

3.4. Неявные интен

Сайт: [Samsung Innovation Campus](#)
Курс: Мобильная разработка на Kotlin
Книга: 3.4. Неявные интен

Напечатано:: Murad Rezvan
Дата: понедельник, 3 июня 2024, 17:50

Оглавление

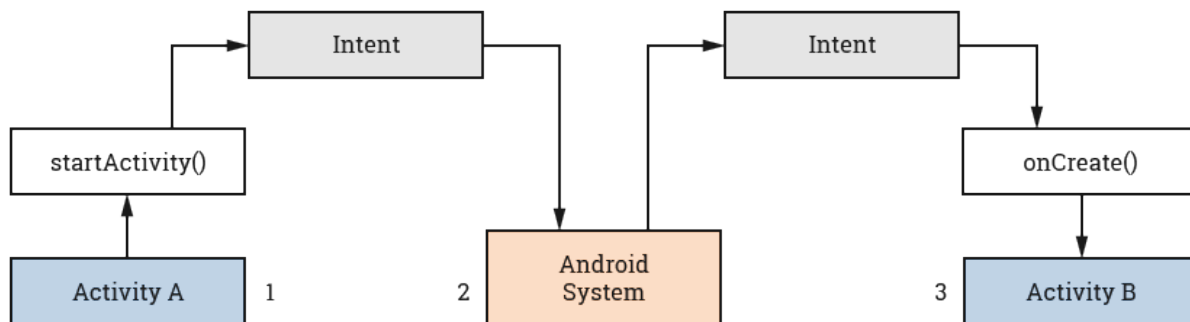
Неявные интен

- Возврат результата из активности

Неявные интен­ты

Явные интен­ты удобны, когда вызываемая активность находится в вашем же приложении, но часто это не так. Операционная система даёт широкие возможности для использования "внешних" приложений: телефонная книга, географическая карта, выбор изображения из галереи или камеры. Направления вызова (интента) вовне вашего приложения реализуется с помощью неявных интен­тов. Удобно, что каждому разработчику не нужно создавать с нуля требуемую функциональность, например, реализовывать телефонный вызов.

При создании объекта Intent вместо класса компоненты вы указываете тип действия, категорию и адрес (или дополнительные сведения) для вызова стороннего приложения. Система проводит поиск приложений, которые смогут обработать такой вызов. Если находится несколько соответствующих условию активностей, пользователю предоставляют выбор, какой именно программой вы хотите воспользоваться. ОС Android сверяет набор параметров вызываемого объекта Intent и наборы параметров Intent Filter для каждой активности. Если наборы совпадают, то активность считается подходящей. Схематично вызов неявного интента можно изобразить так:



Механизм вызова неявного интента очень похож на запрос web-страницы (HTTP): вы так же указываете Uri и действие (action), которое нужно выполнить. В вашем распоряжении сотни различных констант вида `ACTION_...`, среди них такие как:

- `ACTION_VIEW` - показать или обработать содержимое произвольного Uri (веб-страница, гео-точка, номер телефона)
- `ACTION_PICK` - выбор объекта из хранилища или коллекции
- `ACTION_SEND` - передача объекта в другое приложение, например, отправка SMS или публикация фото в Instagram

В качестве простого примера рассмотрим вызов браузера для отображения определённого адреса:

```
startActivity(Intent(Intent.ACTION_VIEW, Uri.parse("https://ya.ru")))
```

Важно отметить, что метод `startActivity(...)` является асинхронным: активность, вызывает другую активность с помощью интента, продолжает работать и ничего "не знает" о результатах работы вызванной активности. Часто такое поведение приемлемо, но в ряде случаев нам потребуется вернуть результат в исходную активность, например контакт из телефонной книги.

Возврат результата из активности

В ряде случаев требуется не только вызвать другую активность или приложение, но и получить из него сведения: контакты, изображения, файл, ссылку и т.п. Для возврата результата в исходную активность мы используем асинхронный механизм, т.е. после вызова активности приложение может продолжить работу ожидая возвратного вызова. С помощью метода `startActivityForResult()` мы вызываем новую активность, передаём объект `Intent` и нумеруем при этом каждый вызов параметром `requestCode` (чтобы впоследствии определить, возврат из какой активности мы обрабатываем). В качестве примера рассмотрим выбор абонента из списка контактов:

```
fun onPickContact(view: View) {
    val intent = Intent(Intent.ACTION_PICK, ContactsContract.Contacts.CONTENT_URI)
    startActivityForResult(intent, PICK_CONTACT)
}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == PICK_CONTACT && resultCode == Activity.RESULT_OK) {
        if (data != null ) {
            Log.d("Intent_demo", data.dataString)
        }
    }
}
```

Метод `onActivityResult()` получает в качестве параметра `requestCode` (чтобы определить, возврат из какой активности мы обрабатываем), `resultCode` (в норме равный константе `RESULT_OK`) и объект `Intent` (содержимое ответа). Объект `Intent`, в свою очередь, содержит ссылку или перечисление переданных сведений (например, выбранных контактов из телефонной книги).

[Начать тур для пользователя на этой странице](#)