

非推奨メソッド等の掲載について

「スマートにプログラミング Android 入門編 第2版 SDK 4/2.3 対応」(2012年6月11日発行 第2版第1刷)において、一部現在では非推奨となったクラス及びメソッドの利用が掲載されておりました。お詫びと共に補足・訂正させていただきます。下記にその対応について示します。

株式会社リックテレコム 書籍出版部
(情報更新日：2012.8.28)

非推奨クラス・メソッド	掲載箇所	推奨クラス・メソッド ¹
AbsoluteLayout	Chapter 4 レイアウトとビュー (132頁)	FrameLayout あるいは RelativeLayout といった、自分でカスタマイズしたレイアウトを代わりに使用するよう推奨されています。
onStart メソッド	Chapter 7 サービス 263頁以降数カ所	onStartCommand メソッドが推奨されています。二つのメソッドの比較については こちら をご参照下さい。また本文の訂正版 PDF は、 こちら をご参照下さい。
managedQuery メソッド	Chapter 9 コンテントプロバイダ 380頁以降数カ所	ContentResolver の query メソッドが推奨されています。また本文の訂正版 PDF は、 こちら をご参照下さい。
コンテキストメニュー 及び TabActivity	Chapter 10 ダイアログ、メニュー、タブ 「10-3 タブ」全体	Android3.0以降については次の二つの仕様変更・推奨がなされております。 ・メニュー： アクションバー (オーバーフローメニュー) 及び コンテキストアクションモード の使用。 ・タブ： Action Bar 及び Fragment の使用。本文の訂正及び追加となる PDF は、 こちら をご参照下さい。

¹ 2012年7月時点の推奨です。

● Service クラスの onStart()/onStartCommand()メソッド

1. 二つのメソッド

サービスは、アクティビティなどから startService()メソッドで起動されます。対象のサービスが起動していない場合、onCreate()メソッドが呼び出されます。その後、**onStart()メソッド**、または **onStartCommand()メソッド**が呼び出されます。

前者の onStart()メソッドは、Android 1.6(API レベル 4)までで使用されていたメソッドで、API レベル 5 以降は非推奨となり、代わりに onStartCommand()メソッドを使用するように推奨されました。したがって API レベル 5 以降を対象とする場合は、onStart()メソッドではなく、onStartCommand()メソッドを使用するようにしましょう。

開発対象が API レベル 5 以降の場合、onStart()がオーバーライドされていて、onStartCommand()がオーバーライドされていない場合、onStart()メソッドが呼び出されます。onStart()と onStartCommand()の両方がオーバーライドされている場合、onStartCommand()メソッドのみが呼び出されます。

API レベル 4 までを対象とする場合、onStart()メソッドと onStartCommand()メソッドの両方を実装します。

・ onStart メソッド :

```
void onStart(Intent intent, int startId)
```

・ onStartCommand()メソッド :

```
int onStartCommand(Intent intent, int flags, int startId)
```

2. onStartCommand()メソッドの戻り値

onStart()メソッドには、戻り値がありませんでしたが、onStartCommand()メソッドには int 型の戻り値があります。onStartCommand()メソッドをオーバーライドするときに、以下の定数のうちどれかを return することでサービスが強制終了されたあとのサービスの動作を指定できます。

・ **START_NOT_STICKY**

サービスが強制終了した場合、サービスは再起動しません。

不必要にサービスの再起動がされないため、サービスを安全に制御できます。

・ **START_STICKY**

サービスが強制終了した場合、サービスが再起動されます。onStartCommand()メソッド

ドが再度呼び出されますが、引数の Intent には null が渡されます。サービスの停止をしない限りは、サービスの実行を終了したくない音楽再生などの処理に使用します。

• **START_REDELIVER_INTENT**

サービスが強制終了した場合、サービスが再起動されます。onStartCommand()メソッドが再度呼び出されて、引数の Intent には直前に動作していたサービスに引き渡された Intent が渡されます。直前の処理を再開したいファイルダウンロードのような処理に使用します。

以下は、START_STICKY を使用した場合の例です。

```
****
public class SampleService extends Service {

(省略) . . . .

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    (省略) . . . .

    return START_STICKY;
}

}
****
```

なお onStart()メソッドは、START_STICKY と同様の動作となります。onStart()メソッドを使用した場合、または、onStartCommand()メソッドで START_STICKY を return した場合、サービスが再起動したときに引数の Intent が null になります。

そのため、onStart()メソッド、もしくは onStartCommand()メソッドの処理で Intent を使用する処理がある場合、Intent が null か否かの判定処理を記述したほうが良いでしょう。

3. onStartCommand()メソッドの引数

onStartCommand()メソッドの第 1 引数には、Intent が渡されます。これは、onStart()メソッドの第 1 引数と同様です。

第 2 引数には、そのサービスが通常起動か、再起動されたのかなどを示す以下の整数が引き渡されます (表 1)。

表 1

第 2 引数の値	状 態
0	通常起動したとき
1	onStartCommand()メソッドの戻り値が START_REDELIVER_INTENT で、サービスが強制終了後の再起動したとき
2	onStartCommand()メソッドから正しく return がなかった場合に、サービスの再呼び出しをしたとき

Service には、これらの判定用に以下の定数が用意されており、onStartCommand()メソッドの処理内で使用することが可能です。

・ **START_FLAG_REDELIVERY**

定数値は 1 で、START_REDELIVER_INTENT で、サービスが 強制終了後の再起動したのかの判定に使用

・ **START_FLAG_RETRY**

定数値は 2 で、サービスの再呼び出しをしたときの判定に使用

第 3 引数は、onStart()メソッドの第 2 引数と同様で、サービスを開始するリクエストを一意に判別するための ID で、stopSelfResult()メソッドの引数に指定することで、該当のサービスを停止することができます。

[最初の頁に戻る](#)