

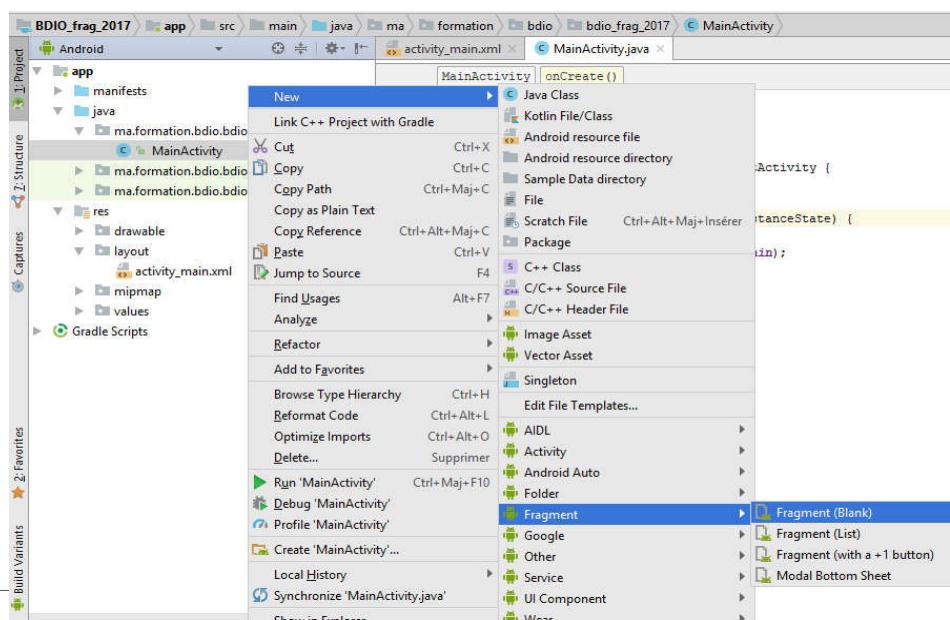
## Qu'est ce qu'un fragment ?

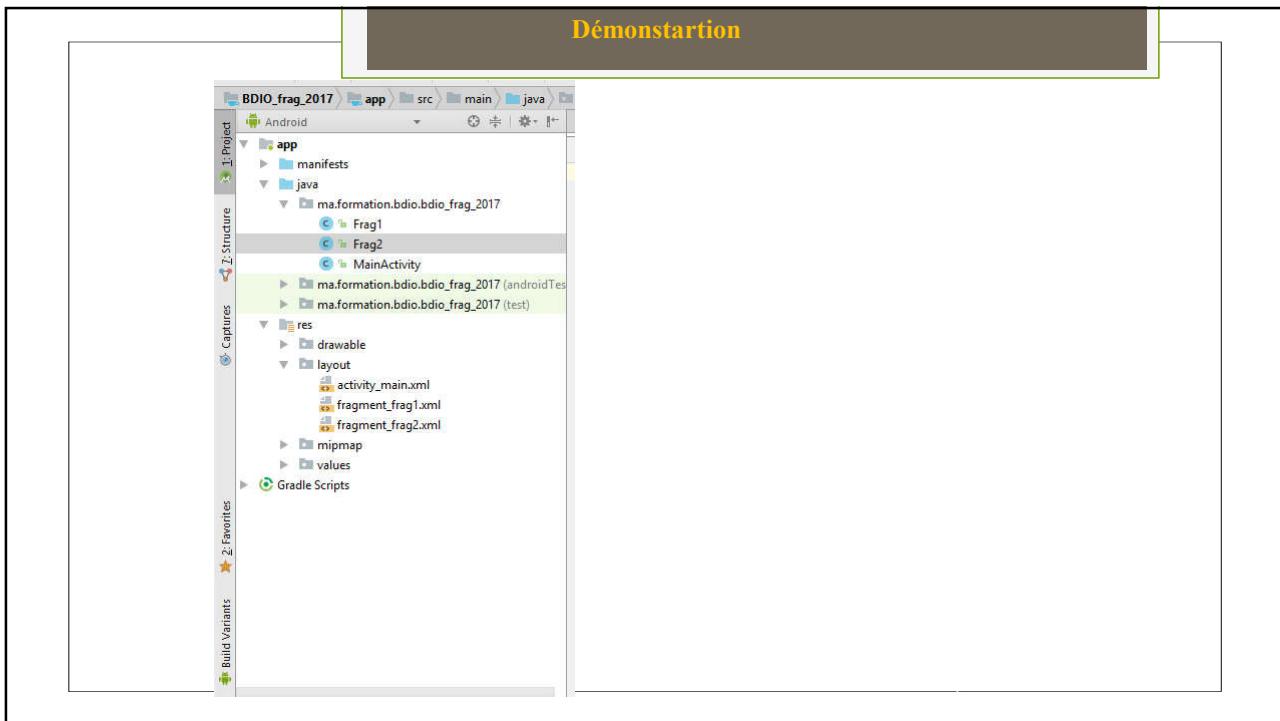
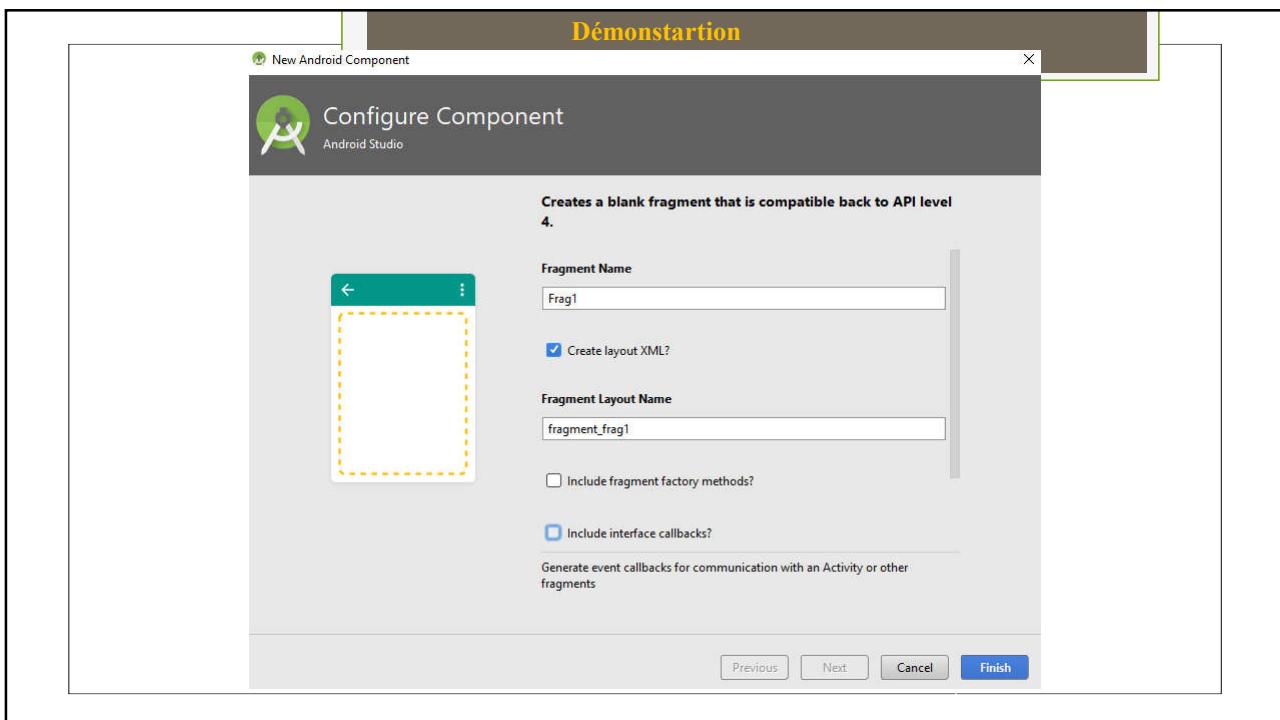
un Fragment comme une “mini-Activity”

