# Intent定义与主要功能

Intent是一个消息传递对象，一般使用它向其他应用组件请求操作。主要用途有以下三种：

（1）启动Activity。利用startActivity()或startActivityForResult()方法启动Activity。

（2）启动Service。利用startService()或是bindService()启动服务或绑定到服务。

（3）传递广播。利用sendBroadcast()、sendOrderedBroadcast()或LocalBroadcastManager发送广播。

# Intent类型

Intent有以下两种类型：

（1）显式Intent：按名称（完全指定类名）指定需要启动的部件，常使用在自己的应用内部，调用确定的应用部件完成任务。创建显式 Intent 启动 Activity 或服务时，系统将立即启动 Intent 对象中指定的应用组件。

（2）隐式Intent：不指定特定的组件，而是声明要执行的操作，从而允许其他应用的组件处理它。创建隐式Intent时，Android系统通过将Intent的内容与在设备上其他应用的Manifest文件中声明的IntentFilter进行比较，如果匹配，则系统将启动该部件；如果有多个匹配结果，系统会弹出一个对话框，让用户选择想要启动的组件。

需要注意的是，从Android5.0（API级别21）开始，如果使用隐式Intent启动Service，系统会引发异常。因此，必须始终使用显式Intent启动Service，且不为Service设置IntentFilter。

# Intent信息

Intent对象携带了让Android系统确定所要启动的组件的信息（准确的组件名称或应该接收该Intent的组件类别）以及收件人组件为了正确执行操作而使用的信息（要采取的操作或要处理的数据）。

Intent中包含的主要信息如下：

（1）组件名称componentname。可选项，如果为Intent指定了组件名称，则Intent是显式的，否则是隐式的。

Intent的这一字段是一个ComponentName对象，可以使用目标组件的完全限定类名（包含包名）指定该对象，或是利用setComponent()、setClass()、setClassName()或Intent构造函数设置组件名称。

（2）操作action。指定要执行的通用操作的字符串。操作在很大程度上决定了其余Intent的组成，特别是数据和Extra。

可以定义自己的操作，也可以使用Intent类定义好的操作。如果选择自定义操作，确保将包名作为前缀。

使用setAction()或Intent的构造函数设置操作。对于广播Intent，action通常是已经发生并且正在报告的操作。其余则为想要执行的操作。

（3）数据data。引用待操作数据|该数据MIME类型的URL(Url对象)。提供的数据一般由执行的操作action决定。

仅设置数据URL：使用setData()；

仅设置MINE类型：使用setType()；

同时设置数据URL与MINE类型：使用setDataAndType()（不能同时调用以上两个方法）。

（4）类别category。包含应处理Intent组件类型的附加信息的字符串，可以有任意个。

使用addCategory()添加类别。

（5）附加信息extra。通过以上4个属性已经可以让Android系统启动正确的组件，extra主要用在一些特定操作中。

利用putExtra()方法设置extra，方法接受两个参数：键名和值。一般使用Intent设置好的Extra，也可自行定义（需要加上包名作为前缀）。

（6）标志flag。在 Intent 类中定义的、充当 Intent 元数据的标志。 标志可以指示 Android 系统如何启动 Activity（例如，Activity 应属于哪个任务），以及启动之后如何处理（例如，它是否属于最近的 Activity 列表）。

# 使用Intent启动组件

显式Intent：利用Intent(Context,Class)，并利用startActivity(Intent)或startActivityForResult(Intent)启动组件。

隐式Intent：参阅

<https://developer.android.google.cn/guide/components/intents-common.html>

# 强制使用应用选择器

当有多个应用能够相应隐式Intent时，系统会弹出应用对话框供用户选择，并将其设置为该操作的默认选项。这在一些情况下很有用（浏览网页等），但在一些场合不适用（分享消息等）。

为保证每次都会弹出应用对话框，可以使用Intent.createChooser()方法创建Intent，并将其传递给startActivity()。

# 补充

如果设备上没有可接收隐式 Intent 的应用，您的应用将在调用 startActivity() 时崩溃。如需事先验证是否存在可接收 Intent 的应用，请对 Intent 对象调用 resolveActivity()。如果结果为非空，则至少有一个应用能够处理该 Intent，并且可以安全调用 startActivity()。 如果结果为空，则您不应使用该 Intent。如有可能，您应停用调用该 Intent 的功能。