



```
<receiver
    android:name=".WidgetProvider"
    android:exported="true">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="android.action.MY_ACTION"/>
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widgetprovider"
    />
</receiver>
```

AndroidManifest

```
<receiver
    android:name=".WidgetProvider"
    android:exported="true">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="android.action.MY_ACTION" />
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widgetprovider"
    />
</receiver>

<activity
    android:name=".MainActivity"
    android:exported="true"
    android:launchMode="singleInstance">
    <intent-filter>

</manifest>
```

Receiver

```
<receiver
  android:name=".WidgetProvider"
  android:exported="true">
  <intent-filter>
    <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    <action android:name="android.action.MY_ACTION"/>
  </intent-filter>
  <meta-data
    android:name="android.appwidget.provider"
    android:resource="@xml/widgetprovider"
  />
</receiver>
```

Receiver

- 특정 이벤트가 발생했을 때 다수의 Application 에게 전달하는 서비스
- Activity 인테트 필터와 비슷하지만 인테트 필터는 사용자에게 선택을 유도하지만, 이것은 등록이 되어 있다면 모두에게 보낸다

Intent-filter

```
<receiver
    android:name=".WidgetProvider"
    android:exported="true">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="android.action.MY_ACTION"/>
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widgetprovider"
    />
</receiver>
```

Intent-filter

- 해당 컴포넌트가 동작될 유형을 정의

```
<action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
```

```
<action android:name="android.action.MY_ACTION"/>
```


Meta-data

```
<receiver
    android:name=".WidgetProvider"
    android:exported="true">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="android.action.MY_ACTION"/>
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widgetprovider"
    />
</receiver>
```


WidgetProvider

```
<receiver
    android:name=".WidgetProvider"
    android:exported="true">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="android.action.MY_ACTION" />
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widgetprovider"
    />
</receiver>
```

