

# Broadcast Recieverとは？

---

- BroadcastReceiver（ブロードキャストレシーバー）とは、ブロードキャストしたintentを受け取る仕組みのこと
- OSのAndroidシステム側でブロードキャストされるイベント（スクリーンのON/OFFなど）以外でも、自分で作成したアプリでも独自のintentを生成してブロードキャストできる

## intentとは？

---

- intentとは、アプリケーションの中の1つ1つの機能のこと。たとえばアプリケーション同士や、アプリケーションとウィジェット、アプリケーションとシステムを橋渡しする仕組みのこと
- BroadcastReceiverを使わずとも、intentだけでも利用が可能で、例えば電話をかけたりカメラを起動することもできる
- BroadcastReceiverでは、例えばアプリをインストールした際などに発生するintentを検知することができます。例えば...
  - スクリーンのON/OFFを検知する
  - 電池の状態を検知する
  - アプリのインストールを検知する
  - 振動を検知する 等

## 使ってみよう

---

アクティビティからサービスを起動 → サービスでレシーバーを登録 という流れでやってみました

```
public class MyReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction() != null) {
            if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
                Log.d("registerReceiver: ", "ON");
            }
            if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
                Log.d("registerReceiver: ", "OFF");
            }
        }
    }
}
```

```
public class MyActivity extends Activity {

    Button startButton;
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    startButton = (Button)findViewById(R.id.start_button);
    startButton.setOnClickListener(new OnClickListener(){
        @Override
        public void onClick(View v) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                startForegroundService(new Intent(this(),
MyService.class));
            } else {
                startService(new Intent(this(), MyService.class));
            }
        }
    })
}
```

```
public class MyService extends Service {

    private MyReceiver mReceiver;
    private IntentFilter mIntentFilter;

    @Override
    public void onCreate() {
        super.onCreate();
        // Receiver登録を実行
        registerScreenReceiver();
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        // Receiverを解除
        unregisterReceiver(mReceiver);
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        startInForeground();
        return START_STICKY;
    }

    // receiverを登録
    private void registerScreenReceiver() {
        mReceiver = new MyReceiver();
        mIntentFilter = new IntentFilter();
        mIntentFilter.addAction(Intent.ACTION_SCREEN_ON);
        mIntentFilter.addAction(Intent.ACTION_SCREEN_OFF);
        registerReceiver(mReceiver, mIntentFilter);
    }
}
```

```
// サービスをアライブさせるために通知を生成
private void startInForeground() {
    Intent notificationIntent = new Intent(this, MainActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
notificationIntent, 0);
    NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, NOTIFICATION_CHANNEL_ID)
        .setSmallIcon(R.mipmap.ic_launcher)

    .setContentTitle(getResources().getString(R.string.app_name))
        .setContentText("test notification")
        .setTicker(getResources().getString(R.string.app_name))
        .setContentIntent(pendingIntent);
    Notification notification = builder.build();

    if (Build.VERSION.SDK_INT >= 26) {
        NotificationChannel channel = new
NotificationChannel(NOTIFICATION_CHANNEL_ID
            , NOTIFICATION_CHANNEL_NAME
            , NotificationManager.IMPORTANCE_DEFAULT);
        channel.setDescription(NOTIFICATION_CHANNEL_DESC);

        NotificationManager notificationManager =
(NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        if (notificationManager != null) {
            notificationManager.createNotificationChannel(channel);
        }
    }
    startForeground(NOTIFICATION_ID, notification);
}
}
```

## 解説

---

- MyReceiverのonReceiveにて、intent.getAction()が登録したintentFilterのアクションと合致したらログを出す様に処理
- MainActivityにボタンを設置。ボタンを押すとサービスを起動するという単純な構造
- MyService内でReceiverの登録を行なっている。startInForegroundはForegroundService用でとりあえず実装（あまり気にしないでいいです）
- サービスが起動している間に、スクリーンをOn/Offにしたらその度にログが出る

## 備考

---

- 最ライフログやヘルスケアのアプリをよく見るので、どんな風にデータを集計しているのか気になったところ、intentを使っているパターンが多い様に感じたので調査してみた
- 今回紹介した以外にも、intentはものすごく沢山あるので、アプリによっては有用なものもある...かもしれない