

# Workshop SQLite

In deze tutorial gaan we door middel van SQLite een programma maken waarin gegevens ingevoerd kunnen worden, die vervolgens gewijzigd of verwijderd kunnen worden. Hiervoor gebruiken we het project dat meegeleverd is in het zip bestand. In dit project is al een edit formulier gemaakt met de functie die daar bij hoort. In de tutorial gaan we de main.class maken waarin de database connectie word gemaakt , de gegevens worden ingevoerd en gewijzigd doormiddel van het edit formulier en gegevens kunnen worden verwijderd door lang op een item in de lijst te klikken.



## onCreate

We beginnen met het maken van een functie die telkens moet worden uitgevoerd als de applicatie wordt geopend. Hierin roepen we eerst de template aan in het xml bestand overview.xml. Daarna maken we een verbinding met de database, hierbij wordt gebruikt gemaakt van de SQLiteOpenHelper. Vervolgens roepen we de fillData() functie aan die verder in deze tutorial gemaakt zal worden. Verder moet aan elke item een ContextMenu die wordt geopend wanneer er lang op een item geklikt wordt ook hiervan wordt de functie verder behandeld.

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.overview);  
  
    mDbHelper = new DbAdapter(this);  
    mDbHelper.open();  
  
    fillData();  
  
    registerForContextMenu(getListView());  
}
```

## fillData

Vervolgens maken we de functie filldata(). Deze functie wordt in andere functies opgeroepen om de ingevoerde/aangepaste in de database tabel te zetten.

```
private void fillData() {  
  
    Cursor c = mDbHelper.fetchAllItems();  
    startManagingCursor(c);  
  
    Log.i("MAIN", "Aantal rijen: "+c.getCount());  
  
    String[] from = new String[]{DbAdapter.T_NAME, DbAdapter.T_AGE};  
    int[] to = new int[]{R.id.text1, R.id.text2};  
  
    SimpleCursorAdapter items = new SimpleCursorAdapter(this, R.layout.row,  
c, from, to);  
    setListAdapter(items);  
}
```

### onCreateOptionsMenu

In deze functie gaan we nieuwe items kunnen toevoegen, dit doen we door gebruikt te maken van een option onderaan het scherm.

```
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    menu.add(0, INSERT_ID, 0, R.string.menu_insert);
    return true;
}
```

### onMenuItemSelected

Na het klikken van OptionMenu wordt de functie createItem wordt aangeroepen, deze functie wordt later in deze tutorial gemaakt.

```
@Override
public boolean onOptionsItemSelected(int featureId, MenuItem item) {
    switch(item.getItemId()) {
        case INSERT_ID:
            createItem();
            return true;
    }
    return super.onOptionsItemSelected(featureId, item);
}
```

### onCreateContextMenu

Om een item te kunnen verwijderen maken we een cotextmenu, dit betekent dat als men lang op een item klikt een actie uitgevoerd kan worden.

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.add(0, DELETE_ID, 0, R.string.menu_delete);
}
```

### onContextItemSelected

Wanneer er een item van het cotextmenu geselecteerd is kan men kiezen om een actie uit te voeren. In dit geval is dit alleen een functie om het geselecteerde item te verwijderen.

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    switch(item.getItemId()) {
        case DELETE_ID:
            AdapterContextMenuInfo info = (AdapterContextMenuInfo)
            item.getMenuInfo();
            mDbHelper.deleteItem(info.id);
            fillData();
            return true;
    }
    return super.onContextItemSelected(item);
}
```

### createItem

Deze functie werd aangeroepen in de functie onMenuItemSelected. De functie zorgt ervoor dat de EditItem class wordt aangeroepen, waarin men de gegevens van de nieuw te maken item ingevoerd kunnen worden

```
private void createItem() {  
    Intent i = new Intent(this, EditItem.class);  
    startActivityForResult(i, ACTIVITY_CREATE);  
}
```

### onListItemClick

De functie onListItemClick zorgt ervoor dat wanneer men op een item in de lijst klikt de pagina met het edit formuliertje wordt geopend. Dit gebeurt door de EditItem class aan te roepen wanneer er op het item is geklikt.

```
@Override  
protected void onListItemClick(ListView l, View v, int position, long id) {  
    super.onListItemClick(l, v, position, id);  
    Intent i = new Intent(this, EditItem.class);  
    i.putExtra(DbAdapter.T_ROWID, id);  
    startActivityForResult(i, ACTIVITY_EDIT);  
}
```

### onActivityResult

De laatste functie wordt altijd uitgevoerd wanneer er een activiteit plaatsvindt de data die is aangepast of is ingevoerd wordt dan in de database gevuld.

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent  
intent) {  
    super.onActivityResult(requestCode, resultCode, intent);  
    fillData();  
}
```

Als alle code goed in de main.class is ingevoerd zou het programma moeten werken.

Eventueel kan in de database gekeken worden welke gegevens er in staan. Dit kan door de volgende FireFox Plugin te downloaden <https://addons.mozilla.org/en-US/firefox/addon/5817>.