

# Подробнее о Intents (неявные), IntentFilters

**Intent** – это механизм сообщений между компонентами приложений. Интенды используются как для создания или общения между компонентами внутри приложения, таких как Activity, Service или BroadcastReceiver, также и для передачи сообщений в другие, например для открытия камеры, отправки почты и прочее.

Intent делятся на два типа: **явные** и **неявные**.

**Явный (Explicit)** – это интент, который явно указывает на компонент, который нужно запустить. Он используется внутри своего приложения, так как мы точно знаем пакет и класс объекта, который хотим запустить. Например, открытие новой activity, запуск сервиса, отправка бродкаст сообщения.

**Неявный (Implicit)** – это интент, который не имеет информацию о компоненте, который будет выполнять действие. В этом случае мы передаем системе интент с action (действием), которое необходимо выполнить, а система (или пользователь) уже решают, какое приложение или компонент будет выполнять действие. Например, если мы хотим, чтобы из нашего приложения можно было сделать звонок, мы создаем intent с action ACTION\_DIAL. Получив это сообщение, система находит все приложения, которые могут выполнить звонок. Если таких приложений больше одного предложит пользователю выбрать, то, которое нужно использовать, если приложение одно, система откроет его. Если приложений нет, то будет ошибка, поэтому в вызов неявных интендов обязательно нужно добавить обработку ошибок.

Интент содержит информацию, при помощи которой система определяет, какой компонент запустить. В нее может входить:

- **Имя компонента**, который необходимо запустить. Это – показатель явного интенда. Если не заполнить этот параметр, то мы оставляем выбор получателя интенда за системой. Обычно именем компонента является имя класса. Задать его можно через сетеры setComponent(), setClass(), setClassName() или через конструктор. Выглядит это так:

```
Intent intent = new Intent(this, MainActivity.class); //  
MainActivity.class - имя компонента
```

- **Action** – действие, которое получатель интенда должен выполнить, открыть камеру, контакты, поделиться текстом и прочее.

Также помимо стандартных экшенов, можно создать свои для общения внутри приложения (но это не рекомендуется) или для вызовов компонентов своего приложения из других.

Что бы добавить action в созданный интент, напишите:

```
intent.setAction(Intent.ACTION_SEND);
```

При отправке action также можно добавить data, это дополнительные данные которые будет использовать получатель action, например мы хотим добавить возможность позвонить сразу из нашего приложения, для этого мы добавляем action, но чтобы приложение телефона знало кому мы будем звонить, мы должны передать ему номер, выглядеть это будет так:

```
intent.setAction(Intent.ACTION_CALL);  
intent.setData(Uri.parse("tel:799912345678"));
```

- **Category** – категория.

Это дополнительные сведения для системы. Большинству интенгов категория не нужна, но некоторые категории можно рассмотреть.

Intent.CATEGORY\_LAUNCHER – категория, которая определяет открываемую activity, при запуске приложения с домашнего экрана телефона

Intent.CATEGORY\_BROWSABLE – данная категория показывает, что activity можно запустить с браузера, например, при нажатии на ссылку или электронную почту

- **Extras** – дополнительные данные

Extras представлены в виде пары ключ – значение.

В экстрас мы можем добавить примитивы или объекты, которые реализуют интерфейсы Parcelable или Serializable, а также массивы этих объектов.

Не стоит пересылать через экстрас большое количество данных, так как размер пересылаемых данных ограничен, на разных версиях андроида лимит отличается, но в среднем это около 500 килобайт.

Добавить экстра можно следующим образом:

```
intent.putExtra("user_id_key", 42);
```

- **Flags** – флаги. Флаги используются при открытии activity, они указывают системе, в каком backstack tasks создать activity и каким образом она будет создана.

Например, если мы хотим чтобы наша activity была единственной в стеке, а все остальные activity были уничтожены системой, нужно добавить такие флаги:

```
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |  
Intent.FLAG_ACTIVITY_NEW_TASK);
```

После создания интенга, его нужно использовать, отправив в качестве аргумента в соответствующий метод.

Для запуска activity используется метод

```
startActivity(intent);
```

Для запуска сервиса

```
startService(intent);
```

Для отправки широковещательного сообщения

```
sendBroadcast(intent);
```

Если нужно, чтобы вторая activity, запущенная из первой, после своей работы возвращала результат в первую activity, то для запуска второй activity надо использовать метод

```
startActivityForResult(intent, 42);
```

где число – requestCode – это код запроса, используемый для определения activity, с которого вернется результат. Это обязательный параметр, так как обработка всех результатов выполняется в методе

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data)
```

здесь мы можем найти наш запрос по requestCode, получить результат выполнения resultCode, и данные, которые будут храниться в extras интента. resultCode – RESULT\_OK, RESULT\_CANCELED – системные флаги результата, плюс можно дополнительно определить свои.

## Intent-filters

Интент фильтр указывает системе, какие типы неявных интентов activity сможет обработать. К примеру, если вы, будучи в галерее фотографий нажмете на кнопку поделиться, то всплывет список с вариантами приложений, через которые вы сможете переслать фотографию. В этом случае в галерее используется неявный интент с action ACTION\_SEND, а в приложениях у соответствующих activity настроен интент-фильтр.

```
<intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="image/*" />
</intent-filter>
```

Здесь мы указываем action, category и данные которыми можно поделиться через наше приложение.

категория DEFAULT – применяется ко всем неявным интентам, поэтому для того чтобы получить неявный интент мы должны обозначить эту категорию.

Если наше приложение поддерживает передачу текста, то мы можем добавить

```
<data android:mimeType="text/plain" />
```

Также мы можем добавить схему принимаемого экшена или полностью путь.

Например при нажатии на строку <https://www.e-legion.com/> из любого приложения в телефоне, вместо браузера, открывалось наше приложение

```
<intent-filter>

    <action android:name="android.intent.action.VIEW" />

    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />

    <data android:scheme="https" />
    <data android:host="e-legion.com" />
</intent-filter>
```

Подробнее про интент фильтры можно прочитать в документации.  
<https://developer.android.com/guide/components/intents-filters.html>