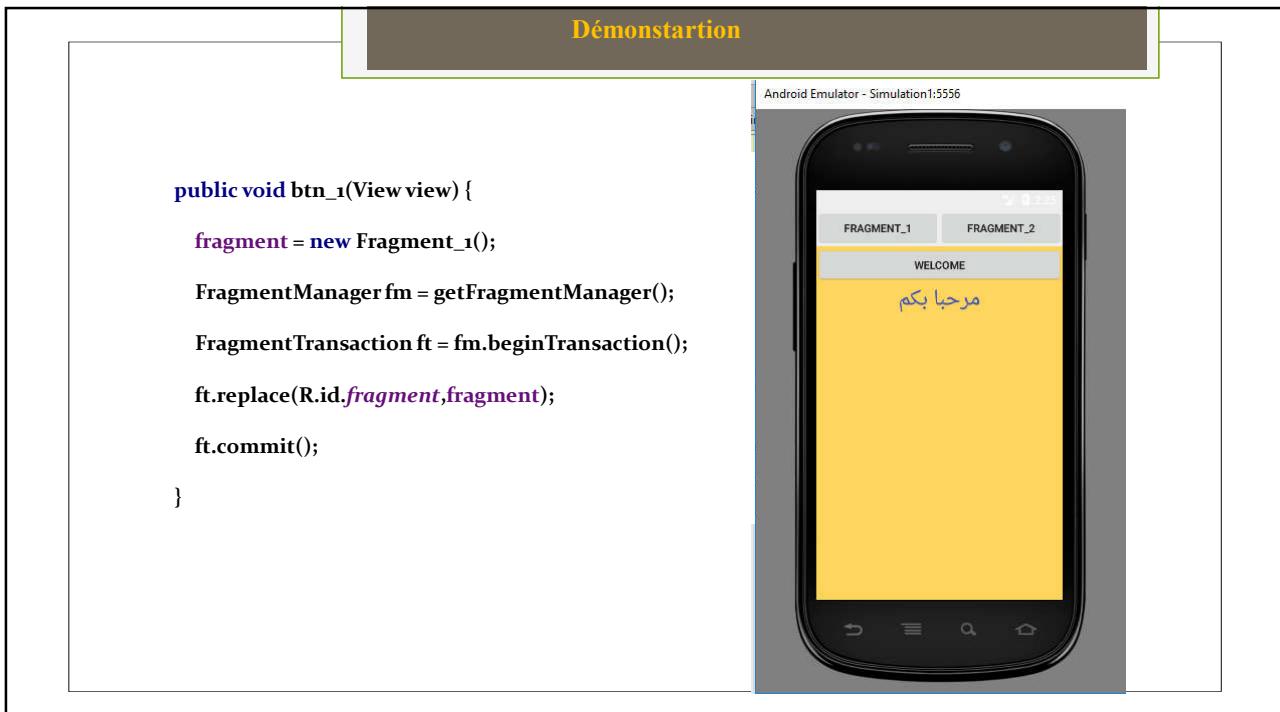
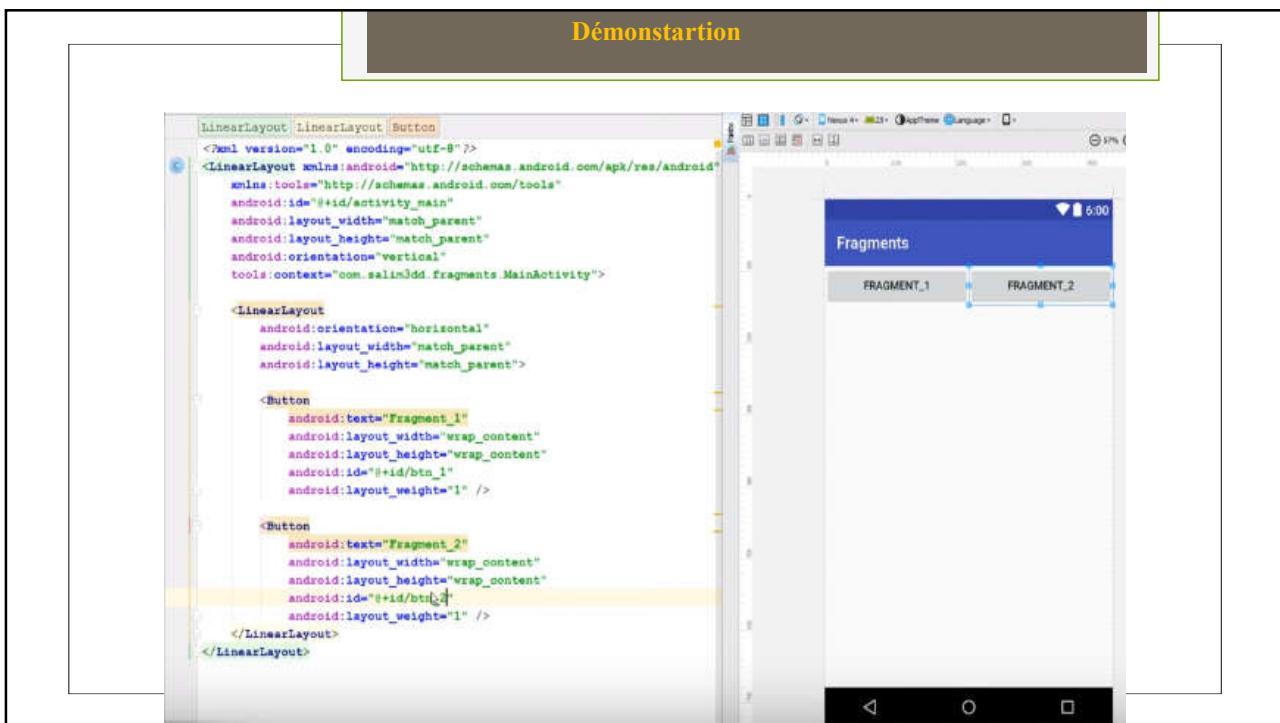


