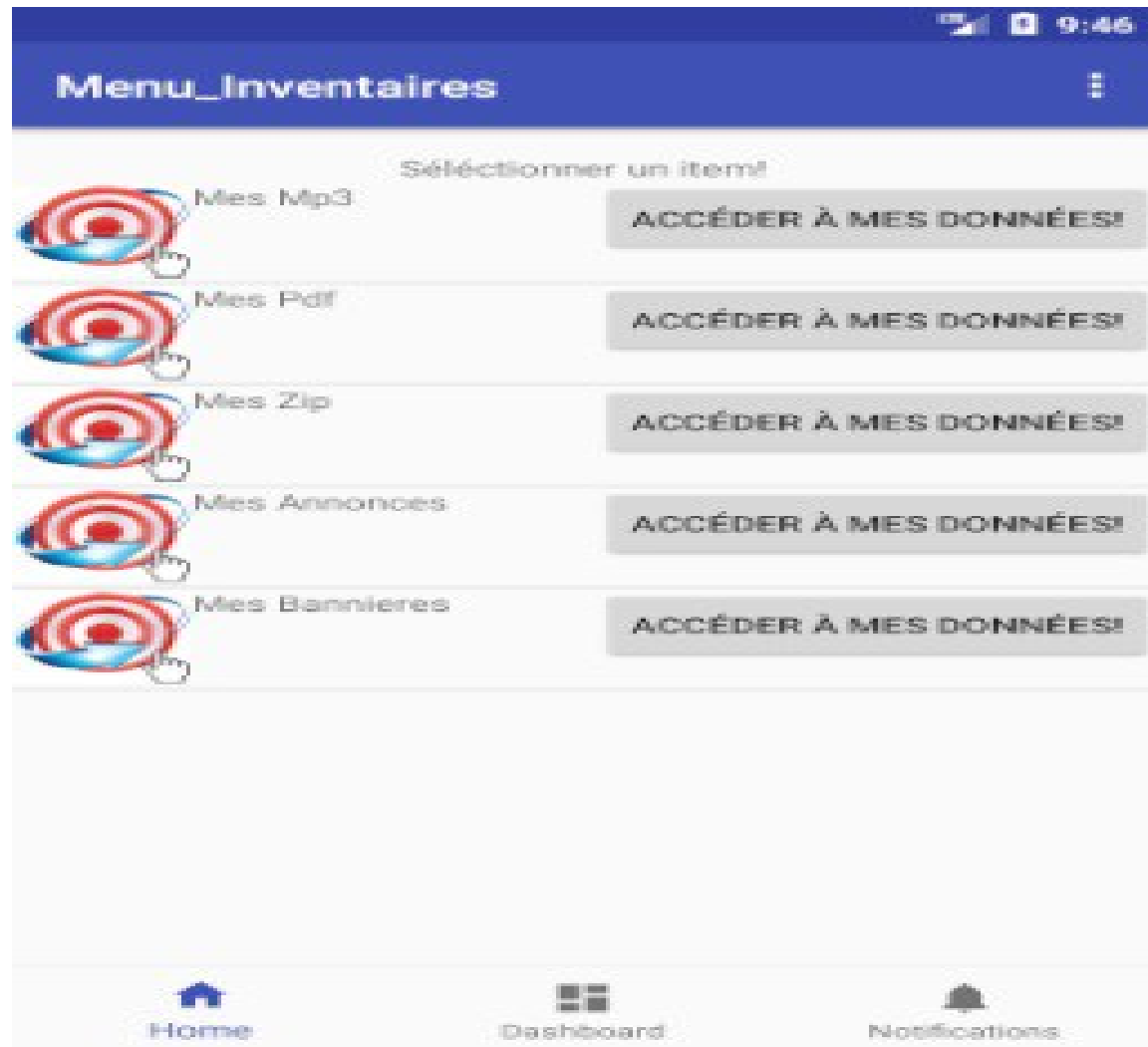
