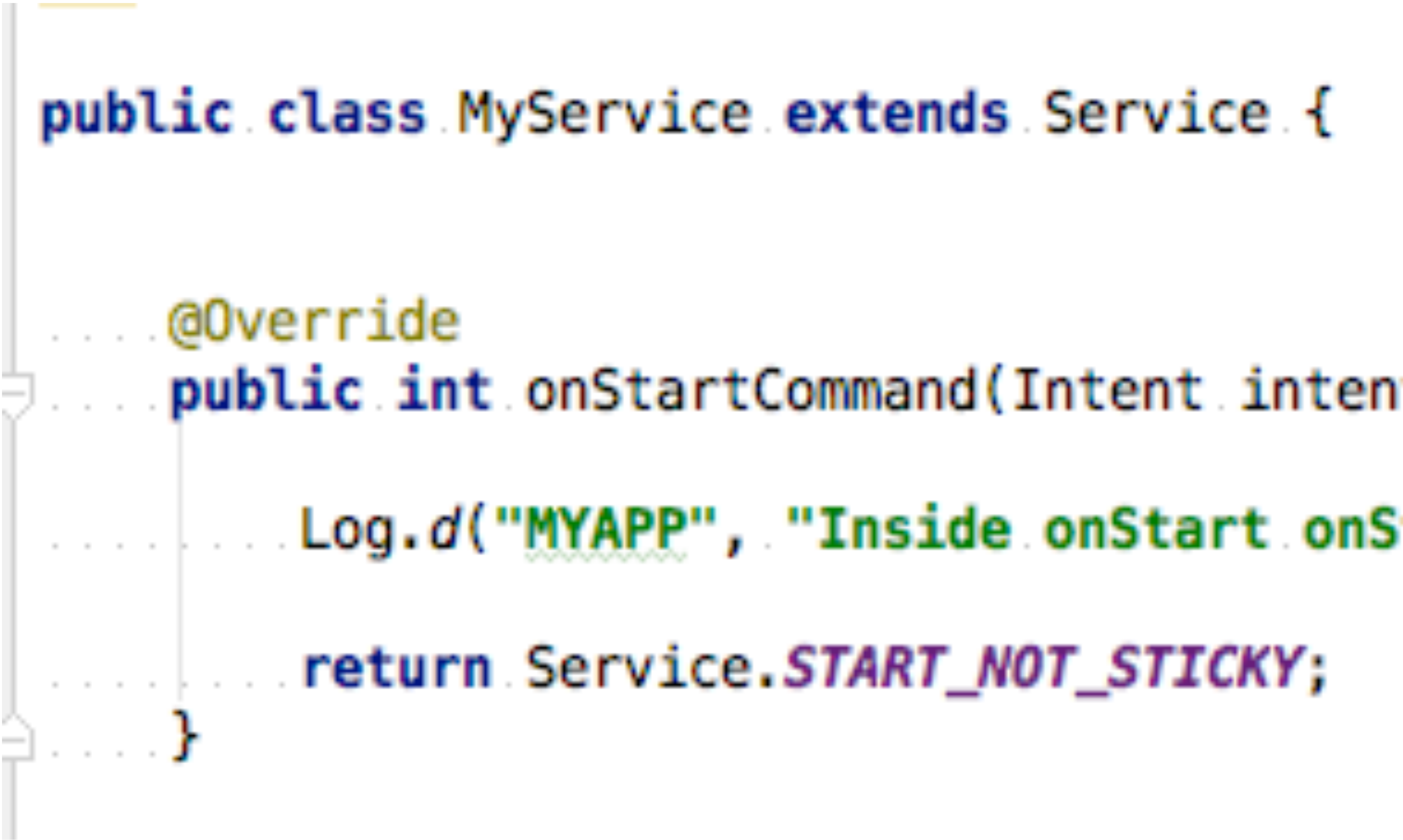
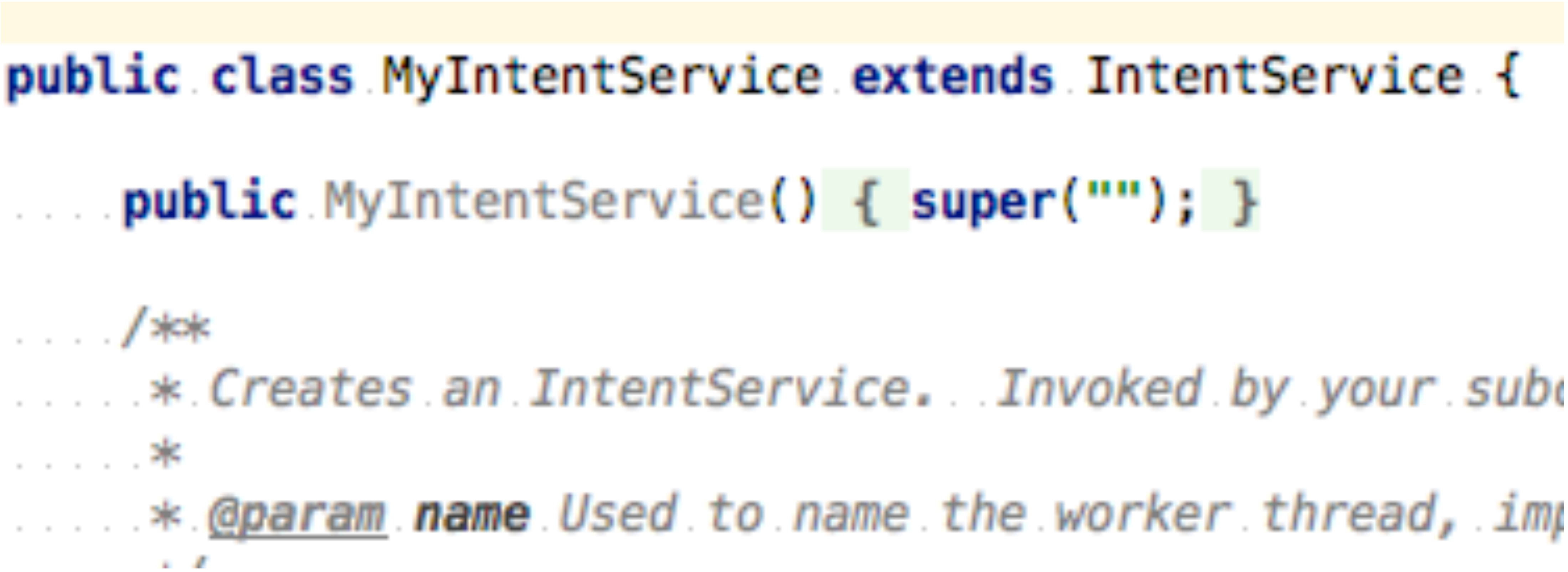
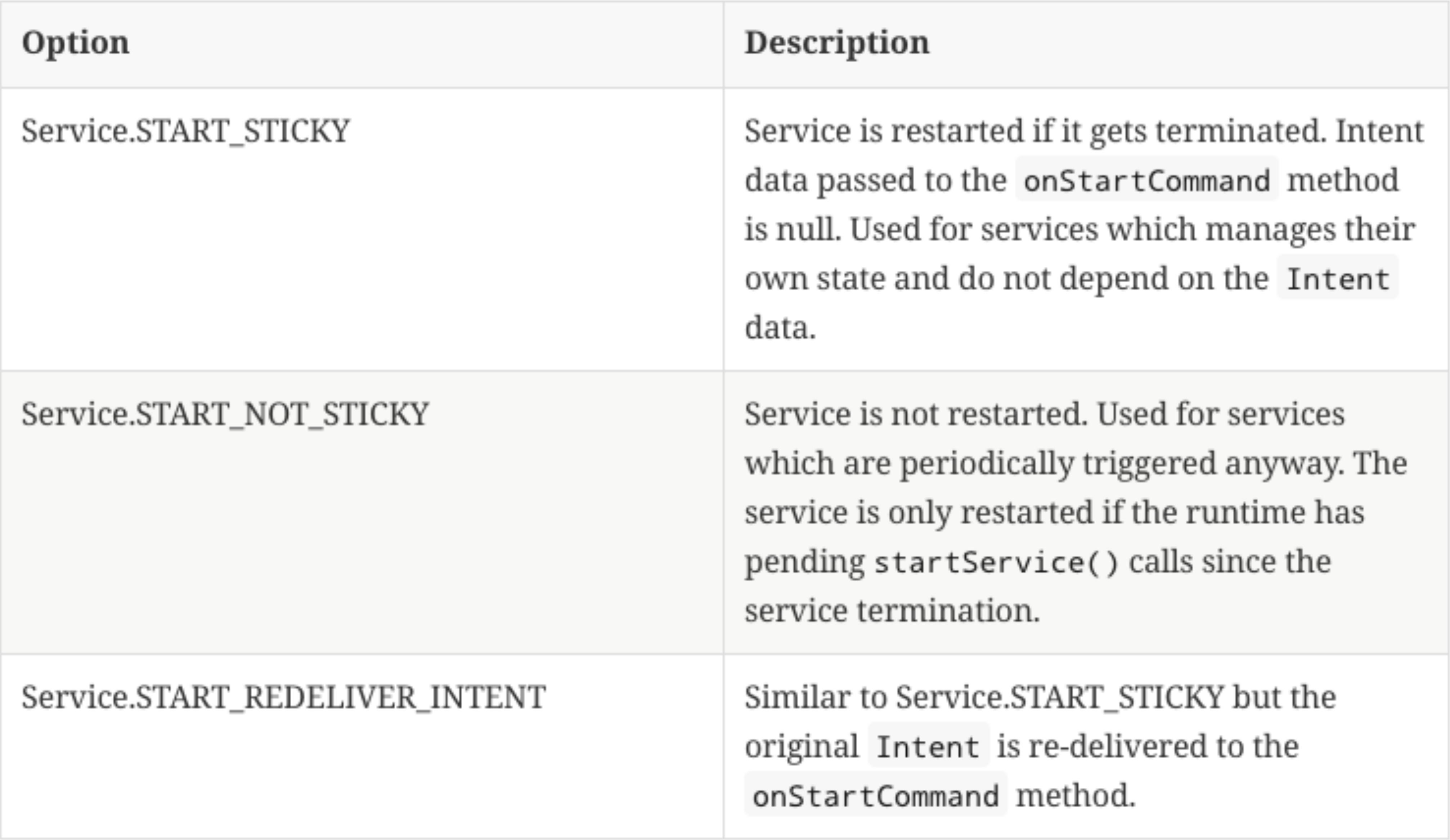
Service



```java
public class MyService extends Service {


    @Override
    public int onStartCommand(Intent inten

        Log.d("MYAPP", "Inside onStart onS

        return Service.START_NOT_STICKY;
    }
```

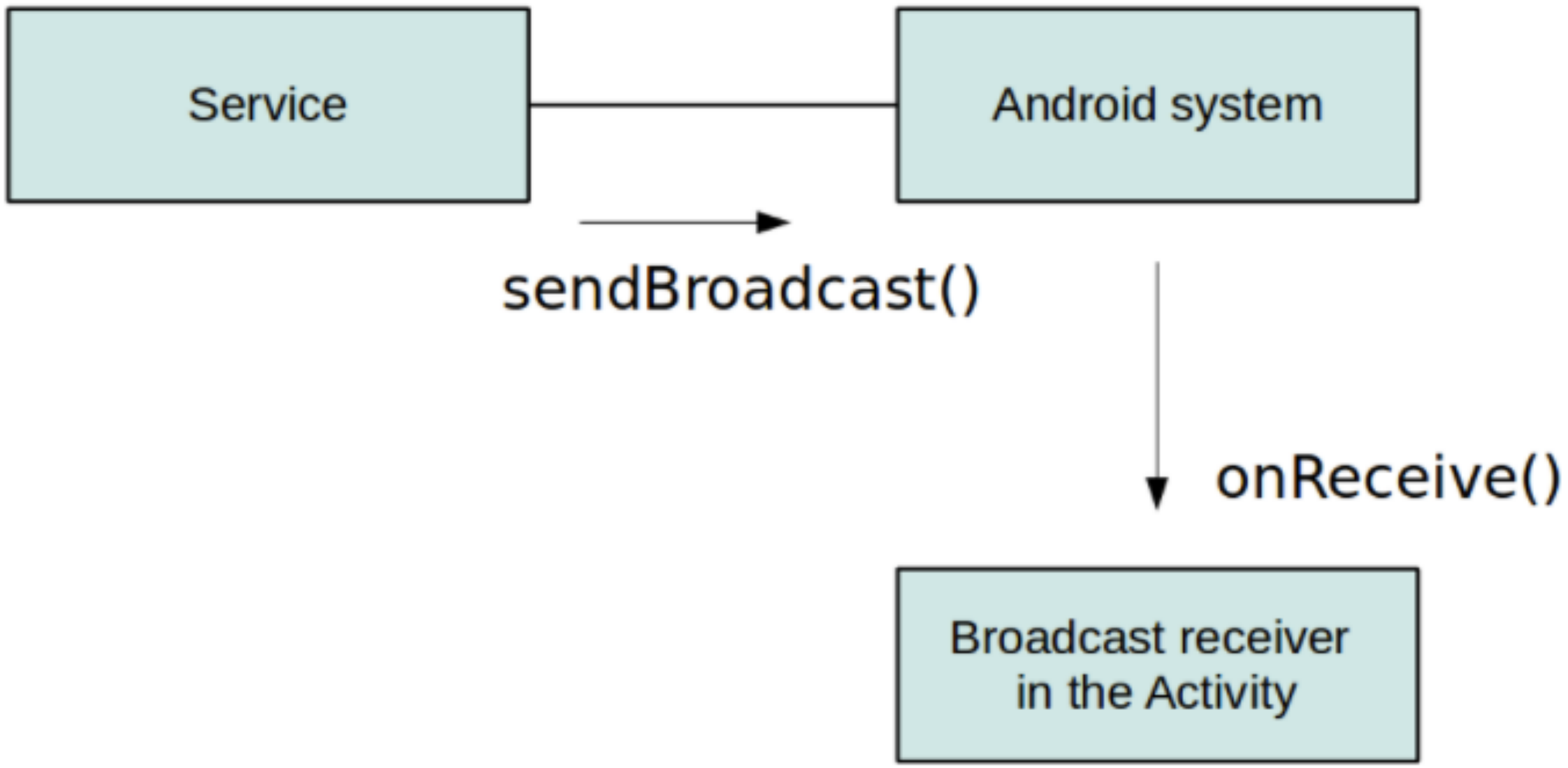


```
