

## Fragmenten

### Klasse

Een fragment wordt afgeleid van de basis klasse "Fragment".

```

package com.android.cursustekst;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnDateChangeListener;
import android.widget.CalendarView;
import android.widget.ListView;
import android.widget.Toast;

public class KalenderFragment extends Fragment {
    private CalendarView cvkalender;
    private ListView lvkalender;
    private ArrayList<String> lvstrings = new ArrayList<String>();
    private ArrayAdapter<String> lvadapter;
    ArrayList<String> activiteitenarray;
    ArrayList<String> activiteitenids;
    ArrayAdapter<String> activiteitenadapter;
    SimpleDateFormat datumformaat;

    public KalenderFragment() {
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setHasOptionsMenu(true);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_kalender, container, false);
    }

    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        try {
            cvkalender = (CalendarView) getActivity().findViewById(R.id.calendarView1);
            cvkalender.setOnDateChangeListener(datechange);
            lvkalender = (ListView) getActivity().findViewById(R.id.lv_kalender);
            datumformaat = new SimpleDateFormat("yyyy-MM-dd");
            String dat = datumformaat.format(Calendar.getInstance().getTime());
            toon_activiteiten(dat);
        }
    }
}

```

```
    } catch (Exception e) {
        Toast.makeText(getActivity(), e.toString(), Toast.LENGTH_SHORT).show();
        Log.v("Kalender fragment onactivitycreated", e.toString());
    }

    OnDateChangeListener datechange = new OnDateChangeListener() {
        public void onSelectedDayChange(CalendarView view, int year, int month,
            int dayOfMonth) {
            String datum = datumformaat.format(view.getDate());
            toon_activiteiten(datum);
        }
    };
}
```

### Afgeleide klassen

Net zoals bij een activiteit kan je een fragment afleiden van een aantal basis fragment klassen zoals bvb. een ListFragment, een DialogFragment of een PreferenceFragment.

### Lifecycle functies

Je kunt meer informatie over fragmenten nalezen op de website van Android: <http://developer.android.com/guide/components/fragments.html>

Een fragment wordt gebruikt om een deel van een gebruikersinterface van een activiteit op te vullen met informatie. Een fragment wordt ook beschreven als een sub-activiteit of een module van een activiteit.

Fragmenten kunnen toegevoegd worden aan een activiteit of ervan verwijderd worden. Daarnaast kan je een fragment verbergen of zichtbaar maken.

De levenscyclus van een fragment is verweven in de levenscyclus van de activiteit waartoe het fragment behoort.

Meestal worden de volgende functies van een fragment gebruikt: onCreate(), onCreateView(), onActivityCreated() en onPause().

In de onCreateView() functie wordt de koppeling gelegd met de layout.

In de onActivityCreated() functie wordt de eigenlijke verwerking van het fragment geprogrammeerd. Als je een layout onderdeel wil ophalen moet je dat doen door te verwijzen naar de activiteit met de functie getActivity(). Een verwijzing naar een ListView in een fragment kan er dus als volgt uitzien:

```
lvkalender = (ListView) getActivity().findViewById(R.id.lv_kalender);
```

### Layout koppeling

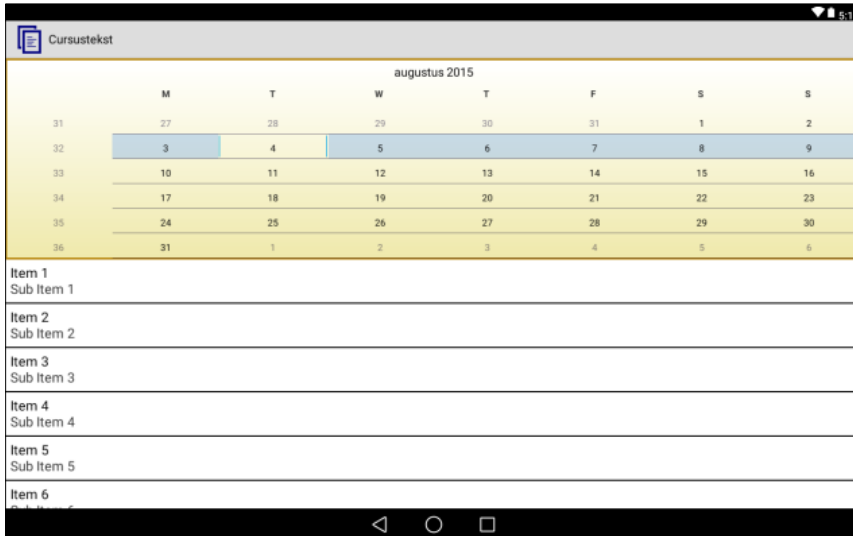
In de onCreateView wordt met de inflate() functie een layout opgehaald. In het voorbeeld zie je een "design" en een "text" weergave. Je kan ook hier gebruik maken van Views en Viewgroups die je met Java code definieert.