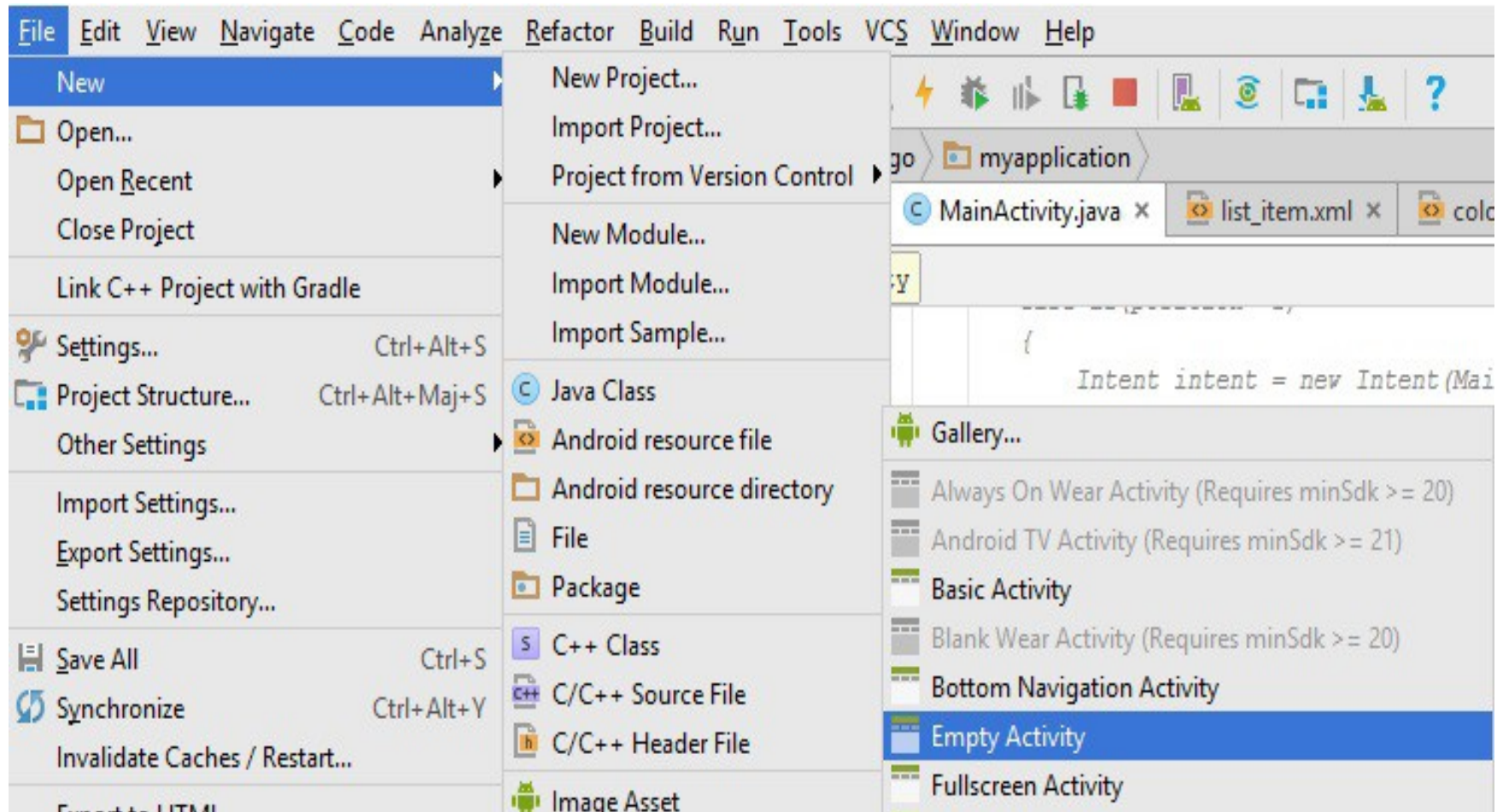


