# 定义

PendingIntent可以理解为一种延时生效的Intent，定义在它里面的行为可以在一段时间之后才被执行。将一个PendingIntent传递给其他应用的话，相当于给予了那个应用所有你所拥有的操作权限，就好像那是你自己的应用一样。因此，在构建PendingIntent时需要格外小心，最好将Intent的ComponentName显式设置成自己的组件名，从而确保它能够正确地得到执行。

# 创建

PendingIntent对象可以通过

getActivity(Context context, int requestCode, Intent intent, int flags)、

getBroadcast(Context context, int requestCode, Intent intent, int flags)、

getService(Context context, int requestCode, Intent intent, int flags)获得。

第一个参数是创建PendingIntent时的Context，

第二个参数是请求码，

第三个参数是需要启动的Intent，

第四个参数是标识位，有以下几个选项：

FLAG\_CANCEL\_CURRENT：如果要创建的PendingIntent已经存在，那么在创建新的之前先取消掉之前的。

FLAG\_IMMUTABLE：该PendingIntent是不可改变的，任何想要改变它的行为会被无视。

FLAG\_NO\_CREATE：如果想要创建的PendingIntent不存在，那么返回null而不是创建一个。

FLAG\_ONE\_SHOT：该PendingIntent是一次性的，执行完后自动取消。

FLAG\_UPDATE\_CURRENT：如果PendingIntent已经存在，那么只替换掉它的Extra部分。

PendingIntent中的内容由系统保存，即使它原本所属的应用进程已经被杀死了，PendingIntent本身还是能够使用。如果应用想要创建一个和之前的PendingIntent一模一样的PendingIntent（相同的操作、类别、数据、组件名、标识），那么系统会直接使用原来的（只要它还有效）而不是重新创建一个。可以使用cancel()方法销毁一个PendingIntent。

正因如此，了解在什么情况下会需要两个相同的Intent。人们常犯的一个错误是创建只有Extra部分不同的PendingIntent对象，并且希望每次都能得到一个不同的PendingIntent对象。这是不可能的，因为Extra部分不参与IntentFilter的匹配，因而你每次都会得到同一个PendingIntent。有以下两种方法可以解决这个问题：

（1）如果你确实需要同时拥有两个相同的PendingIntent，那么必须保证它们在创建过程中稍有不同，可以是不同的Intent属性，也可以是不同的requestCode。

（2）如果在一个时刻只需要一个PendingIntent，可以使用FLAG\_CANCEL\_CURRENT或者FLAG\_UPDATE\_CURRENT去取消或是修改PendingIntent。