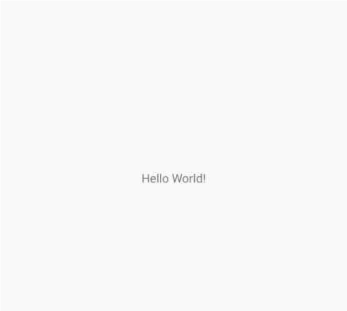
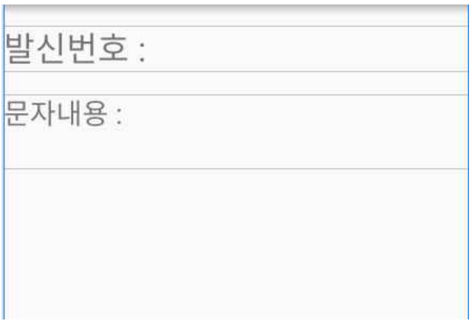
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

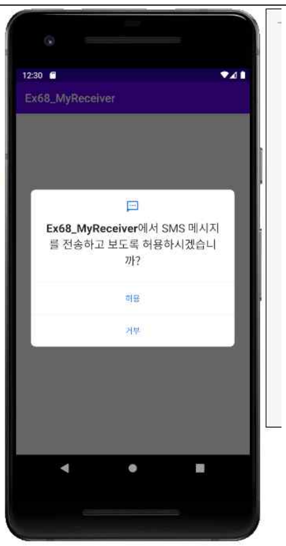

Hello World!

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity2">

<TextView
android:id="@+id/textView1"
android:layout_marginTop="20dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textSize="30sp"
android:text="발신번호 : " />

<TextView
android:id="@+id/textView2"
android:layout_marginTop="20dp"
android:textSize="25sp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="문자내용 : \n" />
</LinearLayout>
```


발신번호 :

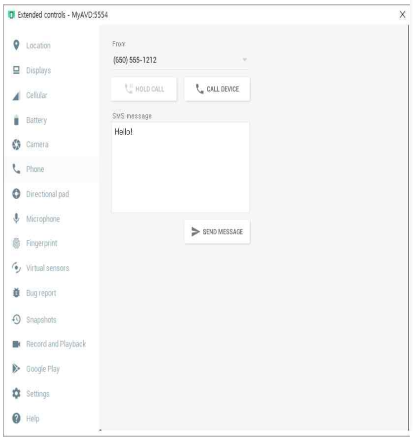
문자내용 :

```java
// 앱 허가 받기 :  Gradle Scripts - build.gradle(Module.app)
// implementation 'com.github.pedroSG94:AutoPermissions:1.0.3'
// allprojects {
//    repositories {
//        maven{url 'https://jitpack.io'}
//    }
// }
public class MainActivity extends AppCompatActivity implements
AutoPermissionsListener {

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

// 모든 위험권한을 자동으로 부여하는 메소드
AutoPermissions.Companion.loadAllPermissions(this, 100);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults);
AutoPermissions.Companion.parsePermissions(this, requestCode,
permissions,this);
}

@Override
public void onDenied(int i, String[] strings) { }
@Override
public void onGranted(int i, String[] strings) { }
}
```



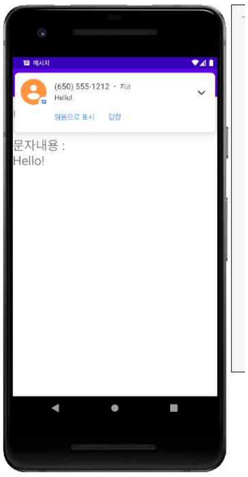
Ex68_MyReceiver에서 SMS 메시지를 전송하고 보도록 허용하시겠습니까?

