout\_height="?attr/actionBarSize"  android:background="?attr/colorPrimary"  app:popupTheme="@style/AppTheme.PopupOverlay"** />  </**android.support.design.widget.AppBarLayout**>   <**LinearLayout  android:id="@+id/addLayout"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:visibility="gone"**>   <**EditText  android:id="@+id/edtCity"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:width="0dp"  android:hint="请输入城市名称"  android:inputType="text"** />   <**Button  android:id="@+id/btnAdd"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:onClick="addCity"  android:text="添加"** />  </**LinearLayout**>   <**ListView  android:id="@+id/listView"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:choiceMode="singleChoice"  android:fillViewport="true"  android:listSelector="@color/highlighted\_text\_material\_dark"** />  </**LinearLayout**> </**android.support.design.widget.CoordinatorLayout**> |

4）在CityListActivity中添加方法ListView初始化方法InitListView

|  |
| --- |
| */\*\*  \* 初始化ListView，显示已保存的城市  \*/* **private void** InitListView() {  *// 获取保存的城市列表* SharedPreferences sharedPref = getPreferences(Context.***MODE\_PRIVATE***);  String cityNames = sharedPref.getString(MainActivity.***StoreListKey***, **"[蒙自]"**);  *// 转换字符串到数组* Gson gson = **new** Gson();  String[] cityArray = gson.fromJson(cityNames, String[].**class**);  *// 建立List adapter，simple\_list\_item\_1是已定义的layout* **adapter** = **new** ArrayAdapter<>(**this**, android.R.layout.***simple\_list\_item\_1***);  *// 添加城市列表* **adapter**.addAll(cityArray);  *// 设置ListView* ListView listView = (ListView) findViewById(R.id.***listView***);  listView.setAdapter(**adapter**); } |

注意：代码中变量adapter的类型是ArrayAdapter<String>，用来给ListView提供数据，因为要在多个方法中使用，所有把adapter定义成类的实例变量。

|  |
| --- |
| **public class** CityListActivity **extends** AppCompatActivity {  ArrayAdapter<String> **adapter**;  …  } |

5）修改CityListActivity中的方法onCreate，调用InitListView方法。

|  |
| --- |
| **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_city\_list***);  Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  setSupportActionBar(toolbar);  getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  **InitListView();** } |

### 4、CityListActivity保存城市列表的方法

添加保存城市列表的方法saveCityList

|  |
| --- |
| */\*\*  \* 保存城市列表，当添加、删除完成后调用此方法。  \*/* **private void** saveCityList() {  *// 获取城市列表* String[] cityArray = **new** String[**adapter**.getCount()];  **for** (**int** i = 0; i < cityArray.**length**; i++) {  cityArray[i] = **adapter**.getItem(i);  }  *// 转换成Json字符串* Gson gson = **new** Gson();  String cityJson = gson.toJson(cityArray);  *// 保存* SharedPreferences sharedPref = getPreferences(Context.***MODE\_PRIVATE***);  SharedPreferences.Editor editor = sharedPref.edit();  editor.putString(MainActivity.***StoreListKey***, cityJson);  editor.apply(); } |

### 5、CityListActivity添加城市方法

添加城市的方法addCity

|  |
| --- |
| */\*\*  \* 添加城市  \** ***@param view*** *view  \*/* **public void** addCity(View view) {  *// 取得输入框中的城市名称* EditText edtCity = (EditText) findViewById(R.id.***edtCity***);  String cityName = edtCity.getText().toString();  *// 如果输入不是空，添加到列表* **if** (cityName.trim().length() > 0) {  **adapter**.add(cityName);  saveCityList();  }  *// 清空输入* edtCity.setText(**""**);  *// 隐藏输入框* View addLayout = findViewById(R.id.***addLayout***);  addLayout.setVisibility(View.***GONE***); } |

### 6、CityListActivity删除城市方法

添加删除城市的方法removeCity

|  |
| --- |
| */\*\*  \* 删除选中的城市  \*/* **private void** removeCity() {  ListView listView = (ListView) findViewById(R.id.***listView***);  **int** position = listView.getCheckedItemPosition();  **if** (position < 0 || position >= **adapter**.getCount()) {  **return**;  }  String city = **adapter**.getItem(position);  **adapter**.remove(city);  saveCityList(); } |

### 7、在CityListActivity的Toolbar上添加按钮

1）添加两个图标(添加ic\_action\_add\_city、删除remove\_city)，参见步骤2.1

2）修改menu/menu\_city\_list.xml，设置按钮，代码如下:

|  |
| --- |
| <**menu xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  tools:context="cn.edu.uoh.cs.weatherforecast.CityListActivity"**>  <**item  android:id="@+id/add\_city"  android:icon="@drawable/ic\_action\_add\_city"  android:title="添加"  app:showAsAction="ifRoom"**/>  <**item  android:id="@+id/remove\_city"  android:icon="@drawable/ic\_remove"  android:title="删除"  app:showAsAction="ifRoom"**/> </**menu**> |

3）修改类CityListActivity.java，把新加的按钮显示在Toolbar上，在CityListActivity.java上加入两个方法，代码如下:

|  |
| --- |
| @Override **public boolean** onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.***menu\_city\_list***, menu);  **return true**; }  @Override **public boolean** onOptionsItemSelected(MenuItem item) {  **int** id = item.getItemId();  *// 单击删除按钮时删除选中的城市* **if** (id == R.id.***remove\_city***) {  removeCity();  **return true**;  }  *// 单击添加按钮是显示添加UI* **if** (id == R.id.***add\_city***) {  View view = findViewById(R.id.***addLayout***);  view.setVisibility(View.***VISIBLE***);  **return true**;  }  **return super**.onOptionsItemSelected(item); } |