RecyclerView 2

2018-11-14

이승진

**학습목표**

RecyclerView 객체를 사용하여 객체 목록을 화면에 표시하기

RecyclerView 항목 클릭 기능 구현

RecyclerView 항목 삭제 기능 구현

**목차**

[1. E05List 프로젝트 2](#_Toc530472436)

[1) 개요 2](#_Toc530472437)

[2) 프로젝트 생성 2](#_Toc530472438)

[2. 레이아웃 리소스 3](#_Toc530472439)

[1) activity\_main.xml 3](#_Toc530472440)

[2) RecyclerView 라이브러리 등록 5](#_Toc530472441)

[3) item.xml 파일 생성 7](#_Toc530472442)

[3. Java 구현 12](#_Toc530472443)

[1) Item.java 12](#_Toc530472444)

[2) MyRecyclerViewAdapter.java 14](#_Toc530472445)

[3) MainActivity.java 15](#_Toc530472446)

[4) 실행 16](#_Toc530472447)

[4. RecyclerView 항목 클릭 기능 구현 17](#_Toc530472448)

[1) MyRecyclerViewAdapter.java 수정 17](#_Toc530472449)

[2) 실행 19](#_Toc530472450)

[5. 삭제 기능 구현 20](#_Toc530472451)

[1) 체크 박스 추가 20](#_Toc530472452)

[2) MyRecyclerViewAdapter.java 수정 22](#_Toc530472453)

[3) 삭제 메뉴 구현 23](#_Toc530472454)

[4) MainActivity.java 25](#_Toc530472455)

# E05List 프로젝트

## 개요

지난 시간에는 RecyclerView를 이용하여 문자열(String) 목록을 화면에 표시하였다.

즉 데이터 항목은 문자열 한 개이었다.

이번 시간에는 데이터 항목에 문자열과 시각을 표시하자.

그러기 위해서는 데이터 항목에 해당하는 클래스를 구현해야 한다. (Item.java)

## 프로젝트 생성

메뉴: File - New - New Project

|  |  |
| --- | --- |
| Application name | E05List |
| Company Domain | skhu.net |

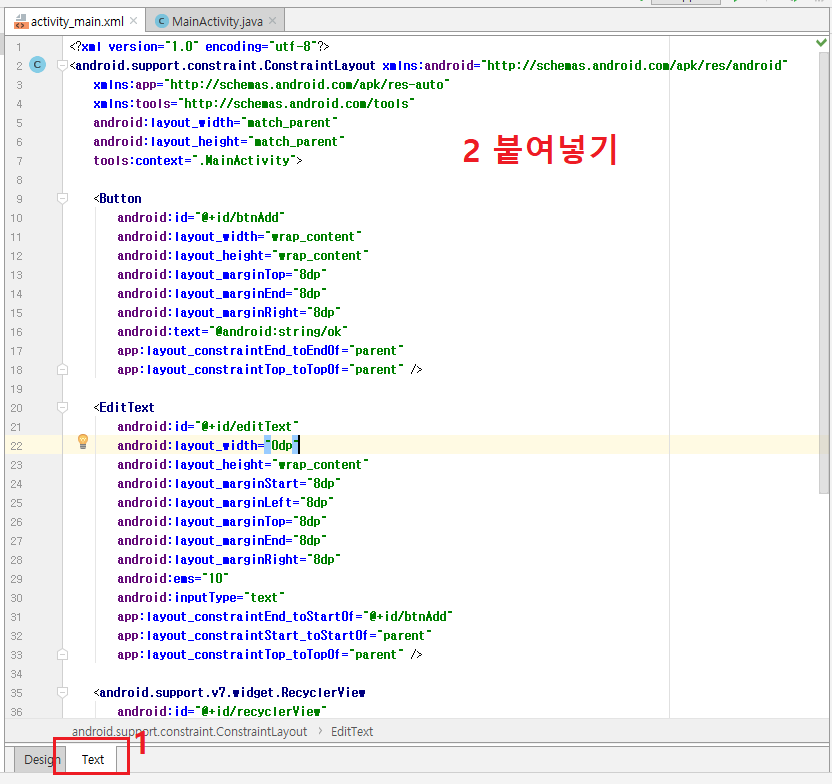
이전 예제들과 동일한 방법으로 새 프로젝트를 생성하자.

# 레이아웃 리소스

## activity\_main.xml

지난 시간에 만들었던 E04List 프로젝트의 activity\_main.xml 파일과 동일한 레이아웃 리소스를 만들려고 한다.

따라서 activity\_main.xml 파일의 소스코드를 복사해서 붙여 넣자.

