



UTILISATION DES FRAGMENTS AVEC ANDROID

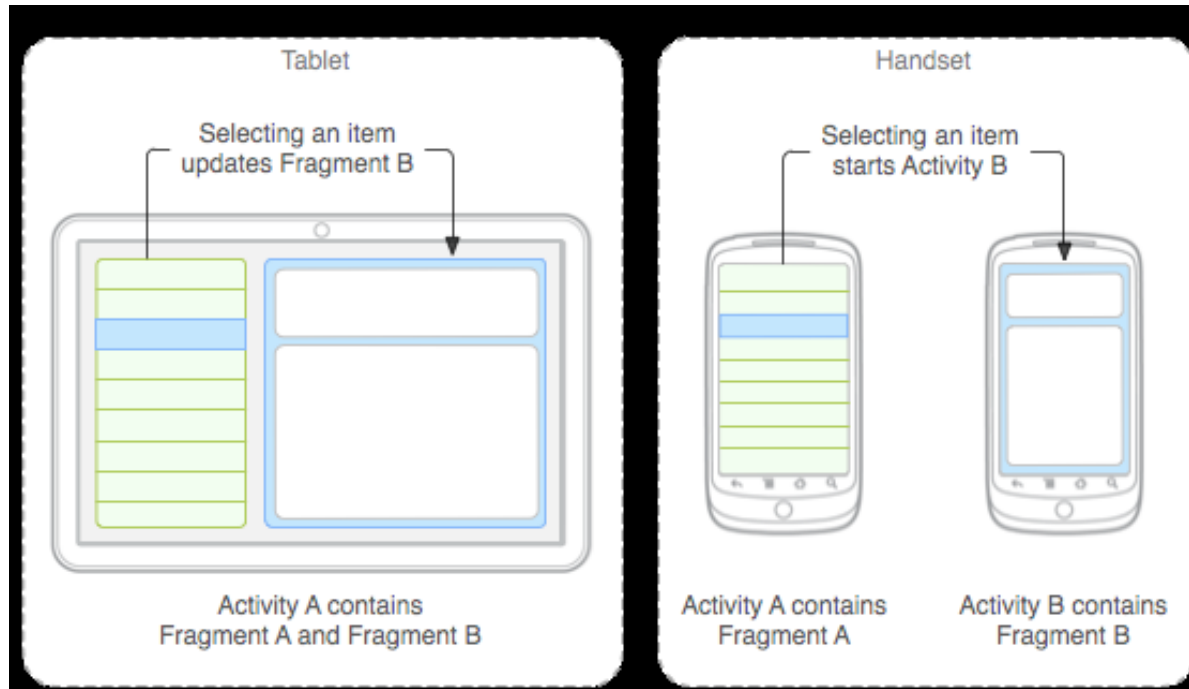


VS

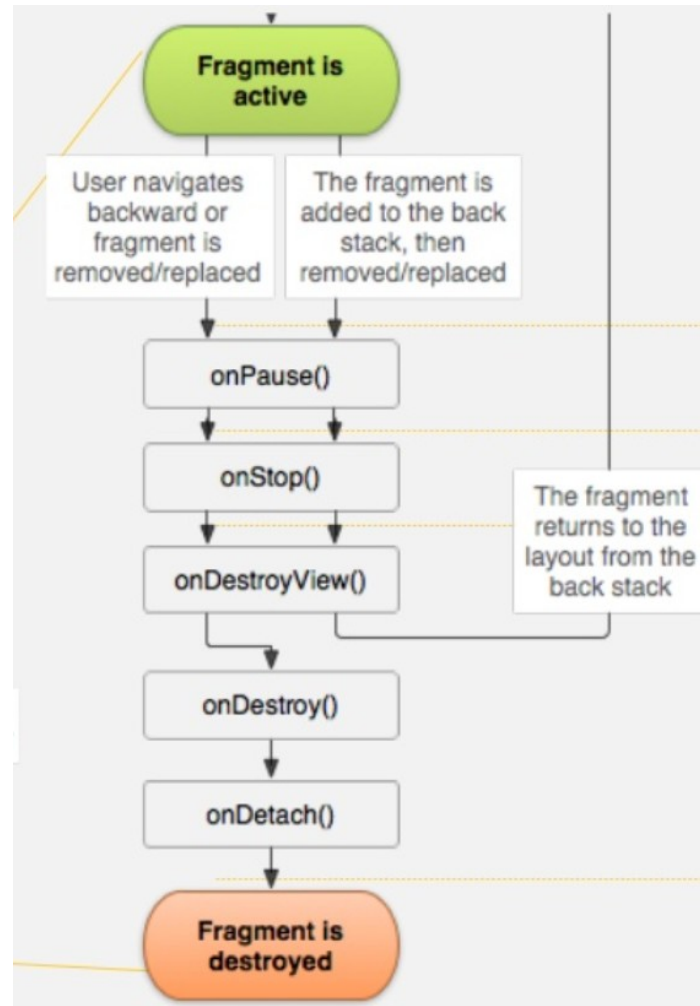
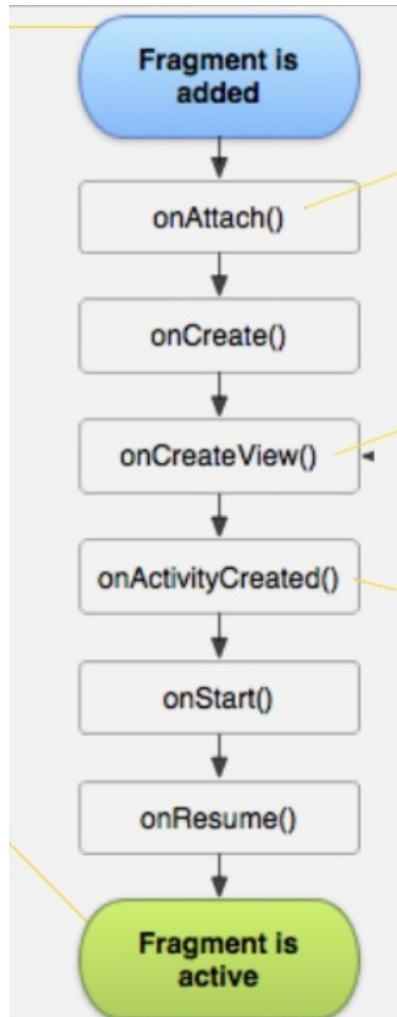


NOTION DE FRAGMENTS

- ❑ Pour ne pas surcharger une activité
- ❑ Pour s'adapter à plusieurs type de périphériques
- ❑ Pour être réutilisable facilement

