

akademia androida



Intencje oraz URI część III

agenda

1. Czym są intencje i po co się ich używa ?
2. Czym jest URI ?
3. Zadanie 1.
4. Zadanie 2 (domowe).

1. Czym są intencje i po co się ich używa ?

Intencje są to „wiadomości”, które pozwalają aplikacji/komponentowi systemu Android żądać jakiejś funkcjonalności od innych aplikacji/komponentów. Przechowują one informację o akcji do wykonania. Inne komponenty mogą się podłączać do takich akcji poprzez filtry intencji o których później. Przykłady akcji zdefiniowanych w klasie `Intent`:

Stała	Cel	Akcja
<code>ACTION_CALL</code>	aktywność	Zainicjowanie rozmowy telefonicznej.
<code>ACTION_VIEW</code>	aktywność	Wyświetlenie danych użytkownikowi.
<code>ACTION_BATTERY_LOW</code>	broadcast receiver	Ostrzeżenie o niskim stanie baterii.
<code>ACTION_HEADSET_PLUG</code>	broadcast receiver	Zestaw słuchawkowy został podłączony lub odłączony.
<code>ACTION_SCREEN_ON</code>	broadcast receiver	Włączono ekran.

Wyróżniamy trzy mechanizmy dostarczania intencji, aby uruchomić określone komponenty:

- do uruchamiania aktywności intencje przekazujemy do
 - `Context.startActivity()`
 - `Activity.startActivityForResult()`
- do uruchamiania usług do
 - `Context.startService()`
- przekazywanie akcji do broadcast receiverów.

Rozróżniamy dwa typy intencji:

- **jawne** – są to takie intencje, w których jawnie deklarujemy, który komponent ma wykonać akcję, np.

```
Intent i = new Intent(this, DrugaAktywnosc.class);
startActivity(i);
```

- **niejawne** – w tego typu intencjach nie wskazujemy bezpośrednio obiektu, jedynie akcję, którą chcemy wykonać. Jeżeli nie mamy domyślnego komponentu, który np. otwiera strony WWW otworzy się dialog z pytaniem, której aplikacji chcemy użyć. System rozpoznaje komponenty, które są w stanie wykonać daną akcję na podstawie filtrów intencji.

```
Intent i = new Intent(Intent.ACTION_VIEW,
                    Uri.parse("http://www.mad.zut.edu.pl"));
startActivity(i);
```

Można również **dodać dane do intencji**, które odbierze wywołany komponent. Przykładowo:

```
Intent i = new Intent(this, DrugaAktywnosc.class);
i.putExtra("Wiadomosc", "To jest wiadomosc do aktywnosci drugiej");
i.putExtra("ID", 12345);
startActivity(i);
```

W drugiej aktywności dane odbieramy poprzez pobranie intencji i wywołanie na niej stosownej metody w zależności od typu danych.

```
Intent intent = getIntent();
String message = intent.getStringExtra("Wiadomosc");
int id = intent.getIntExtra("ID", 0);
```

Jeżeli chcemy otworzyć jakąś aktywność po to, żeby zwrócić nam jakieś dane, zamiast `Context.startActivity()` używamy `Activity.startActivityForResult()`.

```
Intent intent = new Intent(this, DrugaAktywnosc.class);
startActivityForResult(intent, SECOND_ACT_REQUEST);
```

`SECOND_ACT_REQUEST` jest to dowolna stała całkowitoliczbowa, dzięki której będziemy wiedzieli skąd przyszła odpowiedź do naszej aktywności.

```
private static final int SECOND_ACT_REQUEST = 1234;
```

W drugiej aktywności konstruujemy odpowiedź poprzez stworzenie zwrotnej intencji z dowolnymi danymi:

```
Intent resultIntent = new Intent();
resultIntent.putExtra("Odpowiedz", "Jakaś nasza odpowiedz");
setResult(RESULT_OK, resultIntent);
```

Odpowiedź odbieramy poprzez nadpisaną w klasie `Activity` metodę `onActivityResult(int requestCode, int resultCode, Intent data)`

- **requestCode** - kod identyfikujący komponent, od którego otrzymaliśmy odpowiedź,
- **resultCode** – zmienna całkowitoliczbowa, na podstawie której określamy czy żądana przez nas operacja została wykonana poprawnie (`Activity.RESULT_OK`, `Activity.RESULT_CANCELED`),
- **data** – dane, które otrzymaliśmy.