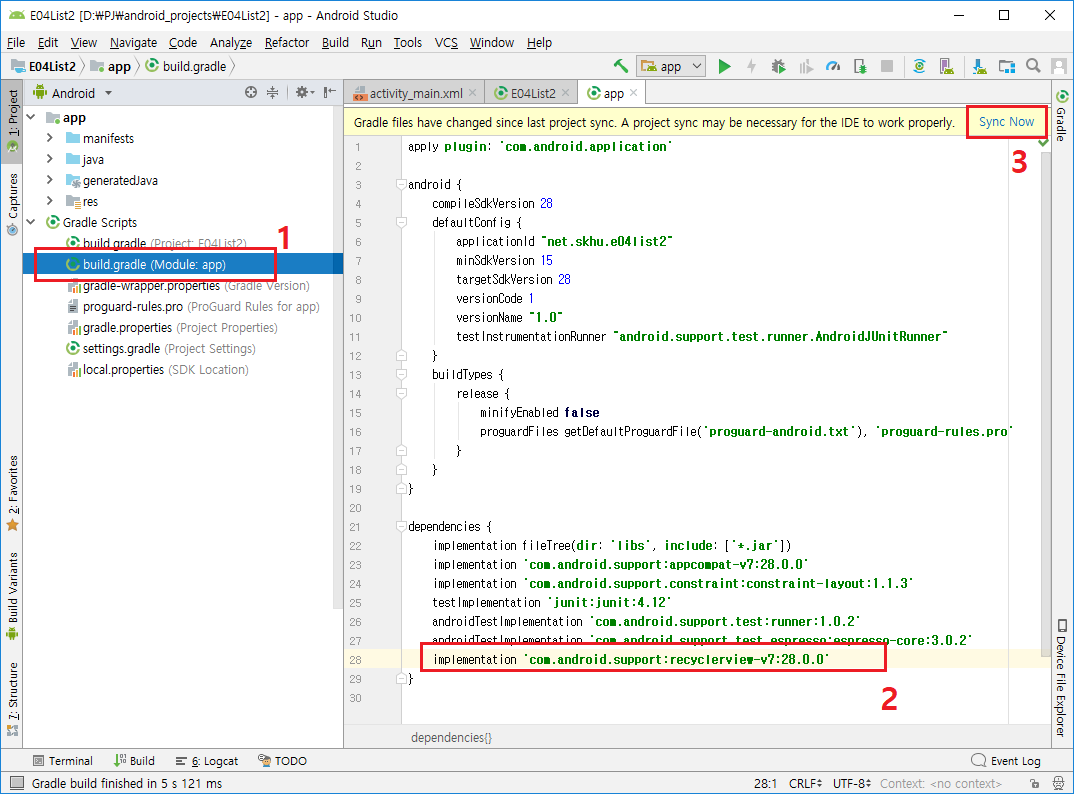


### activity\_main.xml 소스코드

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <android.support.constraint.ConstraintLayout 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"  tools:context=".MainActivity">   <Button  android:id="@+id/btnAdd"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  android:layout\_marginEnd="8dp"  android:layout\_marginRight="8dp"  android:text="@android:string/ok"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <EditText  android:id="@+id/editText"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_marginStart="8dp"  android:layout\_marginLeft="8dp"  android:layout\_marginTop="8dp"  android:layout\_marginEnd="8dp"  android:layout\_marginRight="8dp"  android:ems="10"  android:inputType="text"  app:layout\_constraintEnd\_toStartOf="@+id/btnAdd"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <android.support.v7.widget.RecyclerView  android:id="@+id/recyclerView"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:layout\_marginStart="8dp"  android:layout\_marginLeft="8dp"  android:layout\_marginEnd="8dp"  android:layout\_marginRight="8dp"  android:layout\_marginBottom="8dp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/editText" />  </android.support.constraint.ConstraintLayout> |

## RecyclerView 라이브러리 등록

### app 모듈 build.gradle 파일 수정



app 폴더 아래의 build.gradle 파일에서 dependencies 항목은

프로젝트에 필요한 라이브러리 목록이다.

RecyclerView를 사용하려면 필요한 라이브러리를 여기에 등록해야 한다.

레이아웃 리소스의 디자인 모드에서 RecyclerView를 드래그 드롭하여 추가할 때는

필요한 라이브러리가 자동으로 등록된다.

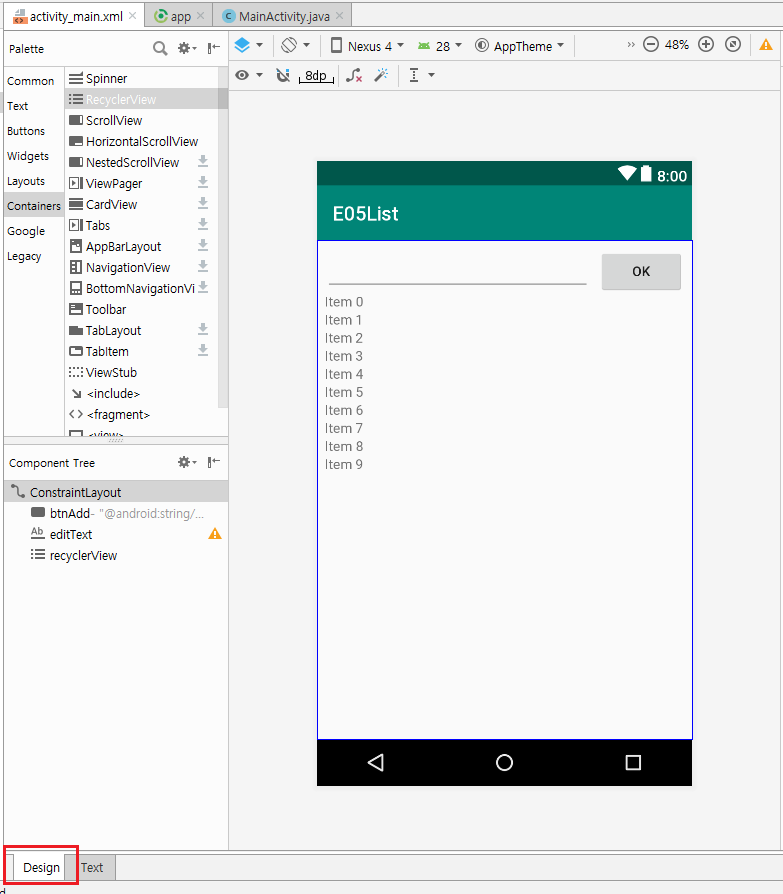
그런데, 레이아웃 리소스 XML 파일의 소스코드에 RecyclerView 항목을 직접 입력하거나 붙여 넣을 때는

필요한 라이브러리가 자동으로 등록되지 않는다.

그래서 직접 등록해 주어야 한다.

### build.gradle

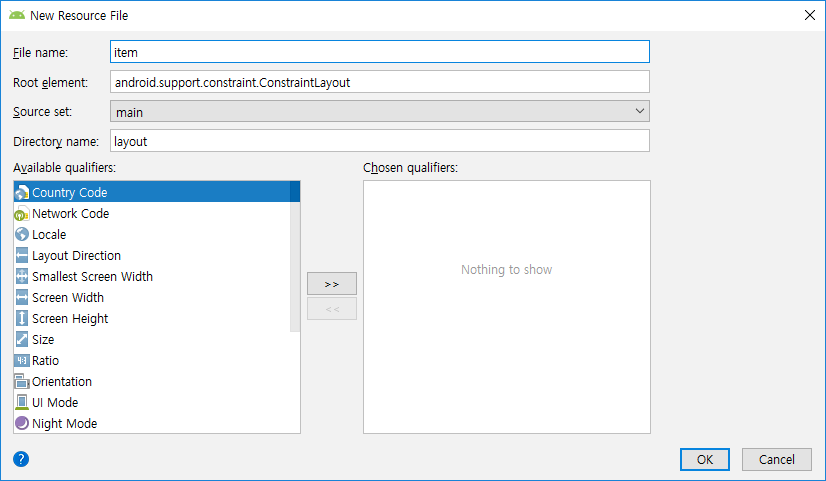
|  |
| --- |
| apply **plugin**: **'com.android.application'** android {  compileSdkVersion 28  defaultConfig {  applicationId **"net.skhu.e04list2"** minSdkVersion 15  targetSdkVersion 28  versionCode 1  versionName **"1.0"** testInstrumentationRunner **"android.support.test.runner.AndroidJUnitRunner"** }  buildTypes {  release {  minifyEnabled **false** proguardFiles getDefaultProguardFile(**'proguard-android.txt'**), **'proguard-rules.pro'** }  } }  dependencies {  implementation fileTree(**dir**: **'libs'**, **include**: [**'\*.jar'**])  implementation **'com.android.support:appcompat-v7:28.0.0'** implementation **'com.android.support.constraint:constraint-layout:1.1.3'** testImplementation **'junit:junit:4.12'** androidTestImplementation **'com.android.support.test:runner:1.0.2'** androidTestImplementation **'com.android.support.test.espresso:espresso-core:3.0.2'** implementation **'com.android.support:recyclerview-v7:28.0.0'** } |



