

Intent隐式意图打开第三方应用的方法

一、通过要操作的意图动作、类别、数据来打开

以下采用隐式方式打开第三方应用的Activity(推荐)

```
//创建意图
Intent intent = new Intent();
//动作，必须 Intent.ACTION_VIEW 打开浏览器的动作
intent.setAction(Intent.ACTION_VIEW);
//类别，可选
intent.addCategory("android.intent.category.DEFAULT");
//数据，可选 打开浏览器要的是一个地址数据
intent.setData(Uri.parse(path));
//跳转到意图指写的Activity
startActivity(intent);
```

二、通过已知的第三方应用的包名，类名来打开

获取第三方应用的包名类名：

通过开发工具链接手机或模拟器（adb）打开任意第三方应用，在日志中可以获取到当前打开应用的包名，类名。

```
Intent intent = new Intent();

//参数一：包类
//参数二：包类.类名

intent.setClassName("com.android.browser","com.android.browser.BrowserActivity");

//真正跳转到意图指定的Activity
this.startActivity(intent);
```

强行打开界面的方法（通过广播打开某个界面时）

```
Intent myIntent = new Intent();
myIntent.setClass(context,MainActivity.class);
//强行创建任务栈，以便于MainActivity装入其中
myIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
context.startActivity(myIntent);
```

Activity通过意图传值到Activity 可以传的数据及方法

```
Intent putExtra(String name, String[] value)
Intent putExtra(String name, Parcelable value)
Intent putExtra(String name, long value)
Intent putExtra(String name, boolean value)
Intent putExtra(String name, double value)
Intent putExtra(String name, Parcelable[] value)
Intent putExtra(String name, char value)
Intent putExtra(String name, int[] value)
Intent putExtra(String name, int value)
Intent putExtra(String name, double[] value)
Intent putExtra(String name, short value)
Intent putExtra(String name, long[] value)
Intent putExtra(String name, boolean[] value)
Intent putExtra(String name, short[] value)
```

```

Intent putExtra(String name, String value)
Intent putExtra(String name, Serializable value)
Intent putExtra(String name, float[] value)
Intent putExtra(String name, Bundle value)
Intent putExtra(String name, byte[] value)
Intent putExtra(String name, CharSequence value)
Intent putExtra(String name, char[] value)
Intent putExtra(String name, byte value)
Intent putExtras(Intent src)
Intent putExtras(Bundle extras)

```

A类操作（传值）： JavaBean集合对象在传递前需要进行序列化

Android序列化对象主要有两种方法，实现Serializable接口、或者实现Parcelable接口。

实现Serializable接口是JavaSE本身就支持的，而Parcelable是Android特有的功能，效率比实现Serializable接口高，而且还可以用在进程间通信(IPC)中。

```
Intent intent = new Intent(mContext, CommodityAccountingActivity.class);
```

通过Bundle 传值

```

Bundle bundle = new Bundle();

bundle.putString("AAA", "A");

intent.putExtras(bundle);

```

```

intent.putExtra("TotalMoney", trim);
startActivity(intent);

```

B类操作（接收）：

通过Intent获取数据（注意传值与接收时的数据类型）

```

Intent extras = getIntent();
String totalMoney = extras.getStringExtra("TotalMoney");

```

通过Bundle获取数据（注意传值与接收时的数据类型）

```

Bundle extras1 = getIntent().getExtras();
String aaa = extras1.getString("AAA");

```

// ----存储----

```

intent.putExtra("id", 1); // int
intent.putExtra("name", "一碗单炒饭"); // String

```

// ----获取----

```

intent.getIntExtra("id", 0); // int
intent.getStringExtra("name"); // String

```

