#Fragment

###Hvad er et Fragment?

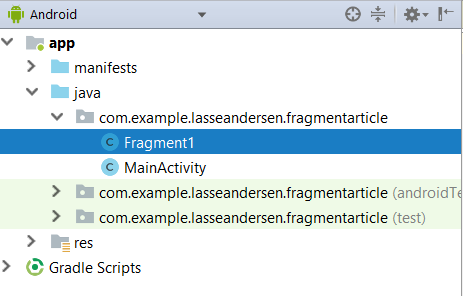
Et Fragment er en slags let Activity. Du kan kombinere flere fragments til én Activity, Dvs. at du kan genbruge til hvert fragment, samtidig med at du kan vise forskellige ting på de 2 fragments. Et Fragment skal altid kobles på en Activity, og hvis man sletter Aktiviteten, så fjerner man også Fragments. Hvis man pauser Aktiviteten så pauser ligeledes alle Fragments. Fordelen ved at bruge Fragments er at det er med til at give et mere dynamisk og fleksibelt Bruger Interface med genanvendeligt kode, som for eksempel en Navigation bar.

I denne artikel vil jeg fortælle om hvordan kan benytte sig af Fragments og vise en nem måde at navigere imellem de forskellige fragments. Fragments er knyttet til en Activity og i stedet for at sende dig til en anden Activity når du skal videre på appen, så er det 2 forskellige fragments du navigerer rundt på. Jeg har også lavet en Second Activity, for at vise en Activity som også kører uafhængigt af Fragments.

###Setup af Fragment:

En Fragment bliver tilføjet til dit projekt ligesom en ny Activity gør.

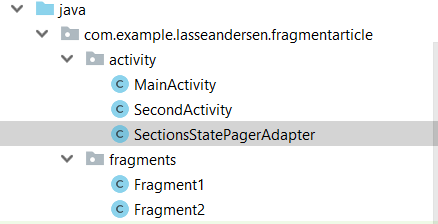
Efter du har lavet et nyt projekt, skal du være i mappen, som vist på billedet, altså din app mappe



Derefter højreklik og vælg **New -> Fragment -> Fragment(Blank)**

Når du laver et nyt Fragment får du også en XML-fil. XML-filerne ligger under mappen **Res -> Layout**, og hedder fragment\_fragment1.xml. Dine XML-filer er der du designer hvordan appen skal se ud, hvorimod Java filerne er til alt logikken og mekanismen til hvordan appen skal fungere.

Det er en god idé at lave en mappe du kalder Fragment og en mappe til Activities, så det er opdelt, og der er styr på det.

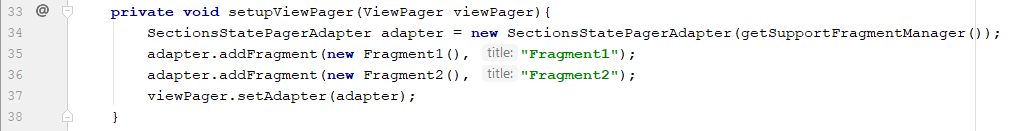


Som det ses på billedet, har jeg lavet 2 Fragments og 2 Activities også en Adapter som indeholder en liste af de Fragments man har.

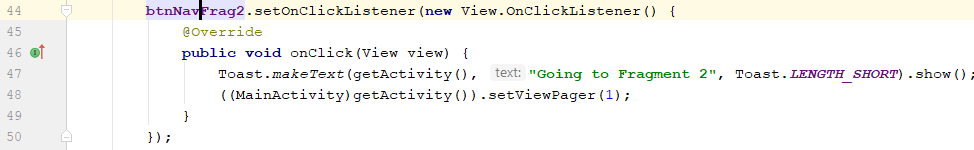
Med Adapteren kan man swipe rundt imellem Fragments, og det virker meget dynamisk og effektivt.

Min MainActivity inderholder har en reference til min SectionStateAdapter som er en klasse hvor der er metoder til at tilføje de fragments, tælle fragments og få fat i ét bestemt fragments.

I min MainActivity tilføjer jeg så Fragments til den adapter lister som kommer inde fra SectionStateAdapter klassen, som tager 2 parametre – En Fragment og en String.



I mine Fragments klasser har jeg lavet nogle Buttons, hvor jeg har sat en ”OnClickListener” på, så når der bliver klikket på knappen, laver den en ny intent og sender dig ind på en anden side, alt afhængigt af om det er en Activity eller Fragment.

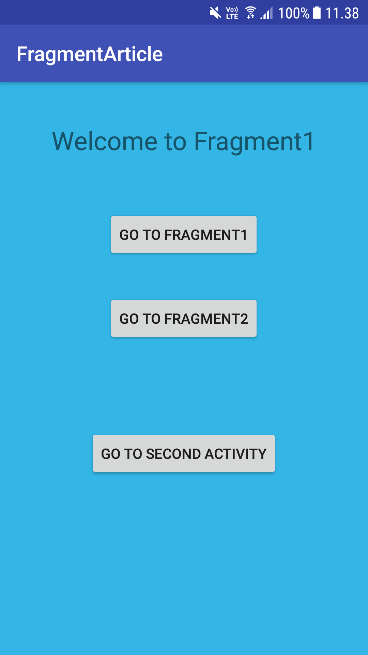
Samtidig med at der bliver trykket på knappen, laver jeg en Toast, så man er sikker på hvad der sker, Dvs. at der kommer tekst frem når man trykker.

### Hvad gjorde du og hvorfor?

Til at starte med vidste jeg ikke helt hvordan jeg ville vise og beskrive Fragments, men så kom jeg i tanke om at vi i vores Eksamensprojekt havde brugt nogenlunde den samme løsning, med adapter og Fragments. Løsningen er en god og simpel løsning som er til at forstå, og jeg synes at det er nemmere at forstå end de generelle løsninger med getFragmentManager.getTransaction osv.

Jeg havde lidt opstartsproblemer, men langt om længe kom jeg videre med emnet. Grunden til jeg havde problemer i starten var at jeg blev i tvivl om eksemplet ville være for lille og for lidt kode, men så tog jeg en beslutning om at det var den rigtige måde at gøre det på.

### Hvordan gik det?

Jeg synes overordnet set det gik godt, det endte med at blive et godt resultat med projektet, og jeg håber at en eventuel person ville kunne forstå min løsning.

### Konklusion

Jeg kan konkludere ud fra min løsning at Fragments er klart at foretrække hvis man vil have en App som virker dynamisk og fleksibelt.

Fragments er det mest udbredte i forhold til visning af sine Layouts. Der er selvfølgelig mange måder at gribe det an på, men jeg kan blot konkludere at min måde er en god løsning. Jeg har valgt at gøre det på en nem måde, men hvis man vil have knapperne skal designes og ligne en navigation bar i bunden eller toppen af skærmen kan man også gøre det.