**Un fragment est un composant indépendant qui peut être utilisé dans une activité. Un fragment encapsule des fonctionnalités qui le rendent facile à réutiliser dans une activité ou dans une mise en page.**

## Démonstration







## JSON & Android

**JSON: JavaScript Object Notation.**

**JSON est un format d'échange de données.**  
**La plate-forme Android inclut les bibliothèques json.org qui permettent de travailler facilement avec des fichiers JSON**

## JSON & Android Sending Data

```

<html>
<body>
<h2>Exemple.</h2>
<script>
var myObj = { "name":"Karim", "age":23, "city":"CasaBlanca" };
var myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;

</script>
</body>
</html>

```

**demo\_json.php:**  
**Karim from CasaBlanca is 23**

[https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)

## JSON & Android Receiving Data

```

<html>
<body>
<h2>Exemple.</h2>
<p id="demo"></p>
<script>
var myJSON = { "name":"Karim", "age":23, "city":"Casablanca" };
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
</script>
</body>
</html>

```

**Exemple.**  
**Karim**

## JSON & Android Demonstration

<https://api.myjson.com/bins/kp9wz>

```
{"employees": [
    {"firstname": "John", "age": 30, "mail": "john@gmail.com"},  

    {"firstname": "Jimmy", "age": 25, "mail": "jimmy@gmail.com"},  

    {"firstname": "Jenny", "age": 22, "mail": "jenny@gmail.com"},  

    {"firstname": "Jeremy", "age": 40, "mail": "jeremy@gmail.com"},  

    {"firstname": "Justin", "age": 32, "mail": "justin@gmail.com"}
]}
```



<http://myjson.com/>

John, 30, john@gmail.com  
 Jimmy, 25, jimmy@gmail.com  
 Jenny, 22, jenny@gmail.com  
 Jeremy, 40, jeremy@gmail.com  
 Justin, 32, justin@gmail.com

### JSON & Android Demonstration

**Ajouter implementation 'com.android.volley:volley:1.0.0' à build.gradle puis Sync**

```

dependencies {
    ...
    implementation 'com.android.volley:volley:1.0.0'
}

```

**Ajouter <uses-permission android:name="android.permission.INTERNET"/>**

```

<manifest ...>
    ...
    <application ...>
        ...
        <uses-permission android:name="android.permission.INTERNET" />
    </application>
</manifest>

```

### JSON & Android Demonstration

**activity\_main.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android">
    <TextView />
    <Button />
</RelativeLayout>