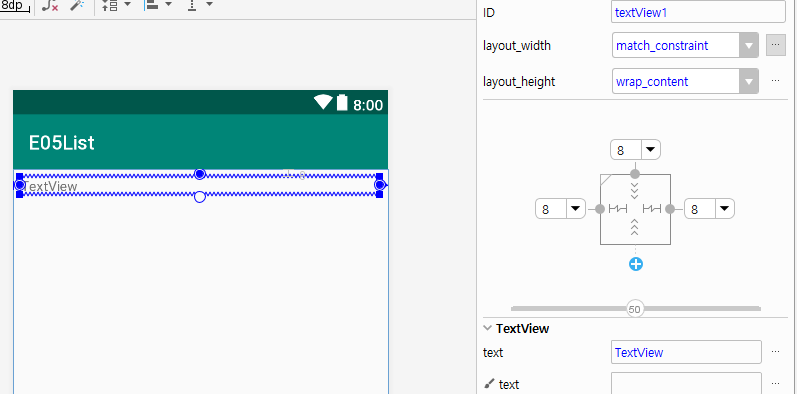
위 작업들을 잘 수행했다면, 위와 같이 보일 것이다.

## item.xml 파일 생성

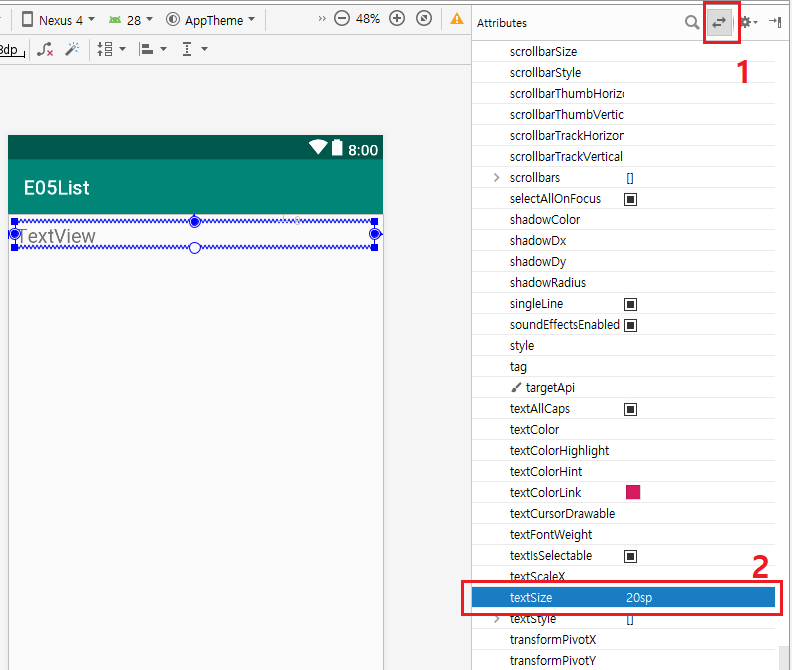
메뉴: File - New - Layout resource file



### TextView 추가

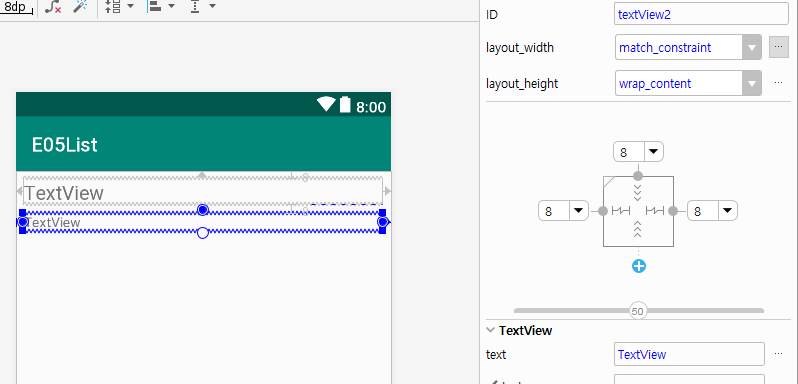


|  |  |
| --- | --- |
| 속성 | 값 |
| ID | textView1 |
| layout\_width | match\_constraint |
| layout\_height | warp\_content |

