

Android Tutorial for Beginners 36 # Bound Services



ProgrammingKnowledge ✓

1,65 M d'abonnés

Rejoindre

S'abonner

<https://www.youtube.com/watch?v=cJsQMisTaa8>

[Bound service extending the Binder class](#)

Bound Services

A bound service is the server in a client-server interface. A bound service allows components (such as activities) to bind to the service, send requests, receive responses, and even perform interprocess communication (IPC).

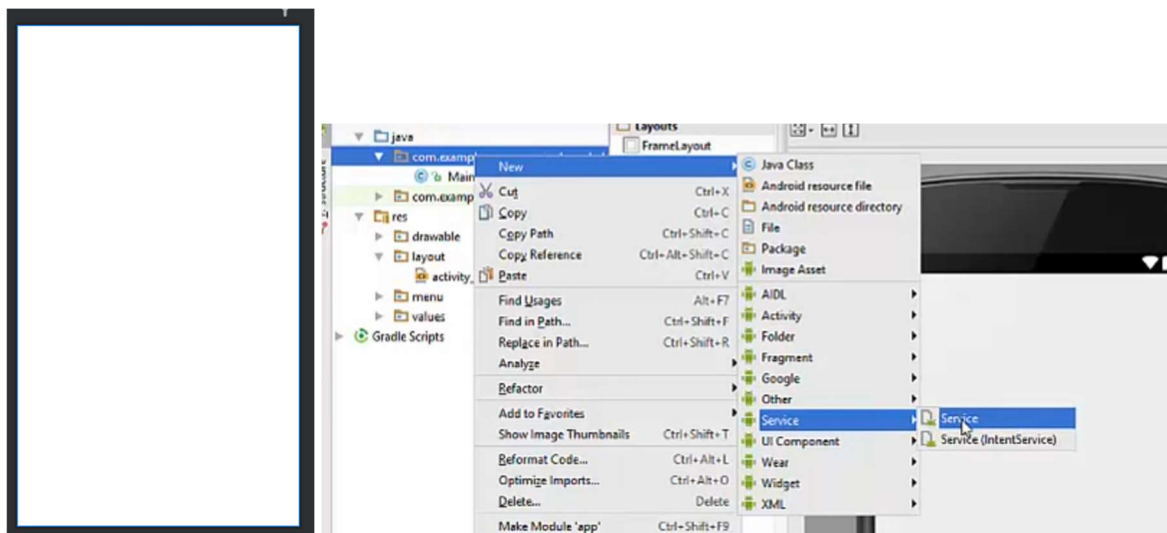
Creating a Bound Service

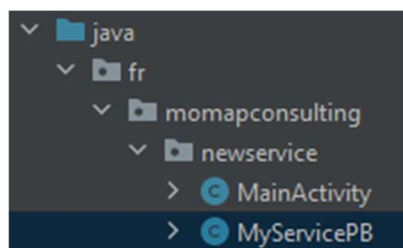
Extending the Binder class :- If your service is private to your own application and runs in the same process as the client (which is common), you should create your interface by extending the Binder class and returning an instance of it from onBind()

Using a Messenger

Using AIDL :- AIDL (Android Interface Definition Language)

I New project et créer un service dans le package





```
1 usage
public class MyServicePB extends Service {
    public MyServicePB() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }
}
```

```
public class MyServicePB extends Service {
    public class LocalBinder extends Binder {
    }

    public MyServicePB() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }
}
```

```
public class MyServicePB extends Service {
    public class LocalBinder extends Binder {
        MyServicePB getService(){
            return MyServicePB.this;
        }
    }

    public MyServicePB() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented")
    }
}
```

Ajouter private final IBinder ibBinder ... et return IBinder

```

3 usages
public class MyServicePB extends Service {
    1 usage
    private final IBinder iBinder= new LocalBinder();

    1 usage
    public class LocalBinder extends Binder {
        MyServicePB getService(){
            return MyServicePB.this;
        }
    }

    public MyServicePB() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        return iBinder;
    }
}

```

```

public class MyServicePB extends Service {
    1 usage
    private final IBinder iBinder= new LocalBinder();
    private final Random mGenerator = new Random();
}