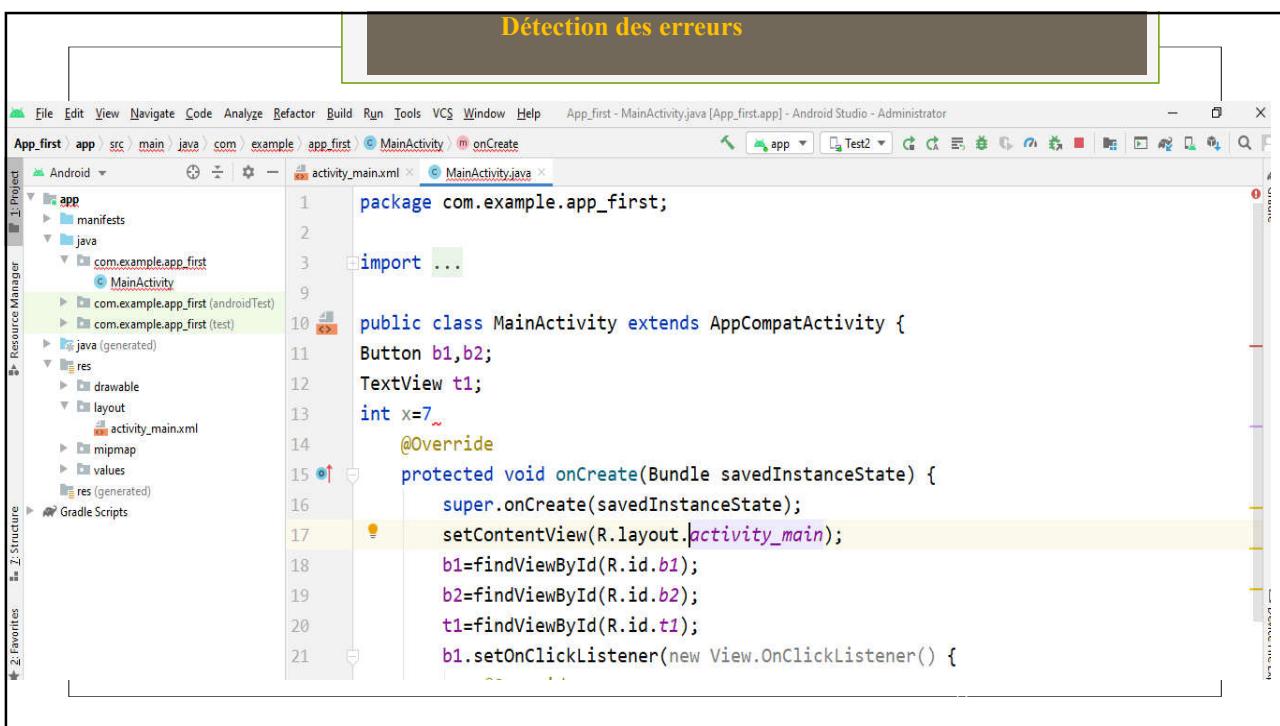
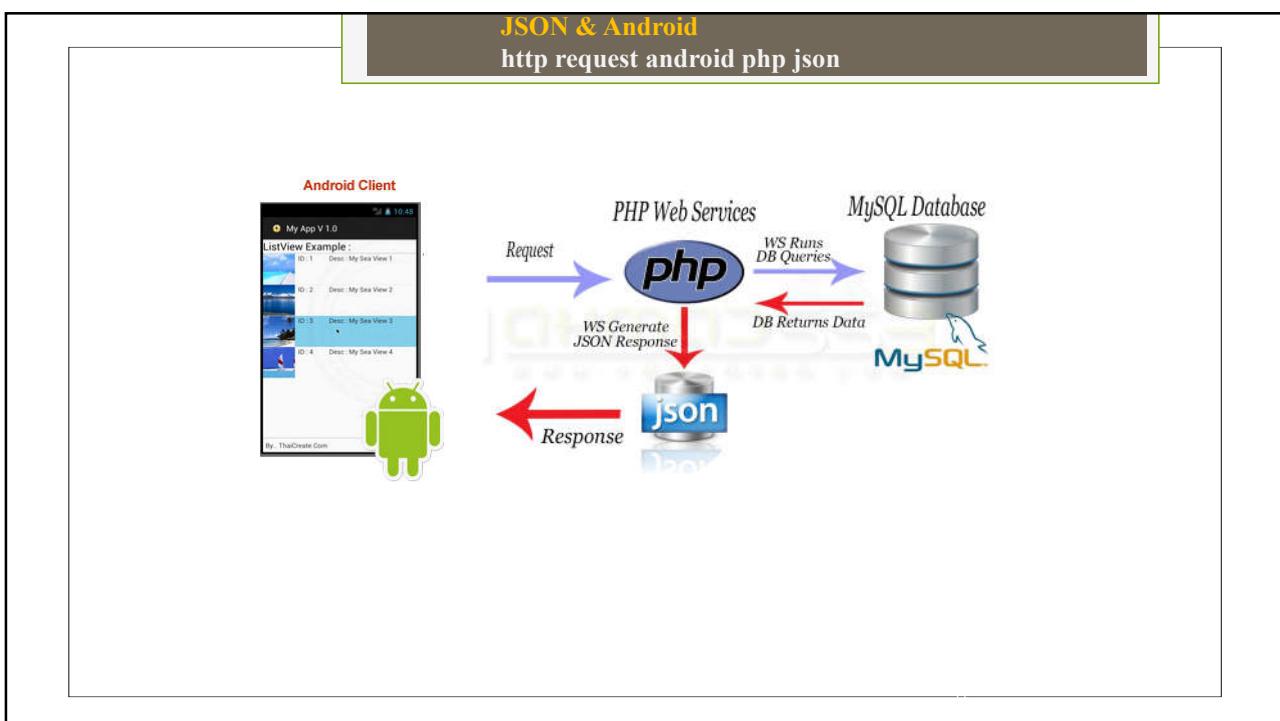
```

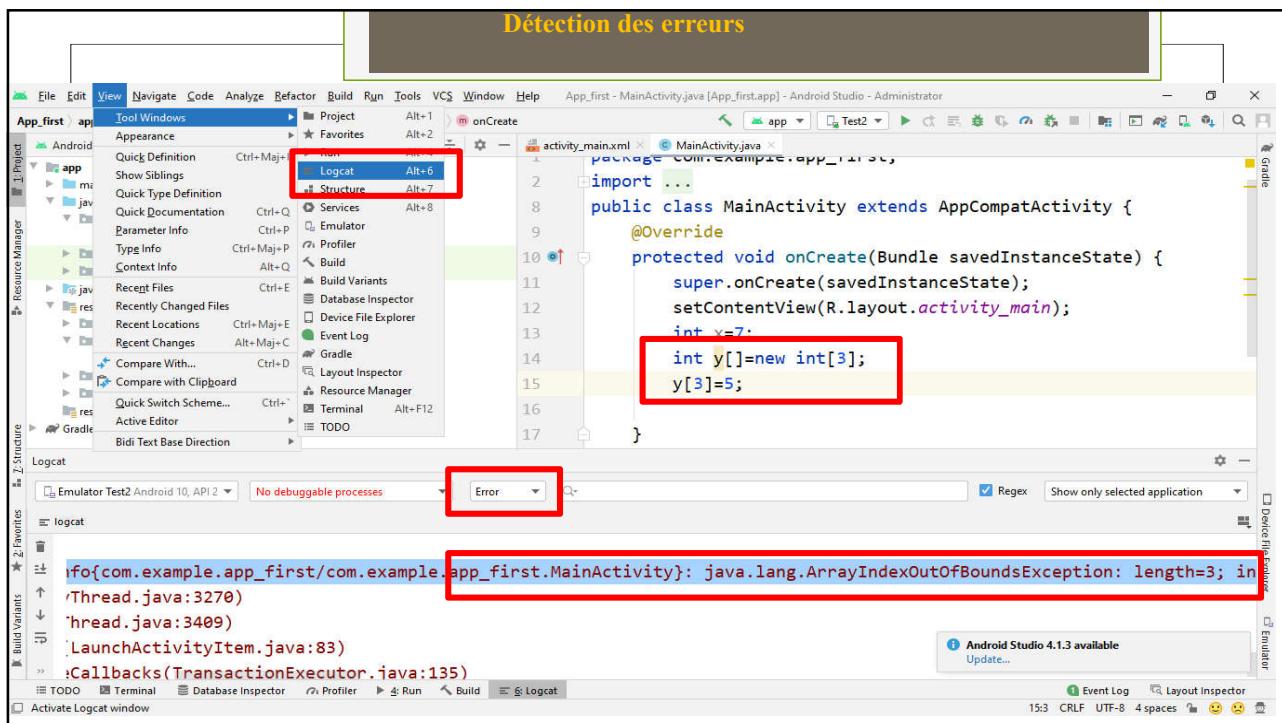
**MainActivity.java**