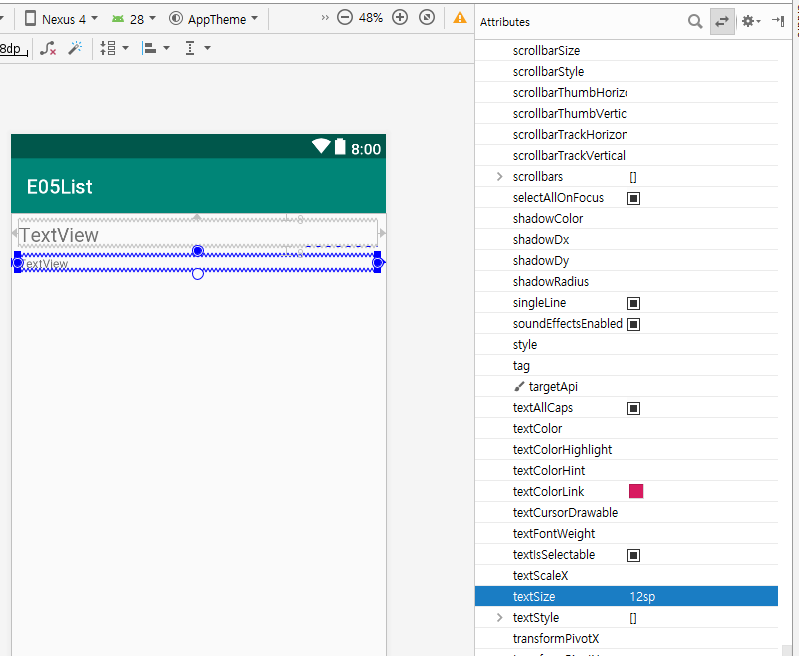


|  |  |
| --- | --- |
| 속성 | 값 |
| textSize | 20sp |

### TextView 한개 더 추가

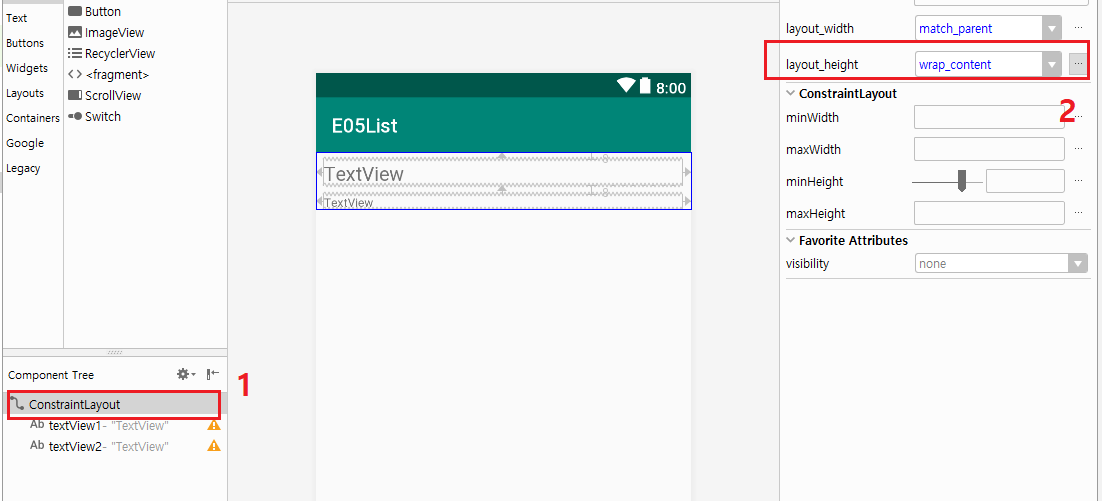


|  |  |
| --- | --- |
| 속성 | 값 |
| ID | textView2 |
| layout\_width | match\_constraint |
| layout\_height | warp\_content |



|  |  |
| --- | --- |
| 속성 | 값 |
| textSize | 12sp |

### ConstraintLayout 속성 변경



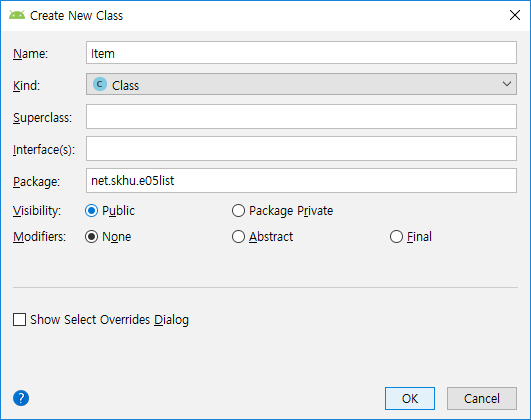
|  |  |
| --- | --- |
| 속성 | 값 |
| layout\_height | wrap\_content |

# Java 구현

## Item.java

데이터 항목에 해당하는 클래스를 구현하자.

메뉴: File - New - Java Class



|  |  |
| --- | --- |
| Name | Item |
| Package | net.skhu.e05list |

### Item.java

|  |
| --- |
| **package** net.skhu.e05list;  **import** java.text.SimpleDateFormat; **import** java.util.Date;  **public class** Item {  String **title**;  Date **createTime**;   **static** SimpleDateFormat *format* = **new** SimpleDateFormat(**"yyyy-MM-dd HH:mm:ss"**);   **public** Item(String title) {  **this**.**title** = title;  **this**.**createTime** = **new** Date();  }   **public** String getTitle() {  **return title**;  }   **public** Date getCreateTime() {  **return createTime**;  }   **public** String getCreateTimeFormatted() {  **return** *format*.format(**createTime**);  } } |

createTime 멤버 변수는 작성 시각을 저장한다.

createTime 값으로부터 "2018-11-15 13:20:35" 형태의 문자열을 생성하기 위ㅐ

SimpleDateFormat 클래스를 사용하여 구현하였다.

## MyRecyclerViewAdapter.java

|  |
| --- |
| **package** net.skhu.e05list;  **import** android.content.Context; **import** android.support.v7.widget.RecyclerView; **import** android.text.Layout; **import** android.util.Log; **import** android.view.LayoutInflater; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.TextView;  **import** java.util.ArrayList;  **public class** MyRecyclerViewAdapter **extends** RecyclerView.Adapter<MyRecyclerViewAdapter.ViewHolder> {   **class** ViewHolder **extends** RecyclerView.ViewHolder {  TextView **textView1**, **textView2**;   **public** ViewHolder(View view) {  **super**(view);  **textView1** = view.findViewById(R.id.***textView1***);  **textView2** = view.findViewById(R.id.***textView2***);  }   **public void** setData() {  Item item = **arrayList**.get(getAdapterPosition());  **textView1**.setText(item.getTitle());  **textView2**.setText(item.getCreateTimeFormatted());  }  }   LayoutInflater **layoutInflater**;  ArrayList<Item> **arrayList**;   **public** MyRecyclerViewAdapter(Context context, ArrayList<Item> arrayList) {  **this**.**layoutInflater** = LayoutInflater.*from*(context);  **this**.**arrayList** = arrayList;  }   @Override  **public int** getItemCount() {  **return arrayList**.size();  }   @Override  **public** ViewHolder onCreateViewHolder(ViewGroup viewGroup, **int** viewType) {  View view = **layoutInflater**.inflate(R.layout.***item***, viewGroup, **false**);  **return new** ViewHolder(view);  }   @Override  **public void** onBindViewHolder(**final** ViewHolder viewHolder, **final int** index) {  viewHolder.setData();  } } |

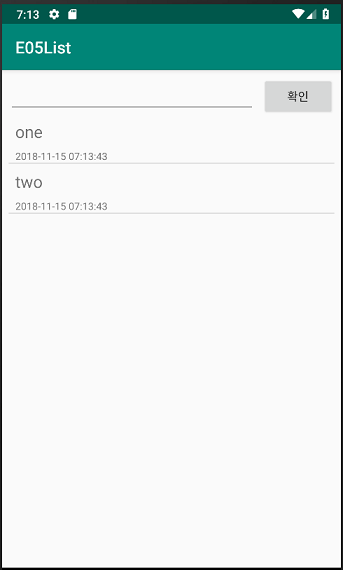
지난 시간에 구현했던 MyRecyclerViewAdapter 소스코드와 비교해서 달라진 부분을 살펴보자.

