

Week 4 - oefening 2: Intent Filters

Doelstelling

In dit labo onderzoek je het gebruik van implicit intents, intentfiltering,...

Vorbereitung

Maak een nieuw Android Application Project aan met volgende instellingen.

Application Name:	Demo Intent Filters
Package Name:	be.howest.nmct.intentfiltersproject
Activity/fragment:	MainActivity & MainFragment
Target SDK:	21 (of eerder)

Design

De gevraagde (aan te maken) activiteiten vragen geen bijzonder design.

Code

Ga zelf even op onderzoek uit door onderstaande vragen te beantwoorden. De achtergrond (en te kennen leerstof) is te vinden op:

<http://developer.android.com/training/basics/intents/sending.html>

<http://developer.android.com/guide/components/intents-filters.html>

Onderzoek 1: bestudeer het AndroidManifest-file.

Maak een nieuwe java klasse 'Constants' aan en plaats hierin de constante:

```
public static final String ACTION_IMPLY="be.howest.nmct.intentfiltersproject.intent.action.IMPLY";
```

Plaats in de layout MainActivity een Button met daaraan de methode launchWithAction

```
public void launchWithAction(View v) {
    Intent intent = new Intent(Constants.ACTION_IMPLY);
    startActivity(intent);
}
```

Unfortunately, Demo Intent Filters has stopped.

OK

Wat merk je? Welke exception(s) krijg je te zien?

Pas de code zodat je een Toast message ziet als de Intent niet bestaat. Maak gebruik van `queryIntentActivities`. Baseer je op:

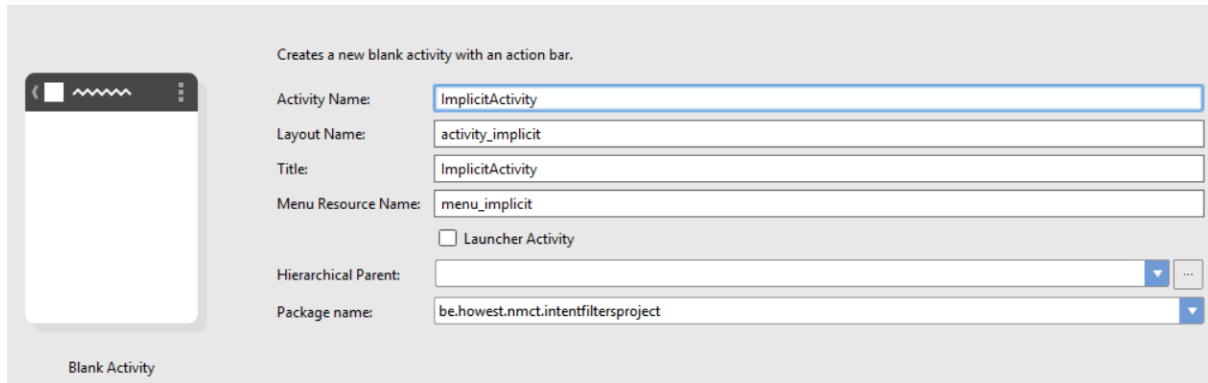
<http://developer.android.com/training/basics/intents/sending.html#Verify>

```
03-04 15:52:22.455 3161-3161/bw.howest.mmc.intensiflirproject E/AndroidRuntime: FATAL EXCEPTION: main
Process: bw.howest.mmc.intensiflirproject, PID: 3161
android.content.ActivityNotFoundException: Activity found to handle Intent ( act=bw.howest.mmc.intensiflirproject.Intent.ACTION_IMPORT )
at android.app.Instrumentation.execFinish(Activity.java:793)
at android.app.Instrumentation.execFinish(Activity.java:793)
at android.app.Activity.startActivityForResult(Activity.java:2319)
at android.app.Activity.startActivityForResult(Activity.java:2637)
at android.app.Activity.startActivityForResult(Activity.java:2637)
```



Onderzoek 2:

Maak een nieuwe Blank Activity 'ImplicitActivity' aan met volgende parameters:



Voeg in het AndroidManifest-file een Intentfilter toe voor deze nieuwe activity

```
<activity
    android:name="be.howest.nmct.intentfiltersproject.ImplicitActivity"
    android:label="@string/title_activity_implicit" >

    <intent-filter>
        <action android:name="be.howest.nmct.intentfiltersproject.intent.action.IMPLY" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

<category>
Declares the intent category accepted, in the **name** attribute. The value must be the literal string value of an action, not the class constant.

Note: In order to receive implicit intents, you **must include** the **CATEGORY_DEFAULT** category in the intent filter. The methods **startActivity()** and **startActivityForResult()** treat all intents as if they declared the **CATEGORY_DEFAULT** category. If you do not declare this category in your intent filter, no implicit intents will resolve to your activity.

Vragen:



- Deploy de activity opnieuw en druk op de knop. Wat gebeurt nu?
- Wat gebeurt er als we de category tag in commentaar zetten? Ga in de documentatie na waarvoor de Default category dient? Is hij altijd nodig?
- Blijft de toepassing werken indien we een tweede category (in het AndroidManifest-file) toevoegen met de naam android.intent.category.LAUNCHER?

Pas jouw code nu als volgt aan:

```
Intent intent = new Intent(Constants.ACTION_IMPLY);
intent.addCategory(Intent.CATEGORY_CAR_DOCK);
```

Vragen:



- werkt nu de methode nog? Waarom niet?
- Pas het AndroidManifest-file zodat de methode opnieuw werkt.

For an intent to pass the category test, every category in the **Intent** must match a category in the filter. The reverse is not necessary—the intent filter may declare more categories than are specified in the **Intent** and the **Intent** will still pass. Therefore, an intent with no categories should always pass this test, regardless of what categories are declared in the filter.

Beschrijf nu zelf hoe Android omgaat met categoriën in de intent en de intentfilter om te beslissen of een activity al dan niet voldoet.



Zie ook op:

<http://developer.android.com/guide/components/intents-filters.html#Resolution>

Onderzoek 3:

Van waar komt de waarde die in het android:name attribuut van de action staat?

Blijft de toepassing werken indien we in het AndroidManifest-file een tweede action met de name android.intent.action.VIEW toevoegen?

Wat gebeurt er als we de code nu als volgt aanpassen? Hoe komt dit?