```

public class MainActivity extends AppCompatActivity {
    private TextView mTextViewResult;
    private RequestQueue mQueue;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mTextViewResult = findViewById(R.id.text_view_result);
        Button buttonParse = findViewById(R.id.button_parse);
        mQueue = Volley.newRequestQueue(this);
        buttonParse.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) { jsonParse(); } });
    }
    private void jsonParse() { mQueue.add(request); }
}

```





Détection des erreurs

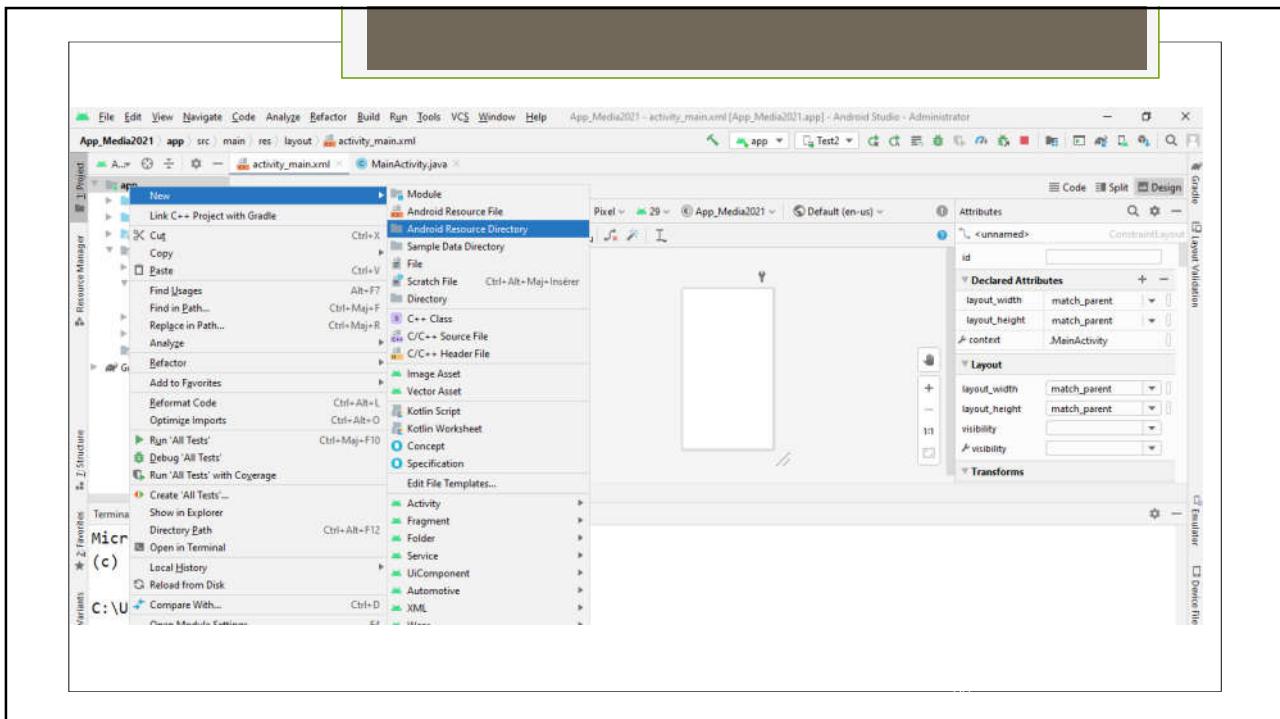
