

Android 3. Intencje

W androidzie, różne komponenty aplikacji komunikują się za pomocą intencji(intents). Intencji używamy do zażądania od innej aktywności lub aplikacji wykonania akcji takich jak wystartowanie aktywności, serwisu, czy wysłania wiadomości do innych aplikacji.

Intencje możemy podzielić na:

Jawne(Explicit Intents) – określa, który komponent ma być uruchomiony – np. nowa aktywność w aplikacji. Odwołanie przez pełną nazwę.

Niejawne(Implicit Intents) – gdy nie wiadomo jaki komponent zostanie użyty. Np otworzenie przeglądarki lub mapy. Odwołanie przez określenie funkcji jaką komponent musi spełniać.

Filtr intencyjny czyli “Intent filter”, to wyrażenie określające, jakie intencje komponent może obsłużyć. Komponent B może być uruchomiony przez inne komponenty dzięki deklaracji takiego filtru. Bez tej deklaracji aktywność może zostać uruchomiona jedynie poprzez jawne wywołanie.

Intencja składa się z nazwy komponentu, akcji, danych, kategorii, danych dodatkowych i flag.

Akcja - ogólne określenie akcji, jaka ma być podjęta. Można zdefiniować swoją własną nazwę akcji, ale zazwyczaj stosuje się zdefiniowane już nazwy możliwych intencji, np.:

Intent.ACTION_VIEW – Otwiera dane, np. stronę www, mapy.

Intent.ACTION_SENDTO – Wysyłanie wiadomości, np. SMS lub e-mail.

Intent.ACTION_SEND – Udostępnianie treści, np. tekstu, plików.

Intent.ACTION_DIAL – Otwiera aplikację telefonu z wypełnionym numerem.

Intent.ACTION_CALL – Wykonuje połączenie telefoniczne (wymaga uprawnień).

Intent.ACTION_MAIN – Uruchamia główną aktywność aplikacji.

Dane, na których ma operować intencja. Jest to URI określające odwołanie do danych, na których ma być wykonana intencja i/lub typ danych. Dla ACTION EDIT powinno się pojawić URI dokumentu do edycji.

Kategorie dodają dodatkowe informacje do intencji i pomagają określić, jakie komponenty mogą zostać uruchomione w odpowiedzi na tę intencję. Są używane głównie do określenia kontekstu działania:

Intent.CATEGORY_DEFAULT – Domyślna kategoria dla intencji, jeśli nie ma innej kategorii.

Intent.CATEGORY_LAUNCHER – Umożliwia wyświetlenie aktywności w launcherze jako ikona aplikacji.

Intent.CATEGORY_BROWSABLE – Pozwala otwierać aktywność przez linki internetowe (np. URL).

Intent.CATEGORY_APP_BROWSER – Wskazuje, że aplikacja może pełnić funkcję przeglądarki internetowej.

Intent.CATEGORY_APP_EMAIL – Wskazuje, że aplikacja może pełnić funkcję programu pocztowego (e-mail).

Intent.CATEGORY_APP_MUSIC – Wskazuje, że aplikacja może pełnić funkcję odtwarzacza muzyki.

Przykład konfiguracji głównej aktywności, która jest widoczna w launcherze i uruchamiana, gdy aplikacja zostanie otwarta:

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Aby uruchomić 2 aktywność z pierwszej i przesłać tam dane, można użyć intencji jawnej z przekazaniem dodatkowej wartości.

W głównej aktywności:

```
var inputText by remember { mutableStateOf("") }

Column {
    TextField(
        value = inputText,
        onChange = { inputText = it },
        label = { Text("Enter text") },
        modifier = Modifier.fillMaxWidth()
    )
    Spacer(modifier = Modifier.height(16.dp))

    Button(onClick = {
        openSecondActivity(inputText)
    }, modifier = Modifier.fillMaxWidth()) {
        Text("Open Second Activity")
    }
}
```

W funkcji openSecondActivity użyjemy klasy **Intent** z argumentami:

-this – skąd intencja jest wywoływana - główna aktywność.

SecondActivity::class.java – referencja do klasy, która ma być uruchomiona (JVM używa .java a Kotlin przekształca KClass w Class.)

-tekstem do przekazania:

```
private fun openSecondActivity(data: String) {
    val intent = Intent(this, SecondActivity::class.java).apply {
        putExtra("message", data)
    }
    startActivity(intent)
}
```

.apply przyjmuje lambda jako argument.

Wewnątrz lambda dostępny jest skonfigurowany obiekt Intent.

.apply zwraca ten sam obiekt, który był skonfigurowany (Dodatkowe dane w putExtra)

```
startActivity(intent)
```

Metoda z klasy Activity. Uruchamia drugą aktywność.