```java
public class MainActivity2 extends AppCompatActivity {
TextView textView1, textView2;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main2);
textView1 = findViewById(R.id.textView1);
textView2 = findViewById(R.id.textView2);

// 리시버에서 호출한 액티비티가 메모리에 존재하지 않았던 액티비티이므로
// onCreate()에서 실행
Intent intent = getIntent();
processIntent(intent);
}

@Override
protected void onNewIntent(Intent intent) {
super.onNewIntent(intent);
processIntent(intent);
}

private void processIntent(Intent intent){
String sender = intent.getStringExtra("sender");
String contents = intent.getStringExtra("contents");
textView1.append(sender);
textView2.append(contents);
}
}
```



From: (650) 555-1212

SMS message: Hello!

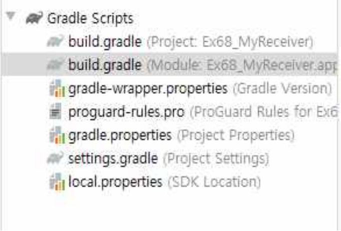


(650) 555-1212 · 지금
Hello!

문자내용 :
Hello!

```java
// Receiver는 BroadcastReceive를 상속 받는다.
// 메니페스트에 등록한다.
public class MyReceiver extends BroadcastReceiver {

// 문자가 들어오면 실행된다.
@Override
public void onReceive(Context context, Intent intent) {
// intent안에 Bundle 객체 존재함 ( 메세지 전달하는 역할을 함 )
Bundle bundle = intent.getExtras();
// 정상적인 프로토콜을 이용해서 문자을 받았을 때
SmsMessage[] msg  =  paresSms(bundle);
if(msg != null && msg.length>0){
String sender = msg[0].getOriginatingAddress();  // 보낸 주소
String contents = msg[0].getMessageBody();

// 화면에 메세지 보내기 (리시버도 UI가 없다. => MainActivity2.java)
sendToActivity(context, sender, contents);
}
}

// 정상적인 프로토콜을 이용해서 문자을 받았을 때
private SmsMessage[] paresSms(Bundle bundle){
Object[] objs = (Object[])bundle.get("pdus");
SmsMessage[] msg = new SmsMessage[objs.length];
// 해당 SDK 버전과 지원하는 버전이 다를때  구분하여 작성
for(int i=0; i < msg.length ; i ++){
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
String format = bundle.getString("format");
msg[i] = SmsMessage.createFromPdu((byte[])objs[i], format);
}else{
msg[i] = SmsMessage.createFromPdu((byte[])objs[i]);
}
}
return msg;
}

private void sendToActivity(Context context, String sender, String
contents){
Intent intent = new Intent(context, MainActivity2.class);
// 서비스에서 화면 호출과 같은 방법
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK¦
Intent.FLAG_ACTIVITY_SINGLE_TOP¦
Intent.FLAG_ACTIVITY_CLEAR_TOP);
intent.putExtra("sender", sender);
intent.putExtra("contents",contents);

context.startActivity(intent);
}
}
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.ict.ex68_myreceiver">

<uses-permission android:name="android.permission.RECEIVE_SMS" />

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity2"></activity>
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name=".MyReceiver"
android:enabled="true"
android:exported="true">
<!-- 목적에 맞는 상황만 받기 -->
<intent-filter>
<action android:name="android.provider.Telephony.SMS_RECEIVED" />
</intent-filter>
</receiver>
</application>
</manifest>
```



Gradle Scripts
build.gradle (Project: Ex68_MyReceiver)
build.gradle (Module: Ex68_MyReceiver.app
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for Ex6
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

```gradle
allprojects {
   repositories {
   maven{url 'https://jitpack.io'}
  }
}

dependencies {
implementation fileTree(dir: "libs", include: ["*.jar"])
implementation 'androidx.appcompat:appcompat:1.2.0'
implementation 'androidx.constraintlayout:constraintlayout:2.0.1'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'androidx.test.ext:junit:1.1.2'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
implementation 'com.github.pedroSG94:AutoPermissions:1.0.3'
}
```