```

```

3 usages
public class MyServicePB extends Service {
    1 usage
    private final IBinder iBinder= new LocalBinder();
    1 usage
    private final Random mGenerator = new Random();

    1 usage
    public class LocalBinder extends Binder {
        MyServicePB getService(){
            return MyServicePB.this;
        }
    }

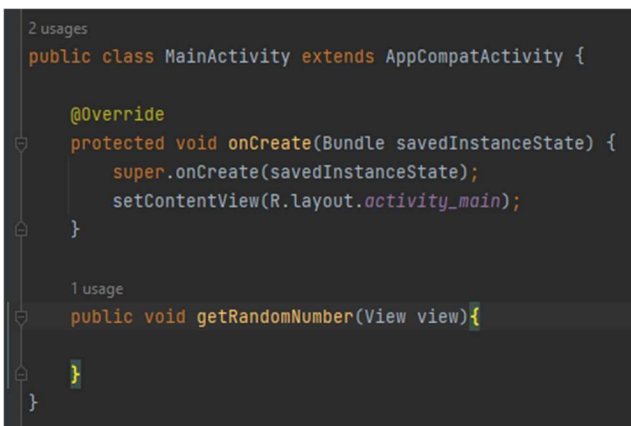
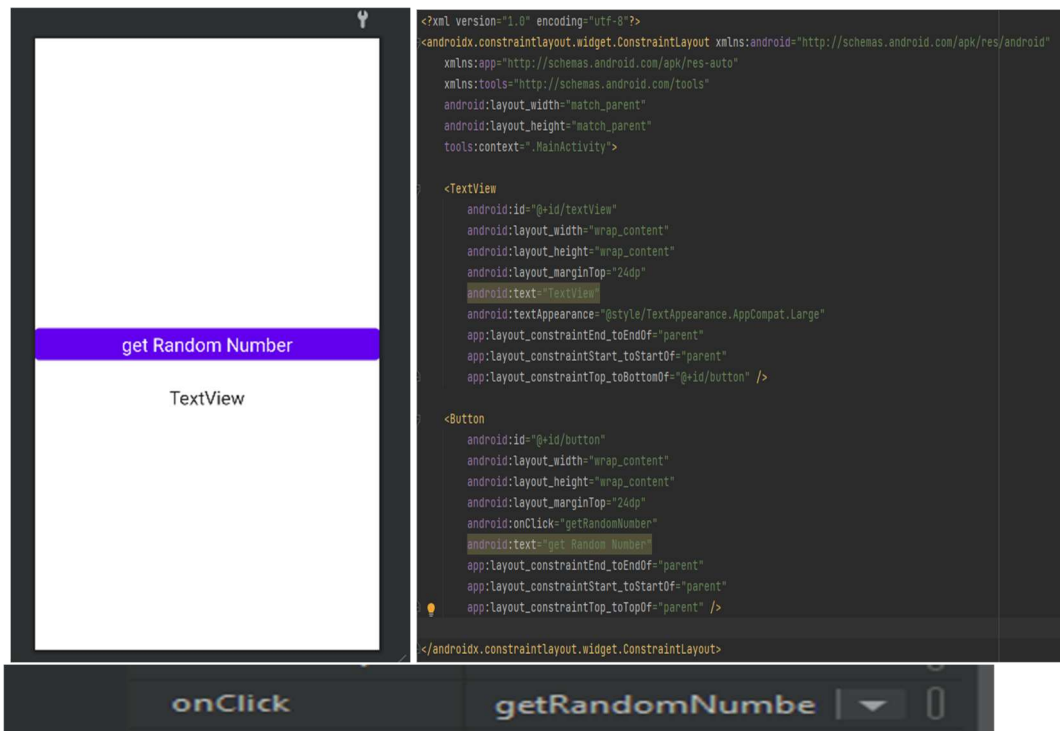
    public MyServicePB() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        return iBinder;
    }

    public int getRandom(){
        return mGenerator.nextInt( bound: 200); // random gener  de 0 a 200
    }
}

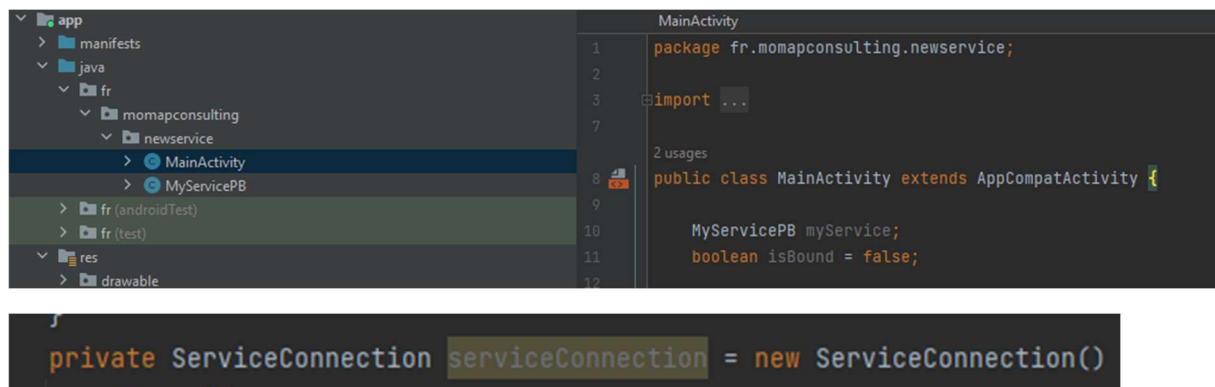
```

Activity-main.xml ajouter bouton (large), onClick getRandomNumber et un TextView



Dans MainActivity, créer méthode onRandomNumber

Créer 2 variables



```

    public void getRandomNumber(View view){

    }

    private ServiceConnection serviceConnection = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName name, IBinder service) {

        }

        @Override
        public void onServiceDisconnected(ComponentName name) {

        }
    };
}

```

```

    private ServiceConnection serviceConnection = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName name, IBinder service) {
            MyServicePB.LocalBinder binder=(MyServicePB.LocalBinder) service;
            myService=binder.getService();
            isBound=true;
        }

        @Override
        public void onServiceDisconnected(ComponentName name) {
            isBound=false;
        }
    };
}

```

Dans onCreate créer un Intent et un bindService

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Intent intent=new Intent( packageContext, this,MyServicePB.class);
    bindService(intent,serviceConnection, Context.BIND_AUTO_CREATE);
}

```

```

1 usage
public void getRandomNumber(View view) {
    TextView textView = (TextView) findViewById(R.id.textView);
    textView.setText(Integer.toString(myService.getRandom()));
}

```

Run