Przykład takiej metody:

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (SECOND_ACT_REQUEST == requestCode) {
        if (resultCode == RESULT_OK) {
            // wykonaj coś
        }
    }
}
```

2. Czym jest URI ?

URI to dosłownie „jednolity identyfikator zasobu”, pozwala on na identyfikację zasobów. Określa on nazwę (URN) lub adres (URL) zasobu, który jest identyfikowany przez dane URI. W praktyce oznacza to tyle, że poza żądaniem dokumentów w sieci WWW (tak jak było to pokazane w przykładzie) można też żądać innych danych takich jak: nr. telefonów, lokalizację etc. Więcej na temat w następnjej części kursu. Kilka przykładów poniżej:

Adres e-mail	mailto:
URI pliku	file://
URI folderu	folder:// directory://
URI lokalizacji	geo:
Google Streetview	google.streetview:
Numer telefonu	tel:
Adres WWW	http:// https://

Budowa URI na podstawie adresu WWW:

http://www.jakis-serwer.pl:8080/katalog1/katalog2/plik?parametr1=wartosc1¶metr2=wartosc2#fragment_dokumentu

/ _/_ _/_ _/_ _/_ _/_ _/_ _/_

| | | | | | |

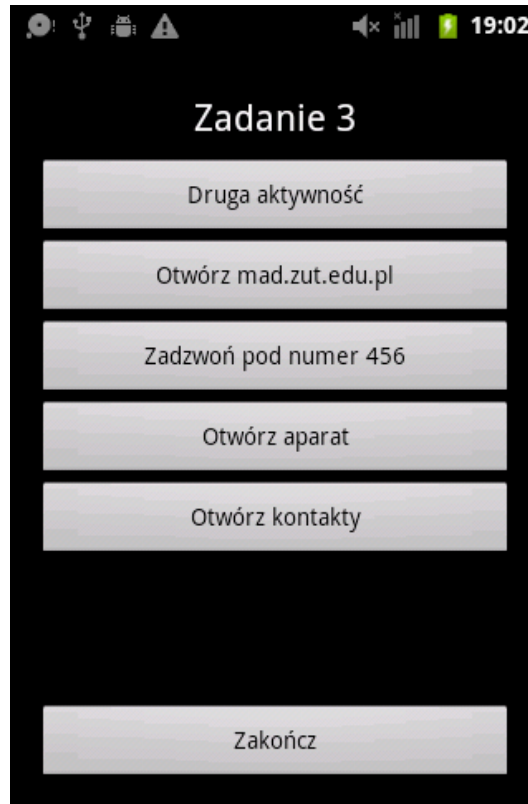
schemat host port ścieżka do pliku zapytanie fragment

(protokół) (nazwa serwera)

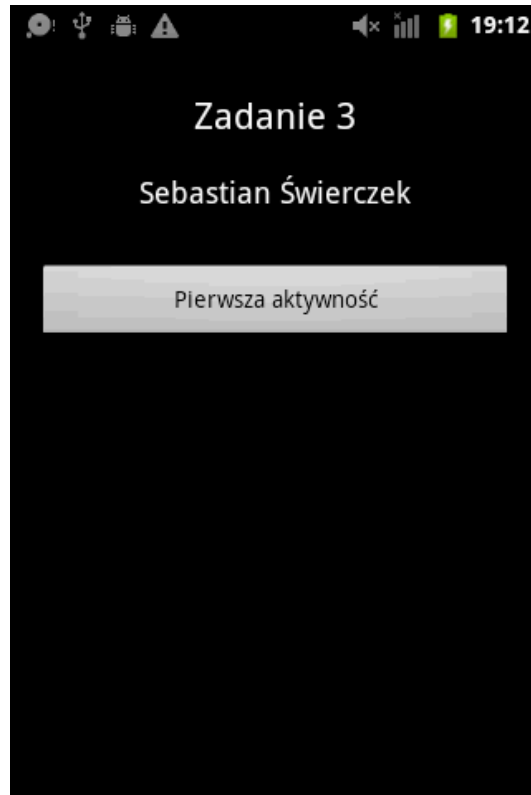
Źródło : http://pl.wikipedia.org/wiki/Uniform_Resource_Identifier

3. Zadanie 1.

- Pobierz zadanie domowe z poprzedniego spotkania.
http://www.mad.zut.edu.pl/akademia_androida/zadanie_2_dom.zip
- Dodaj przyciski wg poniższego zrzutu ekranu:



- Używając intencji zaprogramuj akcję dla każdego z przycisków wg jego nazwy.
- Przycisk „Druga aktywność” poza uruchomieniem drugiej aktywności powinien przekazywać do niej Twoje imię i nazwisko. W drugiej aktywności należy otrzymane dane wyświetlić w TextView. Przykład poniżej.



Uwaga !

Pamiętaj o dopisaniu uprawnień do pliku `AndroidManifest.xml`.

4. Zadanie 2 (domowe).

Stwórz aplikację, w której będą znajdować się dwie aktywności. W pierwszej dodaj przycisk „Żądaj odpowiedzi” oraz pole `TextView`. Przycisk „Żądaj odpowiedzi” powinien otworzyć drugą aktywność w celu odebrania od niej tekstu wpisanego do pola `EditText`. Poza polem `EditText`, druga aktywność powinna zawierać przycisk „Odpowiedź”, po naciśnięciu którego powrócimy do aktywności pierwszej, a w polu tekstowym wyświetli się podany przez nas napis.

Dziękuję za uwagę !

Kontakt:

- sswierczek@wi.zut.edu.pl
- mad@zut.edu.pl