

# ProtoPie Android Broadcast Channel Guide

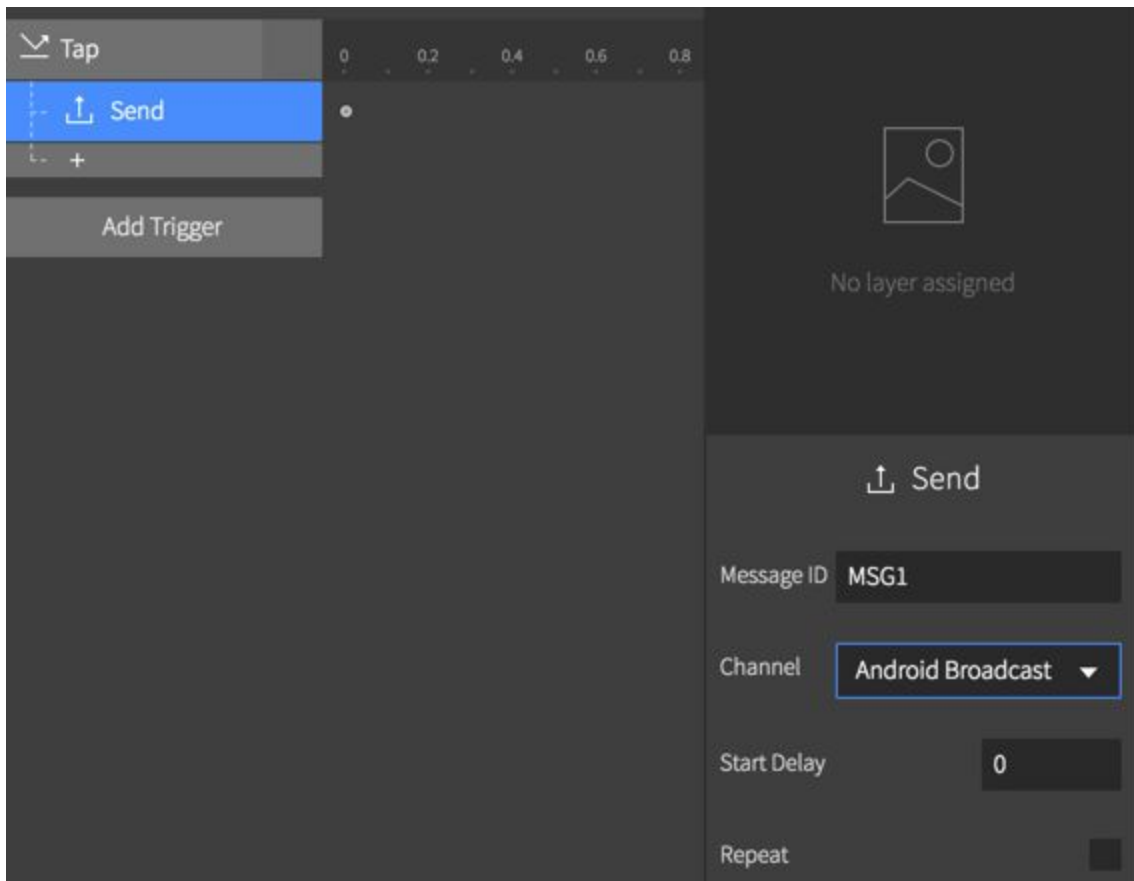
Send response 와 Receive trigger 에서 사용할 수 있는 Android Broadcast channel 은 Android broadcast 로 동일 기기내의 다른 app 과 message 를 주고 받도록 해 줍니다.

Receive trigger 는 ProtoPie 에서 직접 지원하지 않는 입력으로부터 ProtoPie 가 반응하도록 할 수 있습니다. 예를 들면, Android 기기에 연결된 입력 device 의 입력에 반응하거나, 서버로부터 받은 이벤트에 반응하도록 할 수도 있습니다.

반대로 Send response 는 ProtoPie trigger 에 대응하여 ProtoPie 에서 직접 지원하지 않는 출력을 만들어낼 수 있습니다. 예를 들어, Android 에서 API 로 제어가능한 조명장치를 prototype 에서 제어할 수도 있고, 서버 API 를 호출하는 것도 가능합니다.

## Send Response

ProtoPie Studio 에서 Send response 를 추가하고, Channel 로 "Android Broadcast" 를 선택한 후 Message ID 에 적절한 문자열을 정해서 입력합니다.