## MainActivity.java

|  |
| --- |
| **package** net.skhu.e05list;  **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.support.v7.widget.DefaultItemAnimator; **import** android.support.v7.widget.DividerItemDecoration; **import** android.support.v7.widget.LinearLayoutManager; **import** android.support.v7.widget.RecyclerView; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText;  **import** java.util.ArrayList;  **public class** MainActivity **extends** AppCompatActivity {   MyRecyclerViewAdapter **myRecyclerViewAdapter**;  ArrayList<Item> **arrayList**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **arrayList** = **new** ArrayList<Item>();  **arrayList**.add(**new** Item(**"one"**));  **arrayList**.add(**new** Item(**"two"**));   **myRecyclerViewAdapter** = **new** MyRecyclerViewAdapter(**this**, **arrayList**);  RecyclerView recyclerView = (RecyclerView) findViewById(R.id.***recyclerView***);  recyclerView.addItemDecoration(**new** DividerItemDecoration(**this**, DividerItemDecoration.***VERTICAL***));  recyclerView.setLayoutManager(**new** LinearLayoutManager(**this**));  recyclerView.setItemAnimator(**new** DefaultItemAnimator());  recyclerView.setAdapter(**myRecyclerViewAdapter**);   Button b = (Button)findViewById(R.id.***btnAdd***);  b.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View arg0) {  EditText e = (EditText) findViewById(R.id.***editText***);  String s = e.getText().toString();  e.setText(**""**);  **arrayList**.add(**new** Item(s));  **myRecyclerViewAdapter**.notifyDataSetChanged();  }  });   } } |

지난 시간에 구현했던 MainActivity 소스코드와 비교해서 달라진 부분을 살펴보자.

## 실행



# RecyclerView 항목 클릭 기능 구현

## MyRecyclerViewAdapter.java 수정

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 | **package** net.skhu.e05list;  **import** android.content.Context; **import** android.support.v7.widget.RecyclerView; **import** android.text.Layout; **import** android.util.Log; **import** android.view.LayoutInflater; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.TextView; **import** android.widget.Toast;  **import** java.util.ArrayList;  **public class** MyRecyclerViewAdapter **extends** RecyclerView.Adapter<MyRecyclerViewAdapter.ViewHolder> {   **class** ViewHolder **extends** RecyclerView.ViewHolder **implements** View.OnClickListener {  TextView **textView1**, **textView2**;   **public** ViewHolder(View view, ArrayList<Item> arrayList) {  **super**(view);  **this**.**textView1** = view.findViewById(R.id.***textView1***);  **this**.**textView2** = view.findViewById(R.id.***textView2***);  view.setOnClickListener(**this**);  }   **public void** setData() {  Item item = **arrayList**.get(**super**.getAdapterPosition());  **this**.**textView1**.setText(item.getTitle());  **this**.**textView2**.setText(item.getCreateTimeFormatted());  }   @Override  **public void** onClick(View view) {  Item item = **arrayList**.get(**super**.getAdapterPosition());  String s = String.*format*(**"index: %d, title: %s"**, **super**.getAdapterPosition(), item.getTitle());  Toast.*makeText*(view.getContext(), s, Toast.***LENGTH\_SHORT***).show();  }  }   LayoutInflater **layoutInflater**;  ArrayList<Item> **arrayList**;   **public** MyRecyclerViewAdapter(Context context, ArrayList<Item> arrayList) {  **this**.**layoutInflater** = LayoutInflater.*from*(context);  **this**.**arrayList** = arrayList;  }   @Override  **public int** getItemCount() {  **return arrayList**.size();  }   @Override  **public** ViewHolder onCreateViewHolder(ViewGroup viewGroup, **int** viewType) {  View view = **layoutInflater**.inflate(R.layout.***item***, viewGroup, **false**);  **return new** ViewHolder(view, **arrayList**);  }   @Override  **public void** onBindViewHolder(**final** ViewHolder viewHolder, **final int** index) {  viewHolder.setData();  } } |

### ViewHolder 클래스에 View.OnClickListener 인터페이스를 구현한 이유

ViewHolder 클래스에 View.OnClickListener 인터페이스를 구현하였다.

지금까지 어노니머스 이너 클래스 문법을 사용하여 리스너 클래스를 구현했는데,

반드시 그래야 하는 것은 아니다.

리스너 클래스는 그냥 평범한 클래스로 구현해도 된다.

그리고 기존의 클래스들 중 하나가 리스너 클래스의 역할도 같이 담당해도 된다.

RecyclerView 에서 데이터 항목 리스너 객체는, 데이터 항목 뷰 객체 마다 하나씩 달려있어야 한다.

즉 RecyclerView 에서 데이터 항목 리스너 객체의 수는, 데이터 항목 뷰 객체 의 수와 일치해야 한다.

데이터 항목 뷰 객체란, MyRecyclerViewAdapter 클래스의 onCreateViewHolder 메소드에서 레이아웃 인플레이션으로 생성된 뷰(view) 객체이다. (줄59)

|  |
| --- |
| View view = **layoutInflater**.inflate(R.layout.***item***, viewGroup, **false**); |

이 데이터 항목 뷰 객체는 ViewHolder 객체에 들어있다. (줄60)

|  |
| --- |
| **return new** ViewHolder(view, **arrayList**); |

즉 데이터 항목 뷰 객체의 수와 ViewHolder 객체의 수가 일치한다.

ViewHolder 객체의 수 == 데이터 항목 뷰 객치의 수 == 데이터 항목 리스너 객체의 수

위 3 객체의 수는 동일해야 한다.

따라서 리스너 클래스를 새로 구현해서 리스너 객체를 새로 생성하는 것 보다

이미 생성된 ViewHolder 객체들이 리스너 객체의 역할도 같이 담당하는 편이 효율적이다.