Google Play Developer Console

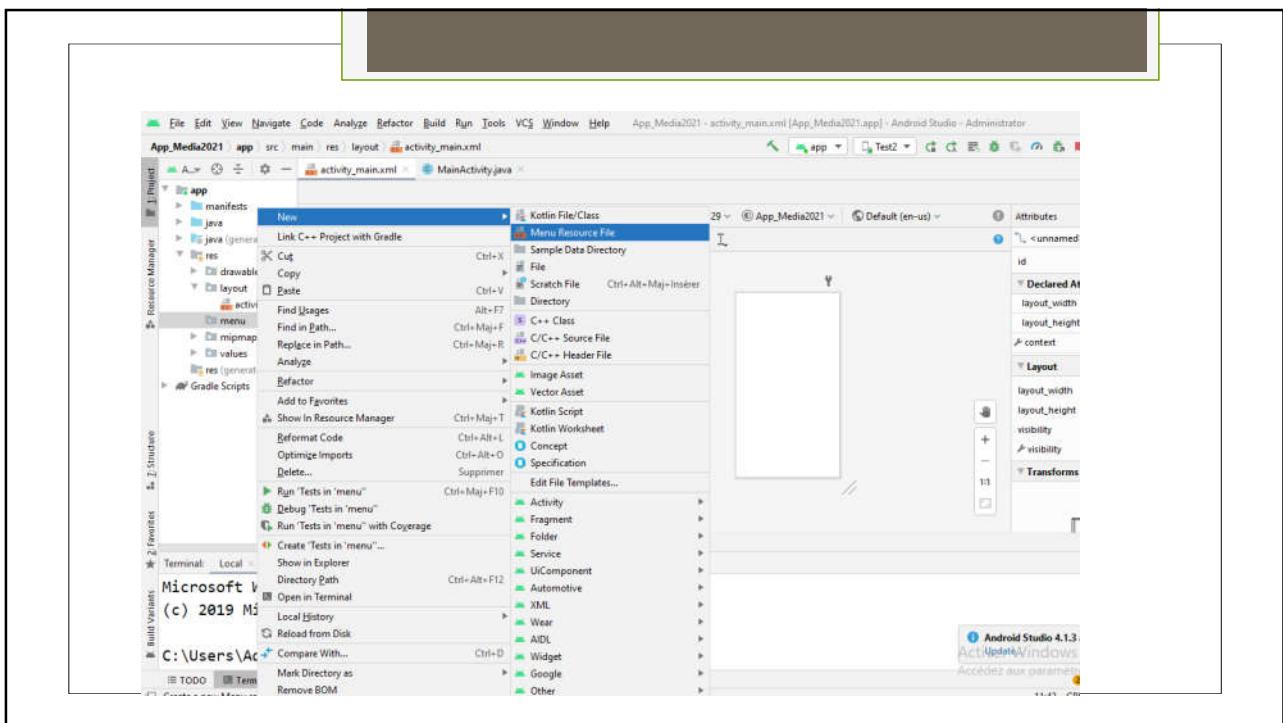
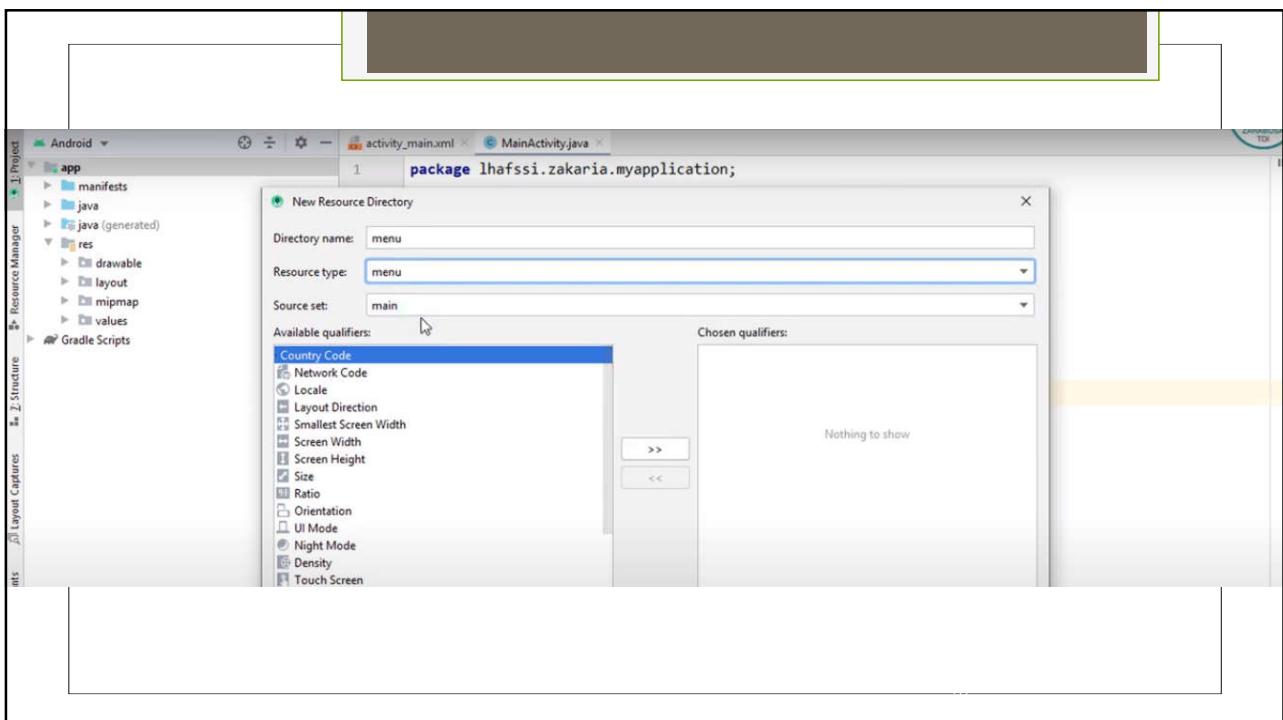
ALL APPLICATIONS

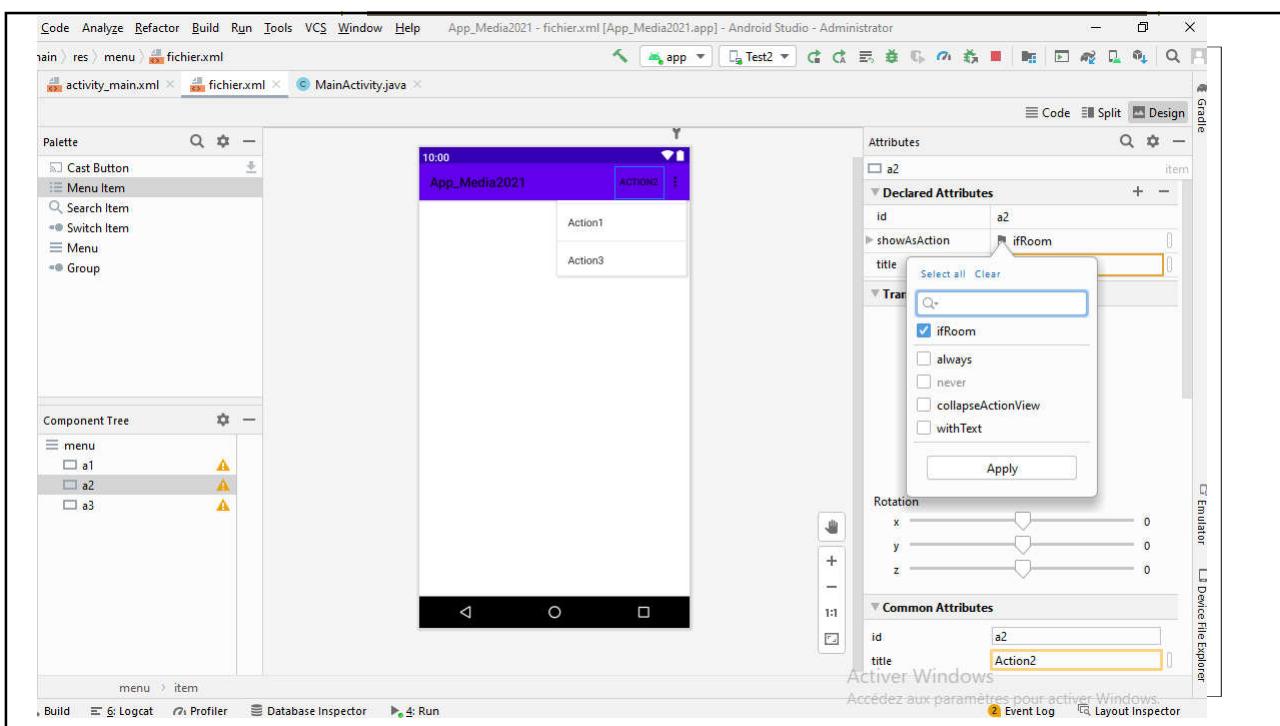
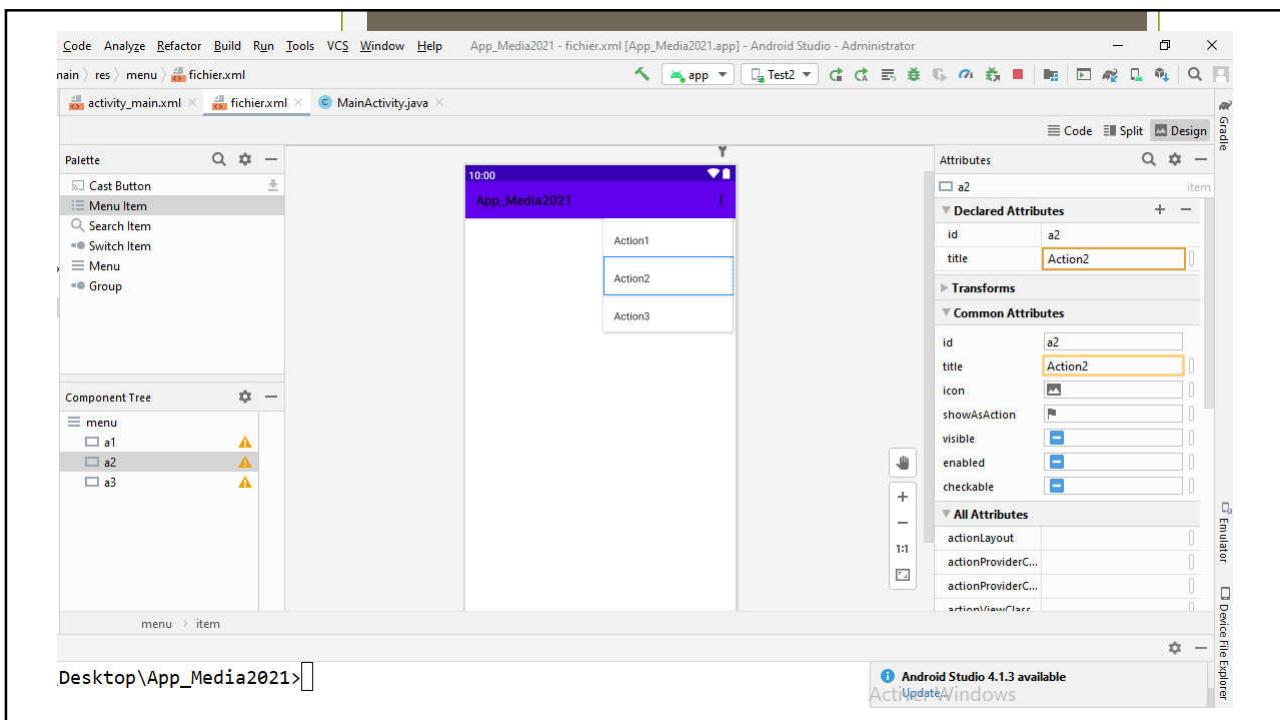
+ Add new application

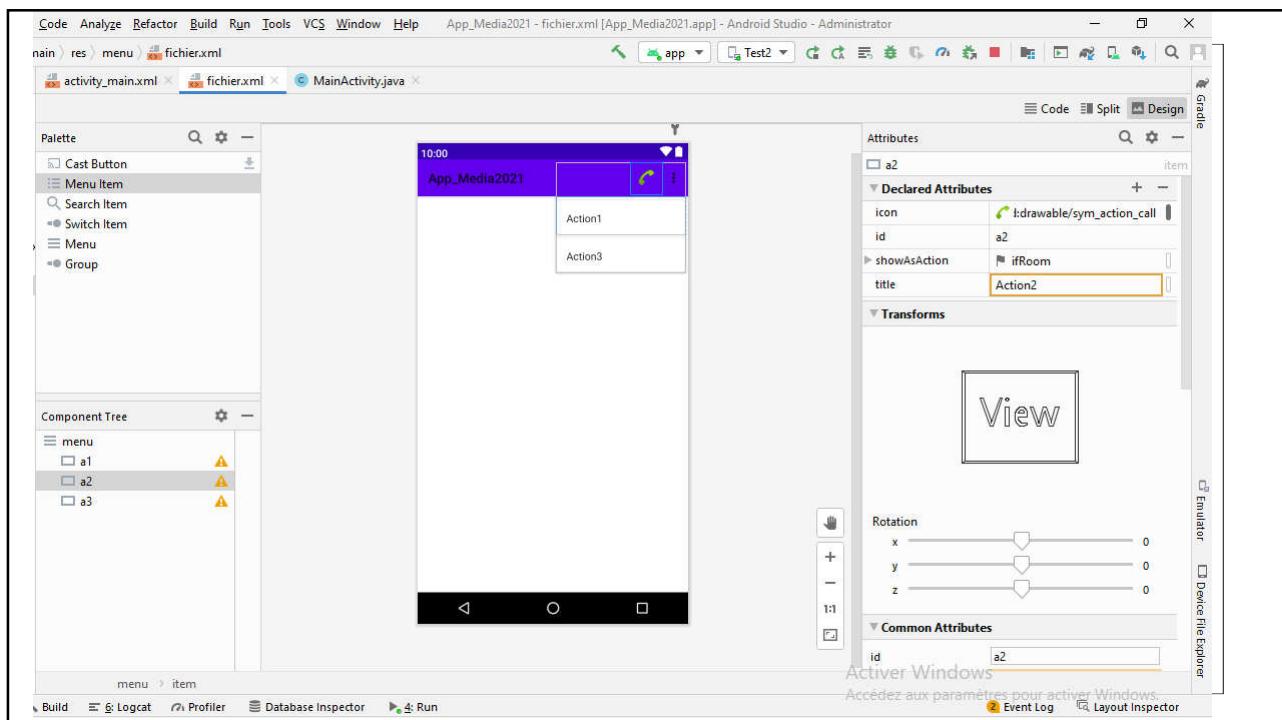
| APP NAME                 | PRICE | CURRENT / TOTAL INSTALLS | AVG. RATING / TOTAL # | CRASHES & ANRS | LAST UPDATE  | STATUS    |
|--------------------------|-------|--------------------------|-----------------------|----------------|--------------|-----------|