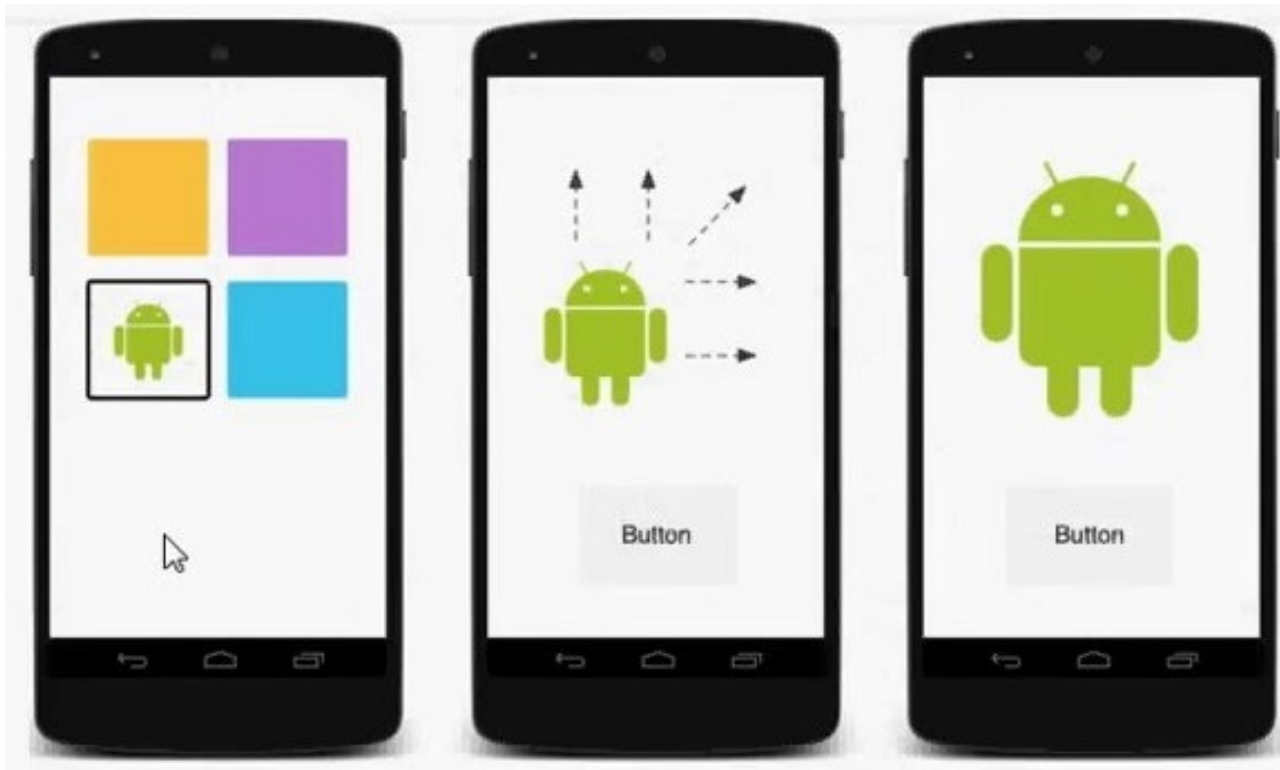


CYCLE DE VIE DES FRAGMENTS



COMMENT CRÉER UN FRAGMENT ?

- ❑ Statiquement
- ❑ Dynamiquement



CRÉER UN STATIQUEMENT

❑ De manière statique

- ◆ permet d'ajouter un fragment directement *depuis le layout* d'une activité
- ◆ Il ne pourra plus être retiré ou modifié dynamiquement



fragment_main.xml



activity_main.xml

```
<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">
```

```
    <Button
```

```
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="go"/>
```

```
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<fragment
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="MainFragment"/>
```

CRÉER UN FRAGMENT STATIQUEMENT

❑ De manière dynamique

- ◆ permet d'ajouter un fragment directement *depuis l'activité parente*
- ◆ Il pourra être retiré ou modifié dynamiquement



MainFragment.java

```
class MainFragment extends Fragment {  
    public MainFragment() { } //constructeur (même vide)  
  
    @Override  
    public View onCreateView( LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState ) {  
        View rootView = inflater.inflate(R.layout.fragment_main, container, false);  
        //do something  
        return rootView;  
    }  
}
```



CRÉER UN FRAGMENT STATIQUEMENT

❑ De manière dynamique

- ◆ permet d'ajouter un fragment directement *depuis* l'activité parente
- ◆ Il pourra être retiré ou modifié dynamiquement



MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Le système Android démarre de lui-même le Fragment défini dans activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<fragment
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="MainFragment"/>
```

- ❑ Une classe qui hérite de Fragment
- ❑ Un constructeur par défaut (même vide !)
- ❑ Surcharger la méthode onCreateView() qui retourne une vue (appelée rootView)

CRÉER UN FRAGMENT DYNAMIQUEMENT



Fragment_main.xml

