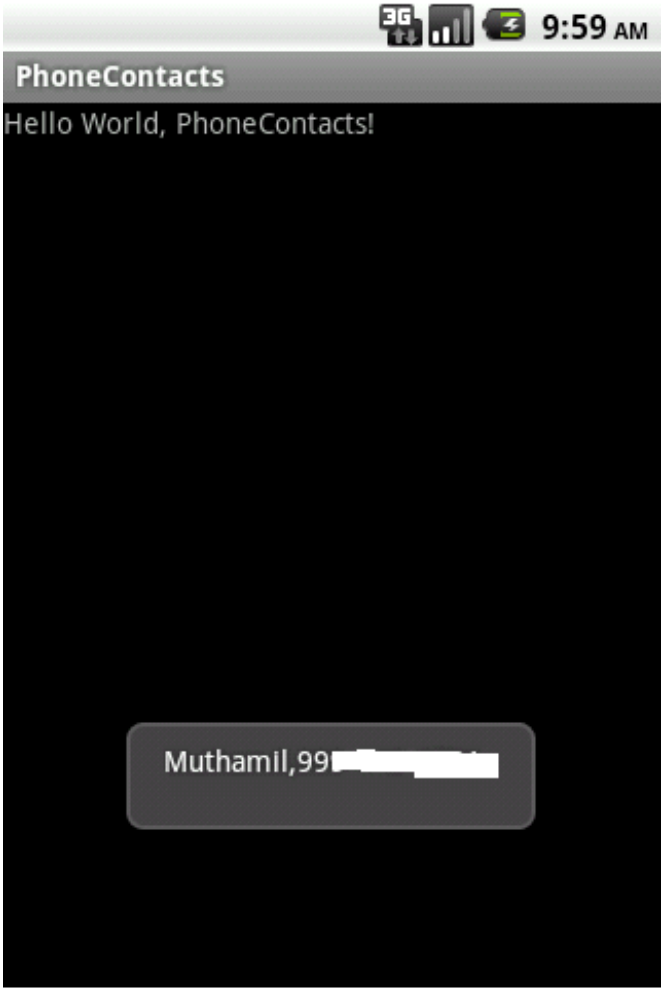
Phone Contacts



Set Permission

AndroidManifest.xml

<uses-permission android:name="*android.permission.READ_CONTACTS*" />
<uses-permission android:name="*android.permission.WRITE_CONTACTS*" />

```
package com.phonecontacts;

import android.app.Activity;
import android.database.Cursor;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.widget.Toast;

public class PhoneContacts extends Activity {
   /** Called when the activity is first created. */
   int count=0;
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
      try{
               Cursor cursor = getContentResolver().query(ContactsContract.
                                 CommonDataKinds.Phone.CONTENT_URI, null, null,null, null);
               while (cursor.moveToNext())
               {
               String name  =cursor.getString(cursor.getColumnIndex
                              (ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME));
               String phoneNumber = cursor.getString(cursor.getColumnIndex
                              (ContactsContract.CommonDataKinds.Phone.NUMBER));
               String data =name+","+phoneNumber+"\n";
               Toast.makeText(getBaseContext(), data, Toast.LENGTH_LONG).show();
               count=count +1;
               }
         }catch(Exception e)
          {
          Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH_LONG).show();
         }
        Toast.makeText(getBaseContext(), "Total Contacts ="+ count,
                                                      Toast.LENGTH_LONG).show();
    }
}
```

```
Cursor cursor = getContentResolver().query(ContactsContract.
                        CommonDataKinds.Phone.CONTENT_URI, null, null,null, null);
```

public ContentResolver getContentResolver ()

Return a ContentResolver instance for your application's package.

public final Cursor query (Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)

Query the given URI, returning a Cursor over the result set.

For best performance, the caller should follow these guidelines:

- Provide an explicit projection, to prevent reading data from storage that aren't going to be used.
- Use question mark parameter markers such as 'phone=?' instead of explicit values in the selection parameter, so that queries that differ only by those values will be recognized as the same for caching purposes.

Parameters

uri	The URI, using the content:// scheme, for the content to retrieve.
projection	A list of which columns to return. Passing null will return all columns, which is inefficient.
selection	A filter declaring which rows to return, formatted as an SQL WHERE clause (excluding the WHERE itself). Passing null will return all rows for the given URI.
selectionArgs	You may include ?s in selection, which will be replaced by the values from selectionArgs, in the order that they appear in the selection. The values will be bound as Strings.
sortOrder	How to order the rows, formatted as an SQL ORDER BY clause (excluding the ORDER BY itself). Passing null will use the default sort order, which may be unordered.

Returns

- A Cursor object, which is positioned before the first entry, or null