

Week 8 - oefening 2

Doelstelling

Inlezen van JSON-data afkomstig van webservice.

Voorbereiding

Maak een nieuw Android Application Project aan met volgende instellingen.

Application Name:	Kot in Kortrijk
Package Name:	be.howest.nmct.studentenhuizenkortrijk
Target SDK:	21
Activity:	StudentenhuizenActivity (<i>layout: activity_studentenhuizen.xml</i>)
Fragments:	StudentenhuizenFragment (<i>layout: fragment_studentenhuizen.xml</i>) <i>Layout rij: row_studentenhuis.xml</i>
Launch icon:	Zie bijlage

Vooraf

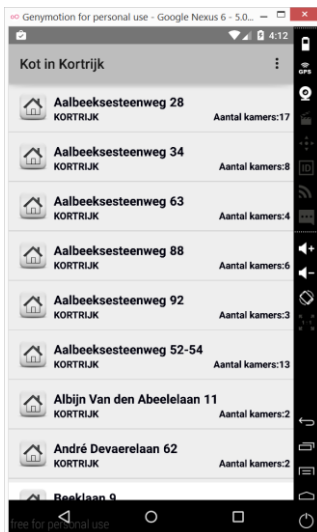
We maken gebruik van de 'Open data Kortrijk': langs deze weg stelt stad Kortrijk data beschikbaar om eigen applicaties uit te bouwen en er creatief mee aan de slag te gaan. Meer info vindt u op: <http://www.kortrijk.be/opendata>

Wij maken gebruik van data aangeleverd voor Studentenhuizen (koten) in Kortrijk, en kiezen hierbij voor het JSON-formaat: <http://data.kortrijk.be/studentenvoorzieningen/koten.json>

"XML en JSON zijn beide plain text formaten (dus niet gecompileerd) die data beschrijven. XML doet dit tussen tags en gebruikt attributen, JSON gebruikt hiervoor de JavaScript notatie (JavaScript Object Notation). Alle verdere info over JSON is te vinden op <http://json.org/>. Als je enkele voorbeelden bekijkt op <http://json.org/> example dan merk je dat je JSON minder karakters nodig heeft dat XML in data te versturen. Dit komt vooral omdat JSON niet telkens met een open en gesloten tag moet werken. Dit is ook de reden dat JSON in de loop der jaren populairder geworden is dan XML." (Bron: cursus Business Applications)

Weergave

De verschillende studentenhuizen in kortrijk worden in een eenvoudige listview weergegeven. Elk item bevat de adresgegevens alsook het aantal beschikbare kamers.



Werkwijze

De opbouw van deze app is sterk gelijkaardig als de oefening uit week 7. We vullen opnieuw een listview op m.b.v. een MatrixCursor en bijhorende SimpleCursorAdapter. Enkel het inladen van de data is verschillend.

Volgende stappen dienen ook hier doorlopen te worden:

- maak een package 'be.howest.nmct.studentenhuizenkortrijk.loader'
- maak een nieuwe klasse '**Contract**' in bovenstaand package waarin de verschillende kolommen uit de MatrixCursor opgesomd staan
- maak een nieuwe klasse '**StudentenhuizenLoader**' in bovenstaand package verantwoordelijk voor inladen data (zie verder)
- maak in de klasse StudentehuizenFragment een adaptarklasse '**StudentenHuizenAdapter**' die erft van SimpleCursorAdapter. Deze klasse staat in voor het visualiseren van de data afkomstig van de cursor. In de 'bindView'-methode gebruik je de views gedefinieerd uit de layoutfile 'row_studentenhuis.xml'. (Maak gebruik van een viewholder-klasse: meer info labo week6-oefening1).
- Zorg ervoor dat StudentehuizenFragment kan communiceren met de **LoaderManager**. Dit doe je door de klasse de interface 'LoaderManager.LoaderCallbacks' te laten implementeren. Werk de drie interface-methodes uit.
- Werk tenslotte de methode **onActivityCreated** in StudentehuizenFragment uit: hier wordt de adapter geïnitieerd en aan listview gekoppeld. Tenslotte wordt de loader geactiveerd via het commando `getLoaderManager().initLoader(...)`

StudentenhuizenLoader

Laat deze klasse erven van AsyncTaskLoader<Cursor>. Baseer je op het labo van week 6 en 7 om deze klasse op te bouwen.