**@Override**

```

public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_kalender, container, false);
}

```



```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/kleuren_kalender"
        android:orientation="vertical" >
        <CalendarView
            android:id="@+id/calendarView1"
            android:layout_width="match_parent"
            android:layout_height="300dp"
            android:firstDayOfWeek="2" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >
        <ListView
            android:id="@+id/lv_kalender"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:divider="#000000"
            android:dividerHeight="2px" >
        </ListView>
    </LinearLayout>
</LinearLayout>

```

## Menu koppeling

Een fragment kan een specifieke menu hebben binnen de activiteit. Dit moet je aanduiden door in de onCreate() functie de onOptionsItemSelected() functie aan te roepen.

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setHasOptionsMenu(true);
}

```

In de `onOptionsItemSelected()` functie kan je de menu-items van de activiteit opnemen en geen actie ondernemen met het statement "return false;". Als die menu-ingangen worden aangestipt dan wordt de menu-ingang van de activiteit uitgevoerd.

```

@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    inflater.inflate(R.menu.kalender_menu, menu);
    super.onCreateOptionsMenu(menu, inflater);
}
@SuppressWarnings("finally")
public boolean onOptionsItemSelected(MenuItem mitem) {
    try {
        Fragment fragment = null;
        switch (mitem.getItemId()) {
            case R.id.menu_settings:
                return false;
            case R.id.menu_beheer_activiteiten:
                fragment = new ActiviteitenFragment();
                if (fragment != null) {
                    android.support.v4.app.FragmentManager fragmentManager = getFragmentManager();
                    FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
                    fragmentTransaction.replace(R.id.content_frame, fragment);
                    fragmentTransaction.commit();
                }
                return true;
            case R.id.menu_endapp:
                return false;
            default:
                return super.onOptionsItemSelected(mitem);
        }
    } catch (Exception e) {
        Log.v("Kalender fragment onOptionsItemSelected", e.toString());
    } finally {
        return true;
    }
}

```

```

<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:id="@+id/menu_beheer_activiteiten"
        android:icon="@drawable/ic_action_new"
        android:title="@string/menu_beheer_activiteiten"
        android:orderInCategory="300"
        android:showAsAction="never" />
</menu>

```

## Fragment manager

Je kan fragmenten toevoegen aan een activiteit layout met de tag "<Fragment>".

Je kan ook fragmenten met Java code toevoegen.

Het beheer van de fragmenten gebeurt door een "FragmentManager" object en een "FragmentTransaction" object. De fragmentManager start een fragmentTransaction met de functie "beginTransaction()". Met de functie "replace()" van het

fragmentTransaction object kan een fragment in het "content frame" worden geplaatst. De functie "commit()" voert de wijzigingen effectief door.

```
fragment = new ActiviteitenFragment();
if (fragment != null) {
    android.support.v4.app.FragmentManager fragmentManager = getFragmentManager();
    FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
    fragmentTransaction.replace(R.id.content_frame, fragment);
    fragmentTransaction.commit();
}
```

In de activiteit layout ("activity\_main.xml") wordt de FrameLayout gedefinieerd.

```
<FrameLayout
    android:id="@+id/content_frame"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Je kan in het "fragmentTransaction" object ook de functies "add()" en "remove()" gebruiken.

Daarnaast kan je de transactie toevoegen aan de "backstack", een interne Android lijst van opgestarte activiteiten en fragmenten. Als je op jouw tablet of smartphone de "back" knop indrukt dan wordt uit de "backstack" de vorige activiteit of het vorige fragment teruggehaald en getoond.

De transactie wordt toegevoegd met "addToBackStack()" functie. Daarna kan je de "commit()" functie laten uitvoeren om alle wijzigingen te laten uitvoeren.