# Assignment 2

***Here are the code and snapshots for creating a Bound Service using Messenger:***

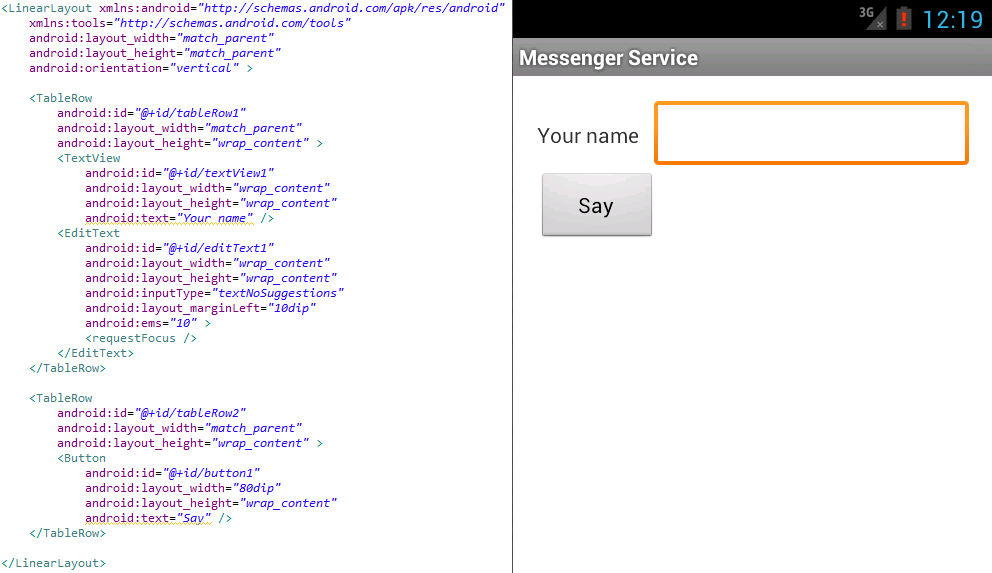
Follow the steps described in this Android Tutorial:

**Step 1:** Create a **new Android Project with the class name MessengerServiceActivity**.

**Step 2:** Add a textview, an edittext and button in layout and name textview as **“Your name”** andbutton **as “Say”**

**Step 3:** **Set the onClick Listeners of button** inside the MainActivity.

**Step 4:** **Create a new class** “**MessengerService**” and extends it to Service Class



Tag to add in Manifest:

<service android:name=*“.MessengerService”*  android:enabled=*“true”* > </service>

**Code : MainActivity.java**

**public** **class** MessengerServiceActivity **extends** Activity {

**private** EditText \_edtName;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_messenger\_service*);

\_edtName = (EditText) findViewById(R.id.*editText1*);

Button btnSay = (Button) findViewById(R.id.*button1*);

btnSay.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

sayHello(\_edtName.getText().toString());

}

});

}

/\*\* Messenger for communicating with the service. \*/

Messenger mService = **null**;

**boolean** mBound;

/\*\* Class for interacting with the main interface of the service. \*/

**private** ServiceConnection mConnection = **new** ServiceConnection() {

**public** **void** onServiceConnected(ComponentName className, IBinder service) {

// This is called when the connection with the service has been

// established, giving us the object we can use to

// interact with the service. We are communicating with the

// service using a Messenger, so here we get a client-side

// representation of that from the raw IBinder object.

mService = **new** Messenger(service);

mBound = **true**;

}

**public** **void** onServiceDisconnected(ComponentName className) {

mService = **null**;

mBound = **false**;

}

};

**public** **void** sayHello(String name) {

**if** (!mBound) **return**;

// Create and send a message to the service, using a supported 'what' value

Message msg = Message.*obtain*(**null**,

MessengerService.*MSG\_SAY\_HELLO*, 0, 0);

Bundle b = **new** Bundle ();

b.putString ("Input name", name);

msg.setData(b);

**try** {

mService.send(msg);

}

**catch** (RemoteException e) { }

}

@Override

**protected** **void** onStart() {

**super**.onStart();

bindService(**new** Intent(**this**, MessengerService.**class**), mConnection,

Context.*BIND\_AUTO\_CREATE*);

Log.*i*("Starting", "MessengerService");

}

@Override

**protected** **void** onStop() {

**super**.onStop();

**if** (mBound) {

unbindService(mConnection);

mBound = **false**;

}

}

}

**Code: MyService.java**

**public** **class** MessengerService **extends** Service {

// Command to service to display msg

**static** **final** **int** *MSG\_SAY\_HELLO* = 1;

// Handler of incoming client msgs

**class** IncomeHandler **extends** Handler {

**public** **void** handleMessage(Message msg) {

**switch** (msg.what) {

**case** *MSG\_SAY\_HELLO*:

Log.*i*("message", "hello! " + msg.getData()

.getString("Input name"));

**break**;

**default**:

**super**.handleMessage(msg);

}

}

}

// Target we publish for clients to send messages to IncomeHandler

**final** Messenger mMessenger = **new** Messenger(**new** IncomeHandler());

// When binding to the service, we return an interface to our messenger

// for sending messages to the service

**public** IBinder onBind(Intent intent) {

**return** mMessenger.getBinder();

}

}