Potrzeba jeszcze klasy z drugą aktywnością:

```
class SecondActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        var receivedText = intent.getStringExtra("message")

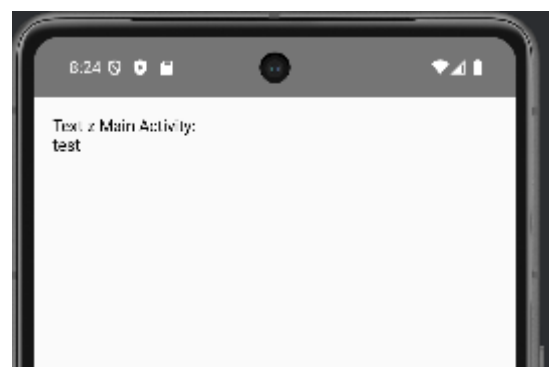
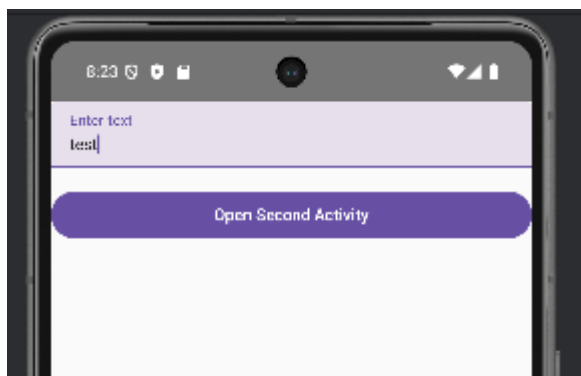
        setContent {
            SecondScreen(receivedText ?: "No message")
        }
    }

    @Composable
    fun SecondScreen(text: String) {
        Column(modifier = Modifier.padding(16.dp)) {
            Text(text = "Message from Main Activity:")
            Text(text = text)
        }
    }
}
```

`var receivedText = intent.getStringExtra("message")` – pobiera dane przekazane przez Intencję z poprzedniej aktywności.

Należy także dodać 2 aktywność w manifestcie!

```
<activity android:name=".SecondActivity" />
```



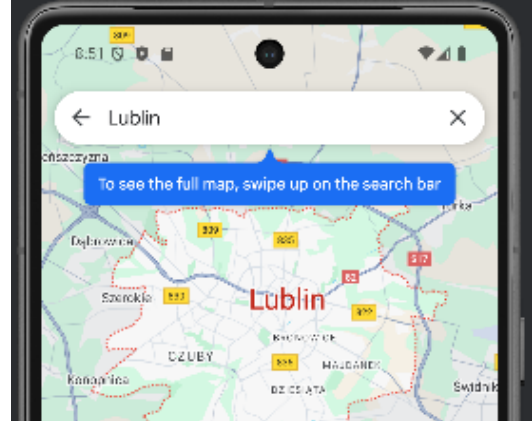
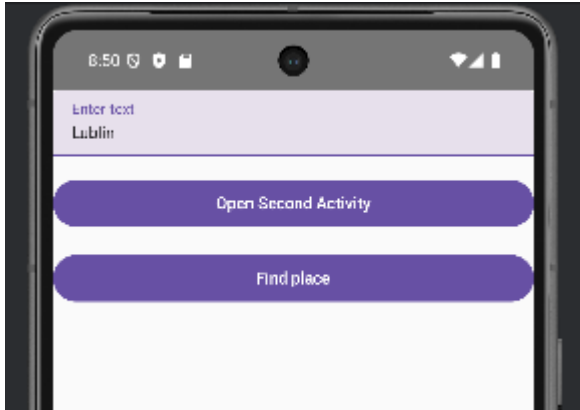
Aby uruchomić inną aplikację w Androidzie użyjemy niejawnej intencji.

Np. przekazując dane o szukanym miejscu na mapie użyjemy `Intent.ACTION_VIEW` przekazując data – Uri z protokołem geo:

```
private fun openMap(location: String) {
    val intent = Intent(Intent.ACTION_VIEW).apply {
        data = Uri.parse("geo:0,0?q=$location")
    }
    startActivity(intent)
}
```

Przycisk wywołujący akcję:

```
Spacer(modifier = Modifier.height(16.dp))
Button(onClick = {
    openMap(inputText)
}, modifier = Modifier.fillMaxWidth()) {
    Text("Find place")
}
```



Więcej o intencjach:

<https://developer.android.com/reference/android/content/Intent>

Przykłady użycia innych aplikacji:

<https://developer.android.com/guide/components/intents-common>

Zadanie3:

Stwórz aplikację, która będzie składała się z dwóch pól edycyjnych i 9 kafelków będących przyciskami. Ich funkcje:

1A. Aktywność A wysyła treść z drugiego pola edycyjnego pod numer telefonu zapisany w pierwszym (SMS)

2A. Aktywność A wyszukuje miejsce o podanej w pierwszym polu nazwie (mapa)

3A. Uruchamia aplikację do tworzenia dokumentów

4A. Uruchamia aplikację mejlową z wypełnionym adresem z pierwszego pola i tytułem mejla z drugiego pola edycyjnego

5A. Uruchamia telefon z numerem z drugiego pola edycyjnego

6A. Uruchamia odtwarzacz muzyki

7A. Uruchamia drugą aktywność (B), której przekazuje listę kontaktów z obu pól edycyjnych.

Aktywność B wyświetla te kontakty.

W drugiej aktywności powinien być przycisk do zamykania aktywności B i drugi. do przejścia na aktywność A.

8A. Uruchamia kalendarz i przekazuje mu z pierwszego pola edycyjnego datę i godzinę spotkania, a z drugiego pola nazwę spotkania.

9A. Zamyka aplikację

Importy z przykładowej aplikacji:

MainActivity:

```
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.provider.CalendarContract
import android.util.Log
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.BasicTextField
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.dp
import java.text.SimpleDateFormat
import java.util.Locale
```

SecondActivity:

```
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.*
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.dp
```