

# 1 Fragments

## 1.1 Was sind Fragments?

Ein Fragment stellt einen Teil der Benutzeroberfläche einer Activity zur Verfügung, dabei kann man mehrere Fragments in einer Activity verwenden und diese zur Laufzeit austauschen. Da Fragments in mehreren Activities wiederverwendet werden können, müssen Ansichten wie Detailviews oder Listen nur einmal programmiert werden und können überall eingesetzt werden. Fragments werden ab Android 3.0 (API level 11) zur Verfügung gestellt.

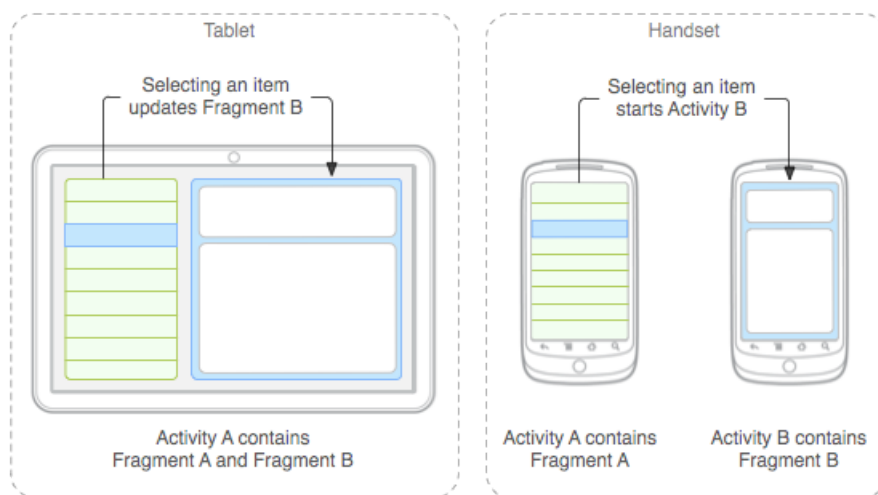


Figure 1: Beispiel man kann mithilfe von Fragments Views erstellen, die sowohl auf einem Tablet als auch auf einem Handy ein optimales Benutzerinterface anbieten

## 1.2 Der Lifecycle

Ein Fragment ist immer eingebunden in eine Activity und ist direkt vom Lifecycle der übergeordneten Activity abhängig. Wird die übergeordnete Activity pausiert oder zerstört werden auch alle untergeordneten Fragments pausiert oder zerstört. Das Aufbauen der Benutzeransicht erfolgt in der onCreateView() Methode.

In einem Fragment:

```
1 @Override
2 public View onCreateView(LayoutInflater inflater , ViewGroup
   container , Bundle savedInstanceState) {
3     super.onCreate(savedInstanceState);
4     return inflater.inflate(R.layout.example_fragment ,
   container , false);
5 }
```

## 1.3 Fragment Transactions

Mittels Transaktionen lassen sich Fragments hinzufügen, entfernen oder ersetzen. Es werden mehrere dieser Aktionen hintereinander abgesetzt und zusammen nach einem commit() ausgeführt. Ein Fragment kann mittels addToBackStack() auch zum BackStack hinzugefügt werden um dadurch, ähnlich wie bei Activities, Navigation mit dem BackButton zu ermöglichen. Dabei ist zu beachten, dass alle Aktionen vor einem commit() gemeinsam auf den BackStack gelegt werden und bei drücken des BackButtons alle gemeinsam aufgehoben werden. Wird addToBackStack() nicht aufgerufen, wird ein Fragment beim Schließen oder beim Wechseln auf ein anderes Fragment zerstört und kann nicht mehr aufgerufen werden.

```
1     fragmentManager fragmentManager = getFragmentManager();
2     FragmentTransaction fragmentTransaction =
   fragmentManager.beginTransaction();
3     fragmentTransaction.addToBackStack(null);
4     fragmentTransaction.replace(R.id.fraPlace , fragment);
5     fragmentTransaction.commit();
```

## 1.4 Verwendung von Fragments

In dieser Arbeit werden Fragments verwendet, um die Benutzeransichten, ausgenommen des NavigationDrawers, anzuzeigen. Dabei wird ein FrameLayout im Layoutfile der MainActivity durch ein Fragment mittels der `displayView()` Methode ersetzt. Navigation durch diese Fragments wird mittels den Buttons im MFragment oder dem NavigationDrawer ermöglicht.