```
Intent intent = new Intent("android.intent.action.VIEW");
startActivity(intent);
```



Action test

To specify accepted intent actions, an intent filter can declare zero or more `<action>` elements. For example:

```
<intent-filter>
  <action android:name="android.intent.action.EDIT" />
  <action android:name="android.intent.action.VIEW" />
  ...
</intent-filter>
```

To get through this filter, the action specified in the `Intent` must match one of the actions listed in the filter.

If the filter does not list any actions, there is nothing for an intent to match, so all intents fail the test. However, if an `Intent` does not specify an action, it will pass the test (as long as the filter contains at least one action).

Onderzoek 4:

Pas de `IntentFilter` in het `AndroidManifest.xml` bestand als volgt aan:

```
<activity
  android:name="be.howest.nmct.intentfiltersproject.ImplicitActivity"
  android:label="@string/title_activity_implicit" >
  <intent-filter>
    <action android:name="be.howest.nmct.intentfiltersproject.intent.action.IMPLY" />

    <data android:scheme="xtp" />

    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

Voeg in de `MainActivity` een extra button toe en koppel aan die button volgende methode. Deze methode ziet er als volgt uit:

```
private void onderzoek4(){
    //Intent intent = new Intent(Constants.ACTION_IMPLY, Uri.parse("xtp:///somedata"));
    //startActivity(intent);

    // test zonder data
    Intent intent = new Intent(Constants.ACTION_IMPLY);
    startActivity(intent);
}
```

When the URI in an intent is compared to a URI specification in a filter, it's compared only to the parts of the URI included in the filter. For example:

- If a filter specifies only a scheme, all URIs with that scheme match the filter.
- If a filter specifies a scheme and an authority but no path, all URIs with the same scheme and authority pass the filter, regardless of their paths.
- If a filter specifies a scheme, an authority, and a path, only URIs with the same scheme, authority, and path pass the filter.

Wat werkt? Verklaar nader. Welke attributen heeft de `data`-tag nog en waarvoor dienen ze? Merk op dat er met de `data` in de tweede activity nog niets gedaan wordt.



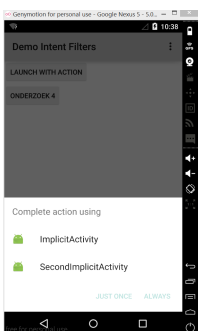
Meer info terug te vinden op:

<http://developer.android.com/guide/components/intents-filters.html#Resolution>

<http://developer.android.com/guide/topics/manifest/data-element.html>

Onderzoek 5:

Maak een nieuwe Blank Activity 'SecondImplicitActivity' aan met volgende parameters



Creates a new blank activity with an action bar.

Activity Name:

Layout Name:

Title:

Menu Resource Name:

☐ Launcher Activity

Hierarchical Parent:

Package name:

Blank Activity

Zorg dat deze (in het `AndroidManifest.xml`) dezelfde `IntentFilter` als `ImplicitActivity` heeft. Wat gebeurt er nu als je op de tweede button duwt? Hoe gaat android hiermee om?



Onderzoek 6:

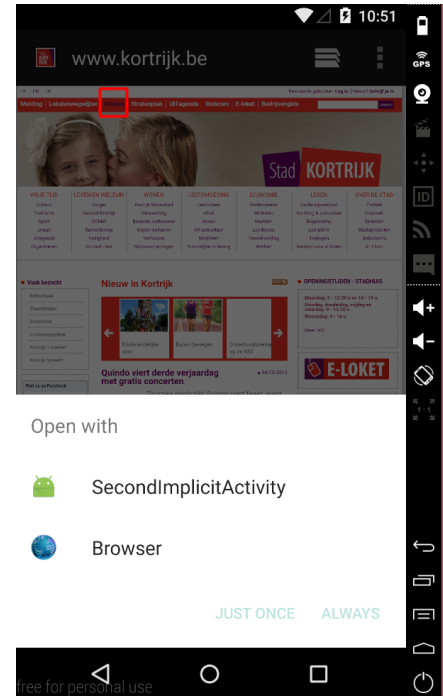
Pas de IntentFilter in het AndroidManifest.xml bestand als volgt aan. Surf nu met de browser op je toestel naar <http://www.kortrijk.be> en klik op nieuws. Wat gebeurt er? Blijft dit gebeuren als de BROWSABLE categorie er niet is? Wat zegt de documentatie over BROWSABLE? Zijn er nog zulke speciale categoriën?

```
<activity
    android:name="be.howest.nmct.intentfiltersproject.SecondImplicitActivity"
    android:label="@string/title_activity_second_implicit" >

    <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:host="www.kortrijk.be"
            android:path="/nieuws"
            android:scheme="http" />
    </intent-filter>
</activity>
```



public static final String CATEGORY_BROWSABLE

Added in API level 1

Activities that can be safely invoked from a browser must support this category. For example, if the user is viewing a web page or an e-mail and clicks on a link in the text, the Intent generated to execute that link will require the BROWSABLE category, so that only activities supporting this category will be considered as possible actions. By supporting this category, you are promising that there is nothing damaging (without user intervention) that can happen by invoking any matching Intent.

Constant Value: "android.intent.category.BROWSABLE"