```
<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" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SHOW ME DETAILS ! " />
</LinearLayout>
```



Rien ne change de ce côté !



CRÉER UN FRAGMENT DYNAMIQUEMENT

Pour qu'une activité puisse charger un fragment, il faut simplement

- ❑ Créer un Fragment `new ExampleFragment()`
- ❑ Démarrer une transaction `getFragmentManager().beginTransaction()`
- ❑ Demander à la transaction de remplacer le container de la vue par le fragment

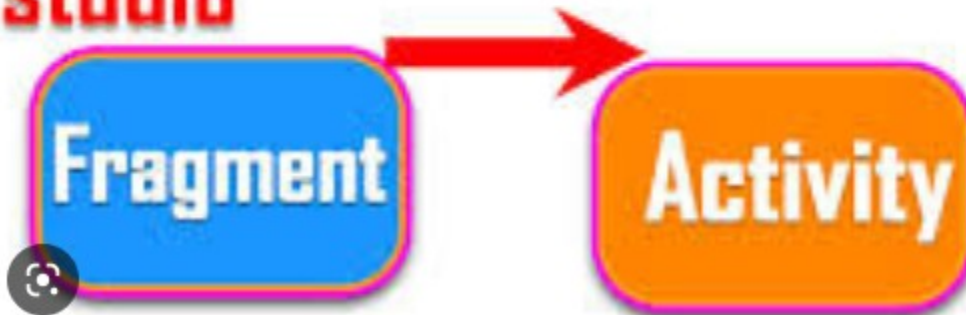
```
Fragment newFragment= new ExampleFragment();  
FragmentTransaction transaction = getFragmentManager().beginTransaction();  
transaction.replace(R.id.fragment_container, newFragment);  
transaction.commit();
```

- 1- On crée une instance (cette fois, ce n'est pas le système Android qui le fait)
- 2- on crée **une transaction**

COMMUNICATION ENTRE FRAGMENT ET ACTIVITE

Send data from
fragment to activity in android
studio

বাংলায়



1 364 x 76



COMMUNICATION ENTRE FRAGMENT ET ACTIVITE

MainActivity contient un fragment

Ce fragment contient un bouton "SHOW ME DETAILS"



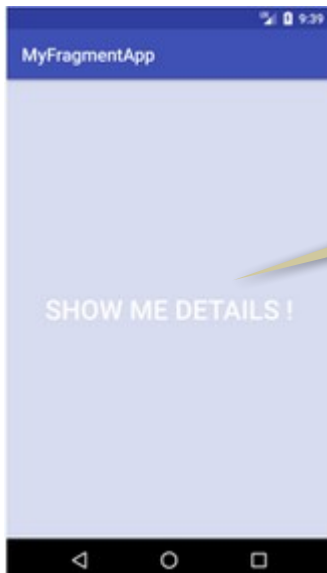
On tient à laisser la responsabilité à l'activité *MainActivity* de gérer le lancement de cette nouvelle activité.

Le rôle d'un fragment est de rendre des comptes à l'activité qui le porte

COMMUNICATION ENTRE FRAGMENT ET ACTIVITE

MainActivity contient un fragment

Ce fragment contient un bouton "SHOW ME DETAILS"



click □ ouvrir une nouvelle activité
contenant également un fragment.

**laisser la responsabilité à l'activité *MainActivity* de
gérer le lancement de cette nouvelle activité.**

Mais comment faire sachant que le
bouton "SHOW ME DETAILS" (et par
conséquent son listener) se trouveront
dans MainFragment ?



Mais comment faire sachant que le bouton "SHOW ME DETAILS" (et par conséquent son listener) se trouveront dans MainFragment ?



MainFragment. java

```
public class MainFragment extends Fragment implements View.OnClickListener {  
    private OnButtonClickedListener mCallback;  
    public interface OnButtonClickedListener { void onButtonClicked(View view); }  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
        View result=inflater.inflate(R.layout.fragment_main, container, false);  
        result.findViewById(R.id.fragment_main_button).setOnClickListener(this);  
        return result;  
    }  
  
    @Override  
    public void onClick(View v) {  
        //todo  
    }  
  
    @Override  
    public void onClick(View v) { mCallback.onButtonClicked(v); }  
  
    @Override  
    public void onAttach(Context context) {  
        super.onAttach(context);  
        this.createCallbackToParentActivity();  
    }  
  
    private void createCallbackToParentActivity(){  
        try {  
            mCallback = (OnButtonClickedListener) getActivity();  
        } catch (ClassCastException e) {  
            throw new ClassCastException(e.toString()  
                + " must implement OnButtonClickedListener");  
        }  
    }  
}
```