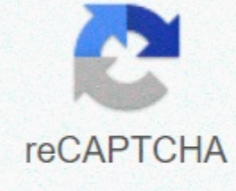




I'm not robot



Continue

Adapter class in android studio

How android ListView with adapter works is not so clear to me until I read this amazing examples and lucas rocha article: Performance tips for Android ListView.Today I'm going to show the code I use when I wanted the user to select an item in alertDialog from ListView. If you are looking for an example of AlertDialog with a simple ListView (no tab), here's a blog entry I made: Android AlertDialog from ListView.A The ListView item here contains the text (item name) and ID (item ID), so when the user selects an item in a pop-up window, you'll be able to get the text and ID of that selected item. This is very useful if the items come from a database - records with IDs and names. Tags (item IDs) were set using the textView.setTag() method. DemoHere video is the ultimate output of our code for today. The name and ID were retrieved and shown with a toast. Download CodeYou can download the code we used here: ListViewPopUpWithAdapter.zipLet's Code! We only need 6 files to successfully run this sample code.1. activity_main.xml – our main layout. Our trigger to show listView is the button. <RelativeLayout xmlns:android= xmlns:tools= tools:layout_width=match_parent android:layout_height=match_parent android:paddingbottom=@dimen/activity_vertical_margin android:paddingleft=@dimen/activity_horizontal_margin android:paddingright=@dimen/activity_horizontal_margin android:paddingtop=@dimen/activity_vertical_margin tools:context=. MainActivity> <Button android:id=@+id/buttonShowPopUp android:layout_width=wrap_content android:layout_height=wrap_content android:layout_centerinparent=true android:text=Show AlertDialog with ListView></Button> </RelativeLayout>2. MainActivity.java – will show the main layout, the button in the middle of the screen. Clicking this button will show you our amazing ListView.package com.example.listViewpopupwithadapter; import android.os.Bundle; import android.view.View; import android.widget.AdapterView; import android.widget.AdapterView.OnItemClickListener; import android.app.AlertDialog; MainActivity public class MainActivity extends Activity { AlertDialog alertDialogStores; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); // button, to pop up from view.OnItemClickListener handler = new View.OnItemClickListener(){ public void onClick(View v) { switch (v.getId()) { case R.id.buttonShowPopUp: showPopUp(); break; } }; findViewById(R.id.buttonShowPopUp).setOnClickListener(handler); } public void showPopUp(){ // add your items, this can be done programically // your items can be from a database ObjectItem[] ObjectItemData = new ObjectItem[20]; ObjectItemData[0] = new ObjectItem(91, Mercury); ObjectItemData[1] = new ObjectItem(92, Watson); = new ObjectItem(93, Nissan); ObjectItemData[3] = new ObjectItem(94, Puregold); ObjectItemData[4] = new new SM); ObjectItemData[5] = new ObjectItem(96, 7 Eleven); ObjectItemData[6] = new ObjectItem(97, Ministop); ObjectItemData[7] = new ObjectItem(98, Fat Chicken); ObjectItemData[8] = new ObjectItem(99, Master Siomai); ObjectItemData[9] = new ObjectItem(100, Mang Inasal); ObjectItemData[10] = new ObjectItem(101, Mercury 2); ObjectItemData[11] = new ObjectItem(102, Watson 2); ObjectItemData[12] = new ObjectItem(103, Nissan 2); ObjectItemData[13] = new ObjectItem(104, Puregold 2); ObjectItemData[14] = new ObjectItem(105, SM 2); ObjectItemData[15] = new ObjectItem(106, 7 Eleven 2); ObjectItemData[16] = new ObjectItem(107, Ministop 2); ObjectItemData[17] = new ObjectItem(108, Fat Chicken 2); ObjectItemData[18] = new ObjectItem(109, Master Siomai 2); ObjectItemData[19] = new ObjectItem(110, Mang Inasal 2); our Instance Card ArrayAdapterItem adapter = new ArrayAdapterItem (R.layout.list_view_row_item, ObjectItemData); create a new ListView, set a tab, and click the item listViewviewItems = new ListView (this); listViewItems.setAdapter(adapter); listViewItems.setOnItemClickListener(new OnItemClickListener(){ public void onItemClick(AdapterView<?> parent, View view, int position, long id) { Context = view.getContext(); TextView viewItem = ((TextView) view.findViewById(R.id.textViewItem)); download TextView, and then set the text (element name) and tag (element ID) value to TextView TextViewItem = (TextView) convertView.findViewById(R.id.textViewItem); textViewItem.setText(objectItem.itemName); textViewItem.setTag(objectItem.itemId); return convertView; } 5. OnItemClickListenerListViewItem.java - Responds to what happens if a user clicks an item in the list?package com.example.listViewpopupwithadapter; import android.content.Context; import android.view.View; import android.widget.AdapterView; import android.widget.AdapterView.OnItemClickListener; import android.widget.TextView; import android.widget.Toast; * * Here you can control what to do next when the user selects the item */ public class OnItemClickListenerListViewItem implements OnItemClickListener { @Override public void onItemClick(AdapterView<?> parent, View view, int position, long id) { Context = view.getContext(); TextView viewItem = ((TextView) view.findViewById(R.id.textViewItem)); get the string list name you clickEditItemText = textViewItem.getText().toString(); get the item ID list you clickItemID = textViewItem.getTag().toString(); simply toast to Toast.makeText(context, Element: + listItemText + , Item ID: + listItemID, Toast.LENGTH_SHORT).show(); ((MainActivity) context).alertDialogStores.cancel(); } 6. list_view_row_item.xml – the layout of each item in the list. You can modify this if you want to have an image or subtitles for each listView. <RelativeLayout xmlns:android= android:layout_width=fill_parent android:layout_height=fill_parent android:padding=10dp> <TextView android:id=@+id/textViewItem android:layout_width=match_parent android:layout_height=wrap_content android:layout_alignparenttop=true android:text=Item name here... android:textsize=15dp></TextView> </RelativeLayout>