### 기존의 클래스에 View.OnClickListener 인터페이스 구현하기

(1) implements View.OnClickListener 이 필요하다. (줄17)

|  |
| --- |
| **static class** ViewHolder **extends** RecyclerView.ViewHolder **implements** View.OnClickListener { |

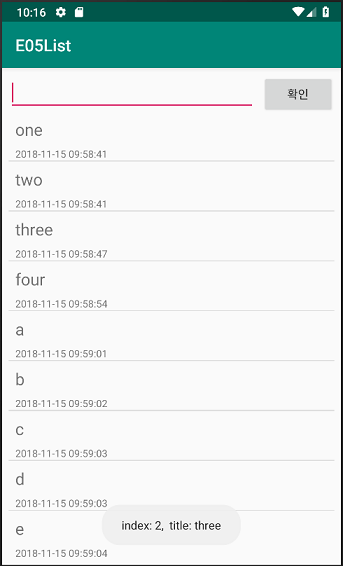
(2) onClick 메소드를 재정의해야 한다. (줄35)

|  |
| --- |
| @Override **public void** onClick(View view) {  Item item = **arrayList**.get(**super**.getAdapterPosition());  String s = String.*format*(**"index: %d, title: %s"**, **super**.getAdapterPosition(), item.getTitle());  Toast.*makeText*(view.getContext(), s, Toast.***LENGTH\_SHORT***).show(); } |

(3) 데이터 항목 뷰 객체에 리스너 객체를 등록해야 한다. (줄26)

|  |
| --- |
| view.setOnClickListener(**this**); |

## 실행

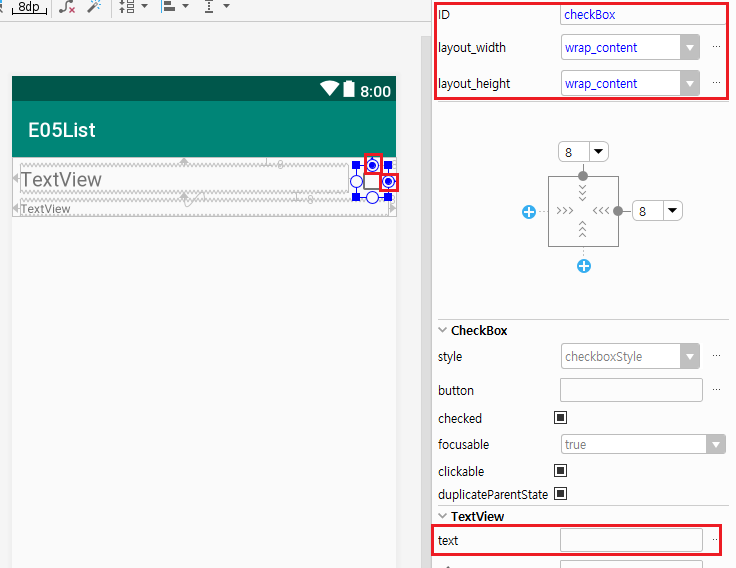


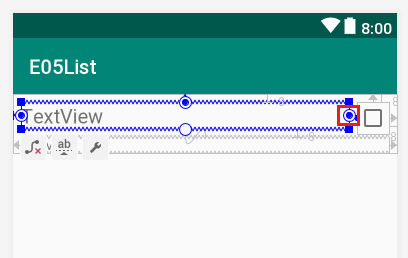
# 삭제 기능 구현

## 체크 박스 추가

삭제할 항목을 선택하기 위한 체크박스를 추가하자.

### item.xml 수정





### Item.java 수정

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | **package** net.skhu.e05list;  **import** java.text.SimpleDateFormat; **import** java.util.Date;  **public class** Item {  String **title**;  Date **createTime**;  **boolean checked**;   **static** SimpleDateFormat *format* = **new** SimpleDateFormat(**"yyyy-MM-dd HH:mm:ss"**);   **public** Item(String title) {  **this**.**title** = title;  **this**.**createTime** = **new** Date();  }   **public** String getTitle() {  **return title**;  }   **public** Date getCreateTime() {  **return createTime**;  }   **public** String getCreateTimeFormatted() {  **return** *format*.format(**createTime**);  }   **public boolean** isChecked() { **return checked**; }  **public void** setChecked(**boolean** checked) { **this**.**checked** = checked; } } |

앞에서 만든 체크박스의 상태 값을 저장할 멤버 변수를 구현하자.

boolean 타입의 get 메소드 이름은 get으로 시작해도 되지만, is로 시작하는 것이 더 바람직하다.

즉 getChecked/setChecked 이름도 가능하지만, isChecked/setChecked 이름이 더 바람직하다.

## MyRecyclerViewAdapter.java 수정

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | **package** net.skhu.e05list;  **import** android.content.Context; **import** android.support.v7.widget.RecyclerView; **import** android.text.Layout; **import** android.util.Log; **import** android.view.LayoutInflater; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.CheckBox; **import** android.widget.CompoundButton; **import** android.widget.TextView; **import** android.widget.Toast;  **import** java.util.ArrayList;  **public class** MyRecyclerViewAdapter **extends** RecyclerView.Adapter<MyRecyclerViewAdapter.ViewHolder> {   **class** ViewHolder **extends** RecyclerView.ViewHolder **implements** CompoundButton.OnCheckedChangeListener {  TextView **textView1**, **textView2**;  CheckBox **checkBox**;   **public** ViewHolder(View view) {  **super**(view);  **this**.**textView1** = view.findViewById(R.id.***textView1***);  **this**.**textView2** = view.findViewById(R.id.***textView2***);  **this**.**checkBox** = view.findViewById(R.id.***checkBox***);  **this**.**checkBox**.setOnCheckedChangeListener(**this**);  }   **public void** setData() {  Item item = **arrayList**.get(**super**.getAdapterPosition());  **this**.**textView1**.setText(item.getTitle());  **this**.**textView2**.setText(item.getCreateTimeFormatted());  }   @Override  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  Item item = **arrayList**.get(**super**.getAdapterPosition());  item.setChecked(isChecked);  }  }   LayoutInflater **layoutInflater**;  ArrayList<Item> **arrayList**;   **public** MyRecyclerViewAdapter(Context context, ArrayList<Item> arrayList) {  **this**.**layoutInflater** = LayoutInflater.*from*(context);  **this**.**arrayList** = arrayList;  }   @Override  **public int** getItemCount() {  **return arrayList**.size();  }   @Override  **public** ViewHolder onCreateViewHolder(ViewGroup viewGroup, **int** viewType) {  View view = **layoutInflater**.inflate(R.layout.***item***, viewGroup, **false**);  **return new** ViewHolder(view);  }   @Override  **public void** onBindViewHolder(**final** ViewHolder viewHolder, **final int** index) {  viewHolder.setData();  } } |

### 체크 박스 리스너 구현

체크 박스를 클릭하여 체크 상태가 변경된 경우에, Item 데이터 항목 객체의 checked 속성에 이 값을 저장해야 한다.