Het grootste verschil met vorige labo's zit achter de methode 'loadInBackground'. Deze methode gaat de json-data binnenhalen en verwerken. We maken gebruik van de JsonReader-klasse. Bekijk even het voorbeeld op onderstaande link.

<http://developer.android.com/reference/android/util/JsonReader.html>

Bestudeer vooraf de data die moet verwerkt worden. We kiezen ervoor om de data manueel te overlopen. Een goede aanzet vindt u alvast hieronder:

```
public void loadCursor() {
    synchronized (lock) {
        if (mCursor != null) return;

        MatrixCursor cursor = new MatrixCursor(mColumnNames);
        InputStream input = null;
        JsonReader reader = null;

        try {
            input = new URL(url).openStream();
            reader = new JsonReader(new InputStreamReader(input, "UTF-8"));

            int id = 1;
            reader.beginArray();
            while (reader.hasNext()) {
                reader.beginObject();

                String adres = "";
                String huisnummer = "";
                String gemeente = "";
                int aantalKamers = 0;

                while (reader.hasNext()) {

                    String name = reader洗nextName();
                    if (name.equals("ADRES")) {
                        adres = reader.nextString();
                    } else if (name.equals("HUISNR")) {
                        //opgelet zowel numerieke waarden (nextInt) als string-waarden
                        //(nextString) komen voor!
                        //Voer controle uit via 'peek'-methode: Returns the type of the next
                        // token without consuming it
                        if (reader.peek().equals(JsonToken.NULL)) {
                            reader.skipValue();
                        } else if (reader.peek().equals(JsonToken.STRING)) {
                            //...
                        } else if (reader.peek().equals(JsonToken.NUMBER)) {
                            //...
                        }

                        // enz...

                    } else {
                        reader.skipValue();
                    }
                }
            }
        }
    }
}
```

```

        MatrixCursor.RowBuilder row = cursor.newRow();
        row.add(id);
        row.add(adres);
        row.add(huisnummer);
        row.add(gemeente);
        row.add(aantalKamers);
        id++;

        reader.endObject();
    }
    reader.endArray();

    mCursor = cursor;
} catch (IOException ex) {
    ex.printStackTrace();
} finally {
    try {
        reader.close();
    } catch (IOException e) {
    }
    try {
        input.close();
    } catch (IOException e) {
    }
}
}
}

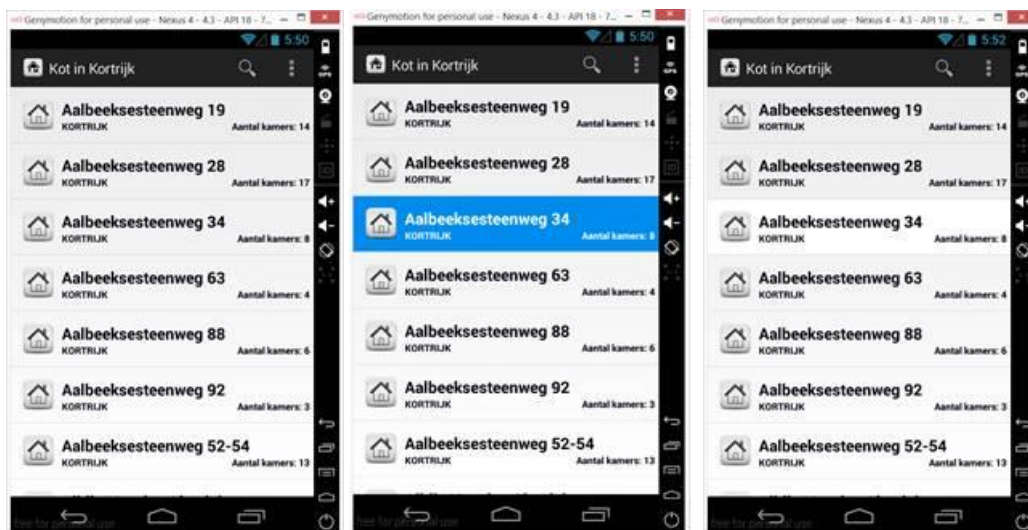
```

Vergeet ook de android-manifest file uit te breiden:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Style (extra)

Pas de stijl van de items in de listview aan zodat er een verschil is tussen de normale status, aangeklikte status en geselecteerde status.



Meer info is snel te vinden.

<http://javatechig.com/android/android-listview-tutorial>

<http://cyrilmottier.com/2011/08/08/listview-tips-tricks-3-create-fancy-listviews/>