MainActivity.java und MFragment.java:

```
1 private void displayView(Fragment fragment) {  
2     FragmentManager fragmentManager = getFragmentManager();  
3     FragmentTransaction fragmentTransaction =  
4         fragmentManager.beginTransaction();  
5     fragmentTransaction.addToBackStack(null);  
6     fragmentTransaction.replace(R.id.fraPlace, fragment);  
7     fragmentTransaction.commit();  
8 }
```

activity\_main.xml:

```
1 <FrameLayout  
2     android:id="@+id/fraPlace"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent" />
```

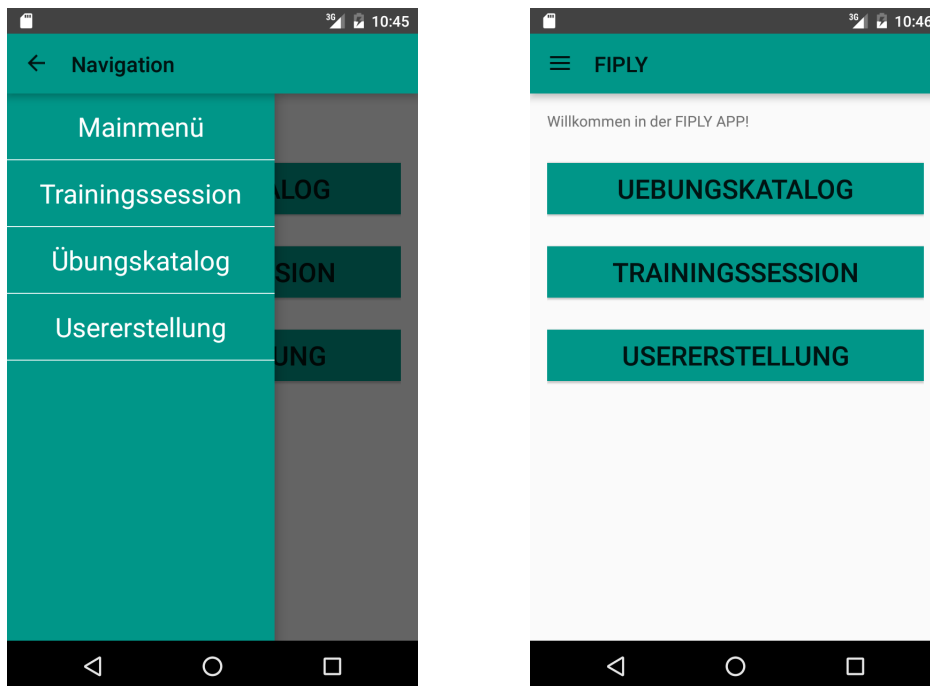


Figure 2: Bild des NavigationDrawers und des FMains

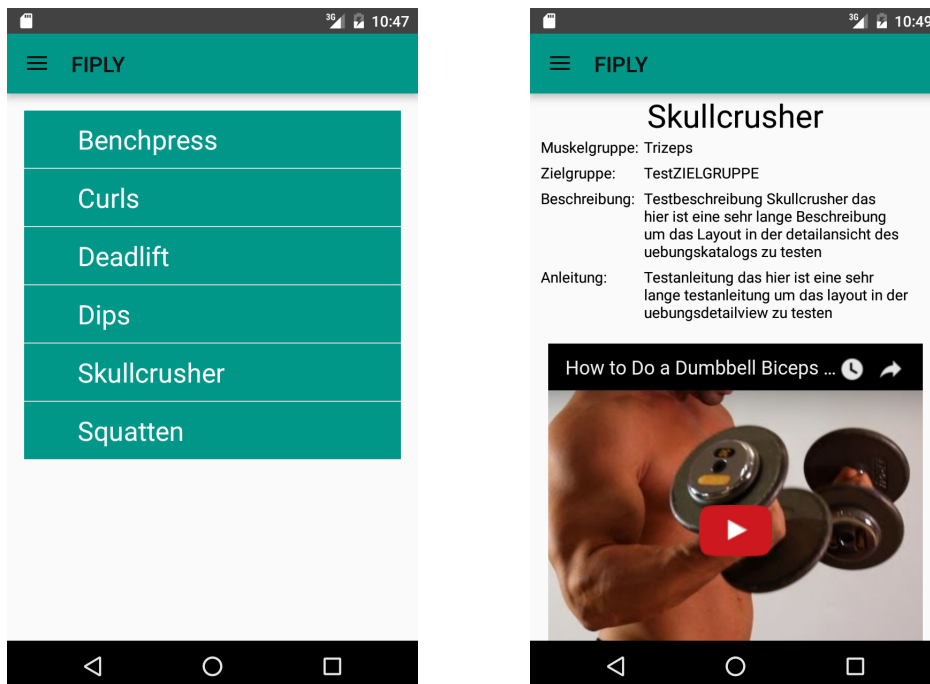


Figure 3: Bei Klicken eines Elements in der ListView wird die zugehörige DetailView aufgerufen