public class MyIntentService extends IntentService {

    public MyIntentService() { super(""); }

    /**
     * Creates an IntentService.  Invoked by your subc
     *
     * @param name Used to name the worker thread, imp
     */
```

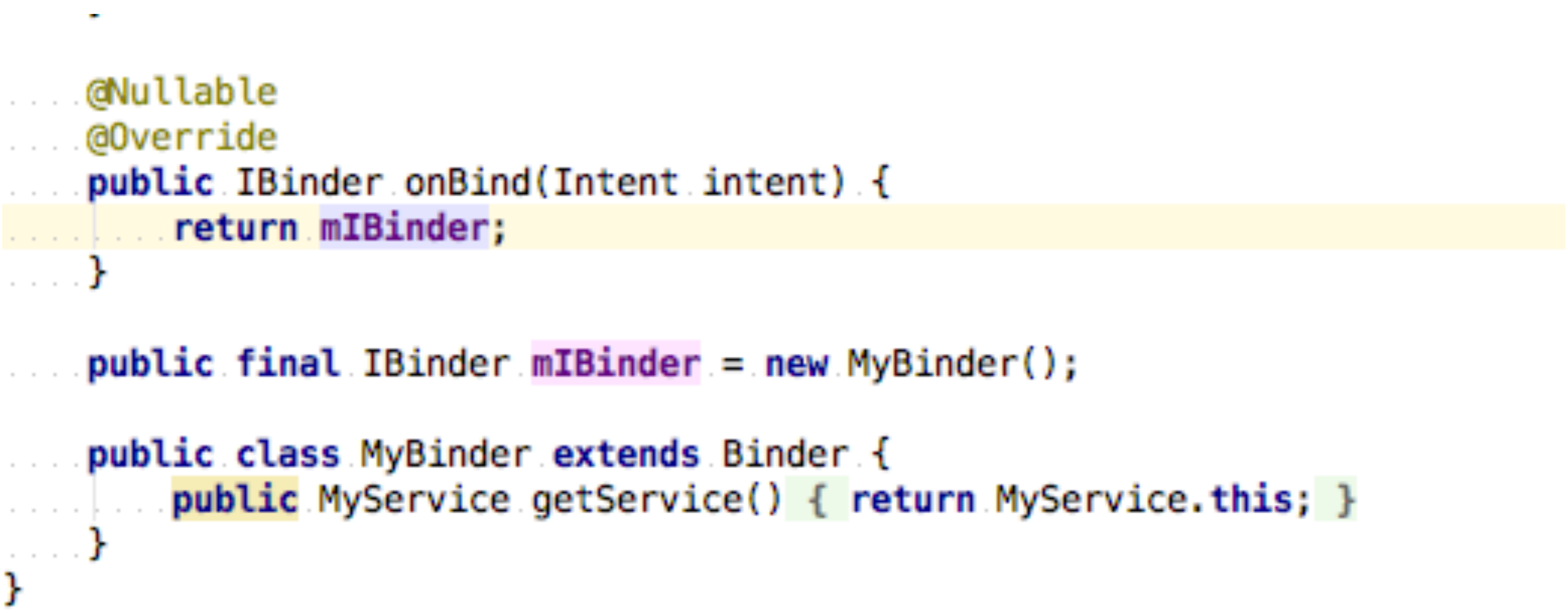


Option	Description
Service.START_STICKY	Service is restarted if it gets terminated. Intent data passed to the `onStartCommand` method is null. Used for services which manages their own state and do not depend on the `Intent` data.
Service.START_NOT_STICKY	Service is not restarted. Used for services which are periodically triggered anyway. The service is only restarted if the runtime has pending `startService()` calls since the service termination.
Service.START_REDELIVER_INTENT	Similar to Service.START_STICKY but the original `Intent` is re-delivered to the `onStartCommand` method.

Service Communication



Bound Services



```java
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return mIBinder;
    }

    public final IBinder mIBinder = new MyBinder();

    public class MyBinder extends Binder {
        public MyService getService() { return MyService.this; }
    }
}
```