그러려면 체크 박스에 OnCheckedChangedListener 객체를 등록해야 한다.

체크 박스의 수와 리스너 객체의 수가 정확히 일치해야 바람직하므로,

ViewHolder 객체가 리스너 객체의 역할도 동시에 담당하는 것이 바람직하다.

그래서 ViewHolder 클래스에 OnCheckedChangedListener 인터페이스를 구현하였다.

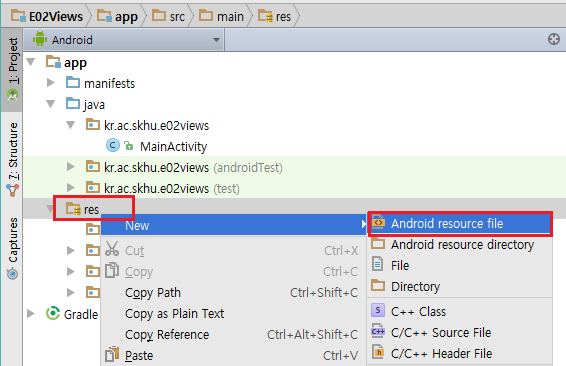
## 삭제 메뉴 구현

### strings.xml 수정

|  |
| --- |
| <**resources**>  <**string name="app\_name"**>E05List</**string**>  <**string name="action\_remove"**>삭제</**string**> </**resources**> |

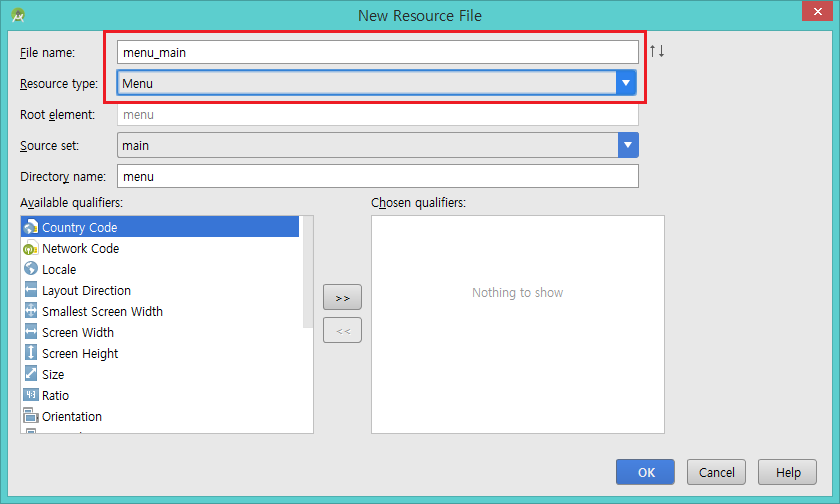
메뉴 항목에 표시할 문자열 리소스 생성

### 메뉴 리소스 파일 생성



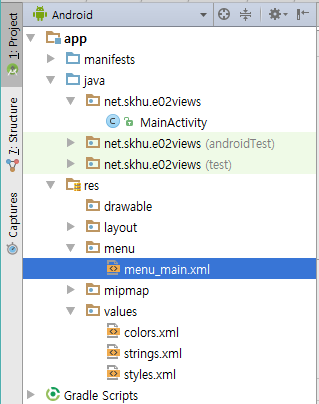
화면 왼쪽 Project 창에서 res 폴더를 마우스 오른쪽 버튼으로 클릭하고

메뉴에서 New - Android resource file 클릭



액티비티의 이름이 MainActivity 이므로, 메뉴 파일의 이름도 menu\_main 이라고 입력한다.

Ok 버튼 클릭.



res/menu/menu\_main.xml 파일이 생성된 메뉴 리소스 XML 파일이다.

이 파일을 더블 클릭하여 열자.

아래쪽의 text 탭을 눌러서 텍스트 편집창을 열자.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | *<?*xml version="1.0" encoding="utf-8"*?>*  <menu xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto">  <item android:id="@+id/action\_remove"  android:title="@string/action\_remove" app:showAsAction="always" />  </menu> |

## MainActivity.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71 | **package** net.skhu.e05list;  **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.support.v7.widget.DefaultItemAnimator; **import** android.support.v7.widget.DividerItemDecoration; **import** android.support.v7.widget.LinearLayoutManager; **import** android.support.v7.widget.RecyclerView; **import** android.support.v7.widget.helper.ItemTouchHelper; **import** android.view.Menu; **import** android.view.MenuItem; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.Toast;  **import** java.util.ArrayList;  **public class** MainActivity **extends** AppCompatActivity {   MyRecyclerViewAdapter **myRecyclerViewAdapter**;  ArrayList<Item> **arrayList**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **arrayList** = **new** ArrayList<Item>();  **arrayList**.add(**new** Item(**"one"**));  **arrayList**.add(**new** Item(**"two"**));   **myRecyclerViewAdapter** = **new** MyRecyclerViewAdapter(**this**, **arrayList**);  RecyclerView recyclerView = (RecyclerView) findViewById(R.id.***recyclerView***);  recyclerView.addItemDecoration(**new** DividerItemDecoration(**this**, DividerItemDecoration.***VERTICAL***));  recyclerView.setLayoutManager(**new** LinearLayoutManager(**this**));  recyclerView.setItemAnimator(**new** DefaultItemAnimator());  recyclerView.setAdapter(**myRecyclerViewAdapter**);   Button b = (Button)findViewById(R.id.***btnAdd***);  b.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View arg0) {  EditText e = (EditText) findViewById(R.id.***editText***);  String s = e.getText().toString();  e.setText(**""**);  **arrayList**.add(**new** Item(s));  **myRecyclerViewAdapter**.notifyDataSetChanged();  }  });  }   @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();  **if** (id == R.id.***action\_remove***) {  **for** (**int** i = **arrayList**.size() - 1; i >= 0; --i)  **if** (**arrayList**.get(i).isChecked())  **arrayList**.remove(i);  **myRecyclerViewAdapter**.notifyDataSetChanged();  **return true**;  }  **return super**.onOptionsItemSelected(item);  }  } |