# Tutoriel créer une liste adaptable Android Studio!



# Tutoriel créer une liste adaptable Android Studio!

- ▶ Première étape création d'une activité vide!



# Tutoriel créer une liste adaptable Android Studio!

- ▶ Deuxième étape ajout de la classe MyListAdapter dans la Class de votre Activité principale en dessous du onCreate(Bundle savedInstanceState)

```
private class MyListAdapter extends ArrayAdapter<String> {
    private int layout;
    private List<String> mObjects;
    private MyListAdapter(Context context, int resource, List<String> objects) {
        super(context, resource, objects);
        mObjects = objects;
        layout = resource;
    }

    @Override
    public View getView(final int position, View convertView, ViewGroup parent) {
        MainActivity.ViewHolder mainViewHolder = null;
        if (convertView == null) {
            LayoutInflater inflater = LayoutInflater.from(getContext());
            convertView = inflater.inflate(layout, parent, false);
            MainActivity.ViewHolder viewHolder = new MainActivity.ViewHolder();

            viewHolder.button = (Button) convertView.findViewById(R.id.list_item_button);
            viewHolder.thumbnail = (ImageView) convertView.findViewById(R.id.list_item_img);
            viewHolder.title = (TextView) convertView.findViewById(R.id.list_item_text);

            convertView.setTag(viewHolder);
        }
        mainViewHolder = (MainActivity.ViewHolder) convertView.getTag();
        mainViewHolder.button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getContext(), "Button was clicked for list item " + position, Toast.LENGTH_SHORT).show();
            }
        });
        mainViewHolder.title.setText(getItem(position));

        return convertView;
    }
}
```

# Tutoriel créer une liste adaptable Android Studio!

- ▶ Troisième étape ajout de la class ViewHolder qui est la vue que l'on souhaite afficher n fois dans la ListView!

```
public class ViewHolder {  
    ImageView thumbnail;  
    TextView title;  
    Button button;  
}
```

- ▶ Quatrième étape déclaration d'une ArrayList privé en tant que champs de l'activité principale!

```
private ArrayList<String> data = new ArrayList<String>();
```

- ▶ Cinquième étape ajout de la ListView au Layout principale!

```
<ListView  
    android:id="@+id/listview"  
    android:layout_width="368dp"  
    android:layout_height="523dp"  
    tools:layout_editor_absoluteY="0dp"  
    tools:layout_editor_absoluteX="8dp"></ListView>
```

# Tutoriel créer une liste adaptable Android Studio!

- ▶ Sixième étape ajout du layout list\_item qui sera le layout affiché n fois dans la ListView!

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

    <ImageView
        android:layout_width="65dp"
        android:layout_height="65dp"
        android:id="@+id/list_item_img"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:src="@drawable/pub_cible"
        android:layout_alignParentLeft="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is row number 15"
        android:textColor="@color/colorPrimary"
        android:textSize="10sp"
        android:textStyle="bold|italic"
        android:id="@+id/list_item_text"
        android:layout_toEndOf="@+id/list_item_img"
        android:layout_toRightOf="@+id/list_item_img"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Visiter"
        android:layout_alignParentRight="true"
        android:id="@+id/list_item_button"/>

</RelativeLayout>
```

# Tutoriel créer une liste adaptable Android Studio!

- ▶ Septième étape ajout des données dans la ArrayList puis création d'un objet ListAdapter avec en paramètre le contexte de l'activité, le layout à répéter n fois(n=Le nombre d'item dans la ArrayList) et la ArrayList!

```
data.add("Partager!");  
data.add("Télécharger!");  
data.add("Mes Zip");  
data.add("Mes Annonces");  
data.add("Mes Bannieres");
```

```
ListView lv = (ListView) findViewById(R.id.listview);  
lv.setAdapter(new MainActivity.MyListAdapter(this, R.layout.list_item,  
data));
```

```
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int  
        position, long id) {  
        Toast.makeText(MainActivity.this, "List item was clicked at " +  
            position, Toast.LENGTH_SHORT).show();  
    }  
});
```

# Tutoriel créer une liste adaptable Android Studio!

- ▶ Huitième et dernière étape je m'abonne à la chaîne You Tube Vertin Go Website pour être constamment informer des derniers tutoriels techniques!