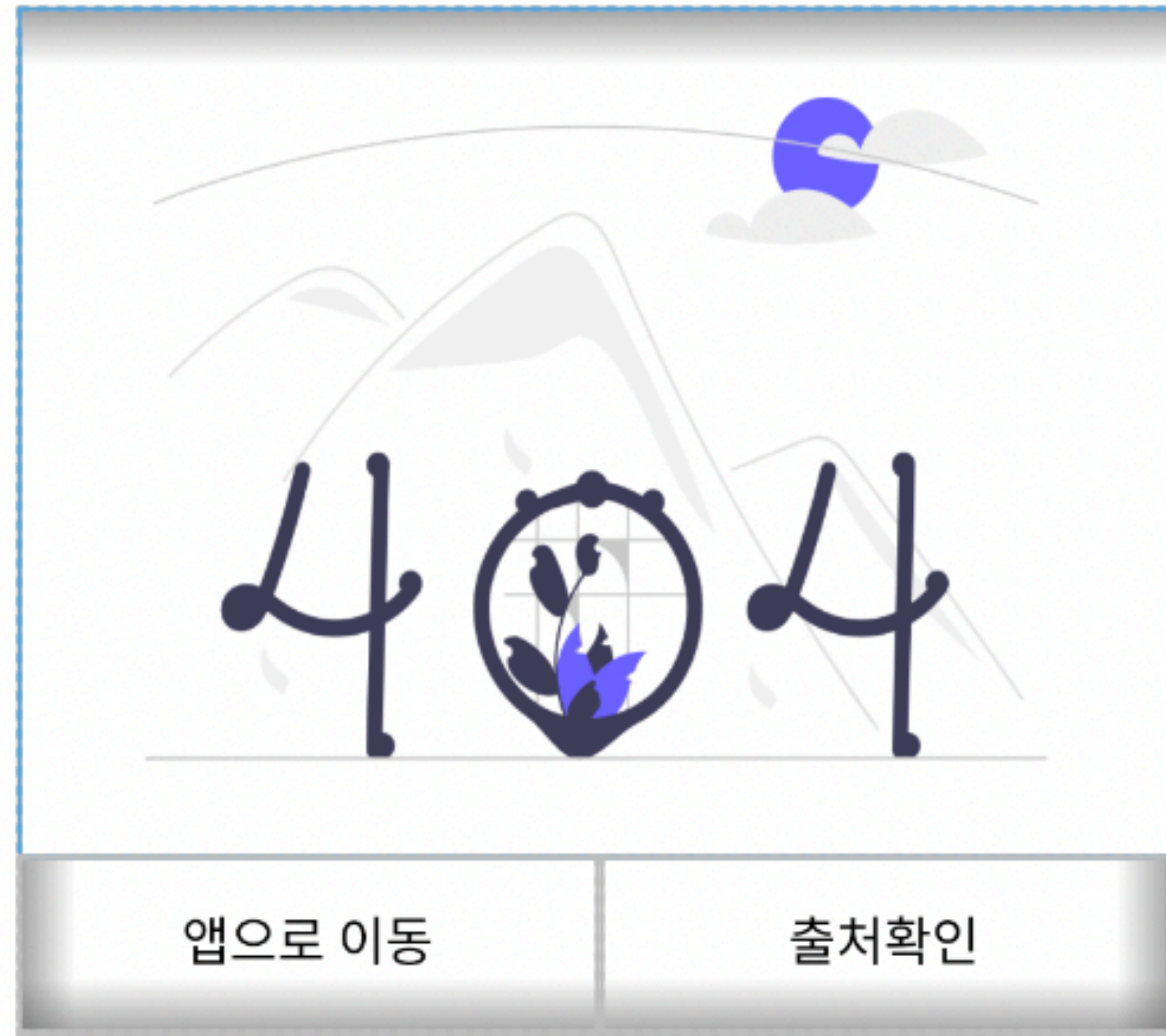
WidgetProvider

```
class WidgetProvider : AppWidgetProvider() {  
    private val MY_ACTION = "android.action.MY_ACTION"  
  
    private fun setMyAction(context: Context?): PendingIntent {  
        val intent = Intent(context, MainActivity::class.java)  
        return PendingIntent.getActivity(context, requestCode: 0, intent, PendingIntent.FLAG_IMMUTABLE)  
    }  
  
    private fun buildURIIntent(context: Context?): PendingIntent {  
        val intent = Intent(Intent.ACTION_VIEW).setData(Uri.parse(uriString: "https://stickcode.tistory.com/150"))  
        return PendingIntent.getActivity(context, requestCode: 0, intent, PendingIntent.FLAG_IMMUTABLE)  
    }  
  
    private fun addViews(context: Context?): RemoteViews {  
        val views = RemoteViews(context?.packageName, R.layout.widget)  
        views.setOnClickPendingIntent(R.id.btn, setMyAction(context))  
        views.setOnClickPendingIntent(R.id.btn2, buildURIIntent(context))  
        return views  
    }  
  
    override fun onUpdate(context: Context?, appWidgetManager: AppWidgetManager?, appWidgetIds: IntArray?) {  
        super.onUpdate(context, appWidgetManager, appWidgetIds)  
        appWidgetIds?.forEach { appWidgetId ->  
            val views: RemoteViews = addViews(context)  
            appWidgetManager?.updateAppWidget(appWidgetId, views)  
        }  
    }  
}
```

```
private fun setMyAction(context: Context?): PendingIntent {  
    val intent = Intent(context, MainActivity::class.java)  
    return PendingIntent.getActivity(context, requestCode: 0, intent, PendingIntent.FLAG_IMMUTABLE)  
}
```

```
private fun buildURIIntent(context: Context?): PendingIntent {  
    val intent = Intent(Intent.ACTION_VIEW).setData(Uri.parse(uriString: "https://stickode.tistory.com/150"))  
    return PendingIntent.getActivity(context, requestCode: 0, intent, PendingIntent.FLAG_IMMUTABLE)  
}
```

```
private fun addViews(context: Context?): RemoteViews {  
    val views = RemoteViews(context?.packageName, R.layout.widget)  
    views.setOnClickPendingIntent(R.id.btn, setMyAction(context))  
    views.setOnClickPendingIntent(R.id.btn2, buildURIIntent(context))  
    return views  
}
```

btn1

btn2

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<appwidget-provider
```

```
    android:initialKeyguardLayout="@layout/widget"
```

```
    android:minHeight="72dp"
```

```
    android:minWidth="145dp"
```

```
    android:updatePeriodMillis="86400000"
```

```
    android:resizeMode="horizontal|vertical"
```

```
xmlns:android="http://schemas.android.com/apk/res/android" />
```


timer (1):(23)

TIMER START

TIMER RESET

TIMER STOP

점