### 선택된 항목 삭제하기 구현

"삭제" 메뉴가 클릭되면, 아래 코드가 실행된다.

|  |
| --- |
| **if** (id == R.id.***action\_remove***) {  **for** (**int** i = **arrayList**.size() - 1; i >= 0; --i)  **if** (**arrayList**.get(i).isChecked())  **arrayList**.remove(i);  **myRecyclerViewAdapter**.notifyDataSetChanged();  **return true**; } |

위 코드에서 for 루프가 역방향 탐색을 하고 있음에 주의하자,

목록을 순방향 탐색을 하면서 항목을 삭제하는 경우에,

삭제하자마자 항목들이 한 칸씩 앞으로 당겨진다. 그래서 그 다음 탐색할 항목이 한 칸 앞으로 당겨져서

현재 항목이 된다. 그런데 ++i를 하게 되어서, 한 칸 앞으로 당겨진 현재 항목 다음 칸을 탐색하게 된다.

즉 삭제한 항목은 if 문으로 검사하지 않고 건너뛰게 된다.

이 문제를 해결하기 위해서, 역방향 탐색을 하였다.

역방향 탐색에서는 이런 문제가 발생하지 않는다.

항목들을 삭제한 후 아답타 객체의 notifyDataSetChanged() 메소드를 호출해야 화면이 다시 그려진다.

# 메뉴 상태 변경

체크된 항목이 있을 때에만, 삭제 메뉴가 표시되도록 수정하자.

## MyRecyclerViewAdapter.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72 | **package** net.skhu.e05list;  **import** android.app.Activity; **import** android.content.Context; **import** android.support.v7.widget.RecyclerView; **import** android.text.Layout; **import** android.util.Log; **import** android.view.LayoutInflater; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.CheckBox; **import** android.widget.CompoundButton; **import** android.widget.TextView; **import** android.widget.Toast;  **import** java.util.ArrayList;  **public class** MyRecyclerViewAdapter **extends** RecyclerView.Adapter<MyRecyclerViewAdapter.ViewHolder> {   **class** ViewHolder **extends** RecyclerView.ViewHolder **implements** CompoundButton.OnCheckedChangeListener {  TextView **textView1**, **textView2**;  CheckBox **checkBox**;   **public** ViewHolder(View view) {  **super**(view);  **this**.**textView1** = view.findViewById(R.id.***textView1***);  **this**.**textView2** = view.findViewById(R.id.***textView2***);  **this**.**checkBox** = view.findViewById(R.id.***checkBox***);  **this**.**checkBox**.setOnCheckedChangeListener(**this**);  }   **public void** setData() {  Item item = **arrayList**.get(**super**.getAdapterPosition());  **this**.**textView1**.setText(item.getTitle());  **this**.**textView2**.setText(item.getCreateTimeFormatted());  }   @Override  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  Item item = **arrayList**.get(**super**.getAdapterPosition());  item.setChecked(isChecked);  **if** (isChecked) ++**checkedItemCount**; **else** --**checkedItemCount**;  **if** (**checkedItemCount** <= 1)  ((Activity) **textView1**.getContext()).invalidateOptionsMenu();  }  }   LayoutInflater **layoutInflater**;  ArrayList<Item> **arrayList**;  **int checkedItemCount** = 0;   **public** MyRecyclerViewAdapter(Context context, ArrayList<Item> arrayList) {  **this**.**layoutInflater** = LayoutInflater.*from*(context);  **this**.**arrayList** = arrayList;  }   @Override  **public int** getItemCount() {  **return arrayList**.size();  }   @Override  **public** ViewHolder onCreateViewHolder(ViewGroup viewGroup, **int** viewType) {  View view = **layoutInflater**.inflate(R.layout.***item***, viewGroup, **false**);  **return new** ViewHolder(view);  }   @Override  **public void** onBindViewHolder(**final** ViewHolder viewHolder, **final int** index) {  viewHolder.setData();  } } |

## MainActivity.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73 | **package** net.skhu.e05list;  **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.support.v7.widget.DefaultItemAnimator; **import** android.support.v7.widget.DividerItemDecoration; **import** android.support.v7.widget.LinearLayoutManager; **import** android.support.v7.widget.RecyclerView; **import** android.support.v7.widget.helper.ItemTouchHelper; **import** android.view.Menu; **import** android.view.MenuItem; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.Toast;  **import** java.util.ArrayList;  **public class** MainActivity **extends** AppCompatActivity {   MyRecyclerViewAdapter **myRecyclerViewAdapter**;  ArrayList<Item> **arrayList**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **arrayList** = **new** ArrayList<Item>();  **arrayList**.add(**new** Item(**"one"**));  **arrayList**.add(**new** Item(**"two"**));   **myRecyclerViewAdapter** = **new** MyRecyclerViewAdapter(**this**, **arrayList**);  RecyclerView recyclerView = (RecyclerView) findViewById(R.id.***recyclerView***);  recyclerView.addItemDecoration(**new** DividerItemDecoration(**this**, DividerItemDecoration.***VERTICAL***));  recyclerView.setLayoutManager(**new** LinearLayoutManager(**this**));  recyclerView.setItemAnimator(**new** DefaultItemAnimator());  recyclerView.setAdapter(**myRecyclerViewAdapter**);   Button b = (Button)findViewById(R.id.***btnAdd***);  b.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View arg0) {  EditText e = (EditText) findViewById(R.id.***editText***);  String s = e.getText().toString();  e.setText(**""**);  **arrayList**.add(**new** Item(s));  **myRecyclerViewAdapter**.notifyDataSetChanged();  }  });  }   @Override  **public boolean** onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.***menu\_main***, menu);  MenuItem menuItem = menu.findItem(R.id.***action\_remove***);  menuItem.setVisible(**myRecyclerViewAdapter**.**checkedItemCount** > 0);  **return true**;  }   @Override **public boolean** onOptionsItemSelected(MenuItem item) {  **int** id = item.getItemId();  **if** (id == R.id.***action\_remove***) {  **for** (**int** i = **arrayList**.size() - 1; i >= 0; --i)  **if** (**arrayList**.get(i).isChecked())  **arrayList**.remove(i);  **myRecyclerViewAdapter**.notifyDataSetChanged();  **return true**;  }  **return super**.onOptionsItemSelected(item);  }  } |