ProtoPie 는 Send response 가 실행될 때 동일 Android 기기내의 app 에게 Broadcast 를 보냅니다. Broadcast 의 action 값은 io.protopie.action.ONE\_TIME\_RESPONSE 이고, intent 의 extra parameter 로 messageId 가 전달됩니다.

이를 수신하기 위해서는 [BroadcastReceiver](#) 의 subclass 를 생성한 후 Manifest 에 등록해야 합니다.

아래 예제 코드는 ProtoPie 로부터 받은 messageId 값을 Toast 로 띄웁니다.

```
package io.protopie.protopiebridgesample;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class ProtoPieReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if ("io.protopie.action.ONE_TIME_RESPONSE".equals(intent.getAction())) {
            // Message received from ProtoPie.
            String messageId = intent.getStringExtra("messageId");
            String toastText = "Received " + messageId + " from ProtoPie";
            Toast.makeText(context, toastText, Toast.LENGTH_LONG).show();
        }
    }
}
```

이렇게 생성한 class 는 AndroidManifest.xml 에 등록해야 합니다. AndroidManifest.xml 파일을 열고, application 하위에 다음과 같이 추가합니다.

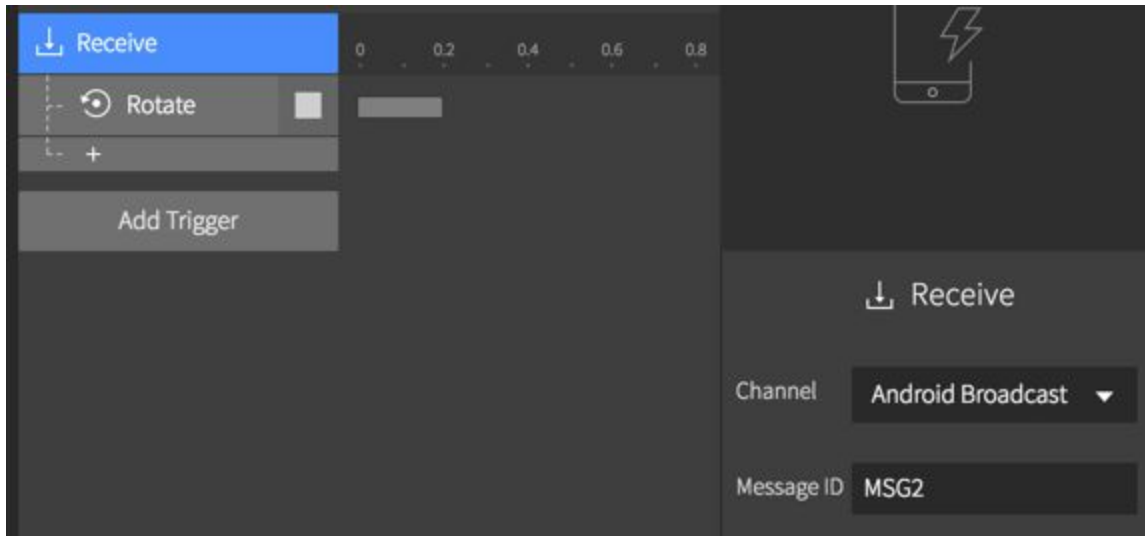
```
<receiver android:name=".ProtoPieReceiver">
    <intent-filter>
        <action android:name="io.protopie.action.ONE_TIME_RESPONSE"/>
    </intent-filter>
</receiver>
```

## Receive Trigger

ProtoPie 의 Receive trigger 는 동일 Android 기기 내의 다른 app 이 보내는 broadcast 를 수신할 수 있습니다.

ProtoPie 에서 Receive trigger 를 추가하고, Channel 을 Android Broadcast 로 변경한 후 Message ID 에 적절한 문자열을 정하여 입력해주시오.

이제 Receive trigger 하위에 적절한 response 를 붙입니다. 아래 예제는 message 를 수신할 때마다 사각형이 회전하도록 했습니다.



Android app 에서 ProtoPie 로 message 를 전송하기 위해서는, Android Broadcast 를 생성하기만 하면 됩니다. Broadcast 의 action 값으로는 io.protopie.action.ONE\_TIME\_TRIGGER 을 사용하고, message 를 구분하기 위한 messageId 는 extra data 로 전달합니다.

다음은 ProtoPie 로 message 를 전송하는 sample code 입니다.

```
private void sendMessageToProtoPie(Context context, String messageId) {
    Intent intent = new Intent("io.protopie.action.ONE_TIME_TRIGGER");
    intent.putExtra("messageId", messageId);
    context.sendBroadcast(intent);
}
```