| advanced Calculator 2.0  | Free  | 6 / 61                   | ★ 5.00 / 1            | 1              | May 4, 2015  | Published |
| Best offers 4.0          | Free  | 76 / 195                 | ★ 4.87 / 15           | —              | Jan 3, 2016  | Published |
| Find my Car 3.0          | Free  | 43 / 163                 | ★ 4.40 / 5            | 1              | Mar 18, 2015 | Published |
| Find my phone 14.0       | Free  | 7,981 / 75,497           | ★ 3.82 / 381          | 15             | Dec 18, 2015 | Published |
| Find my phone by PC 18.0 | Free  | 755 / 5,232              | ★ 3.56 / 34           | 9              | Dec 18, 2015 | Published |
| Flashlight 1.0           | Free  | 22 / 181                 | ★ 5.00 / 3            | —              | Dec 27, 2014 | Published |
| Infix To Postfix 2.0     | Free  | 9 / 84                   | ★ 3.67 / 3            | 1              | May 4, 2015  | Published |
| Quran Listen Online 18.0 | Free  | 12,915 / 63,586          | ★ 4.56 / 584          | 46             | Dec 27, 2015 | Published |

Page 1 of 1







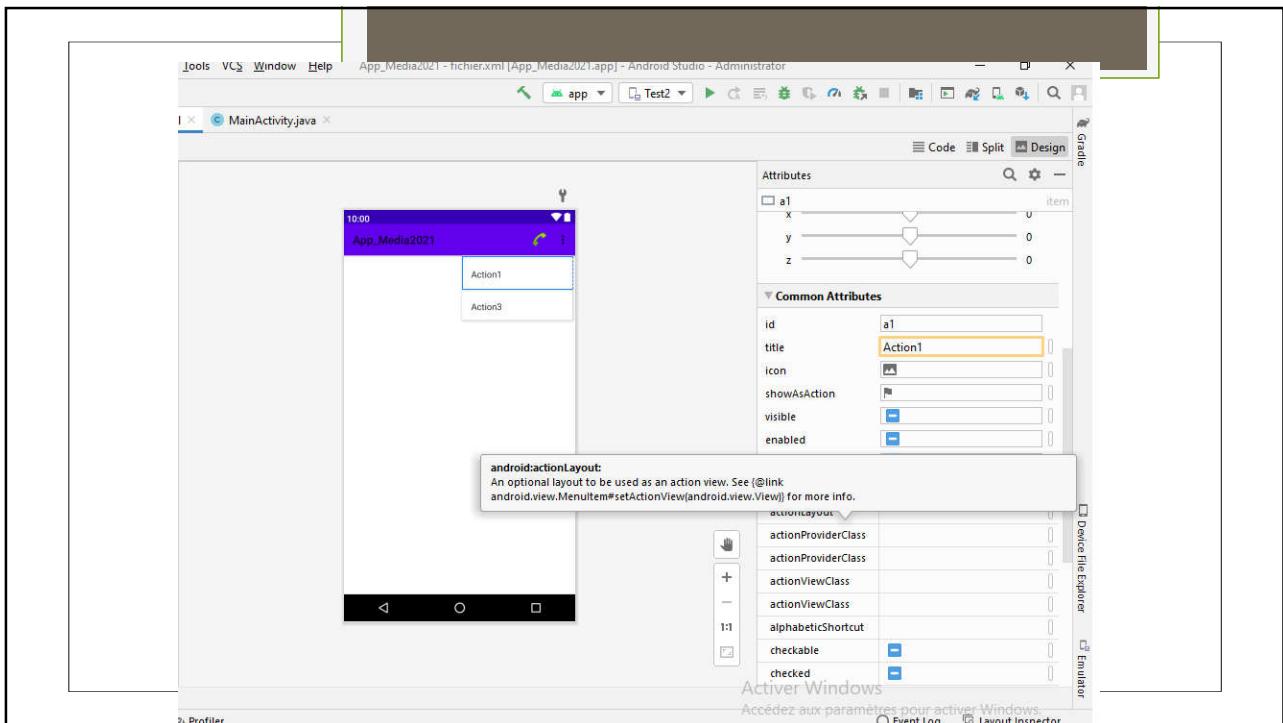


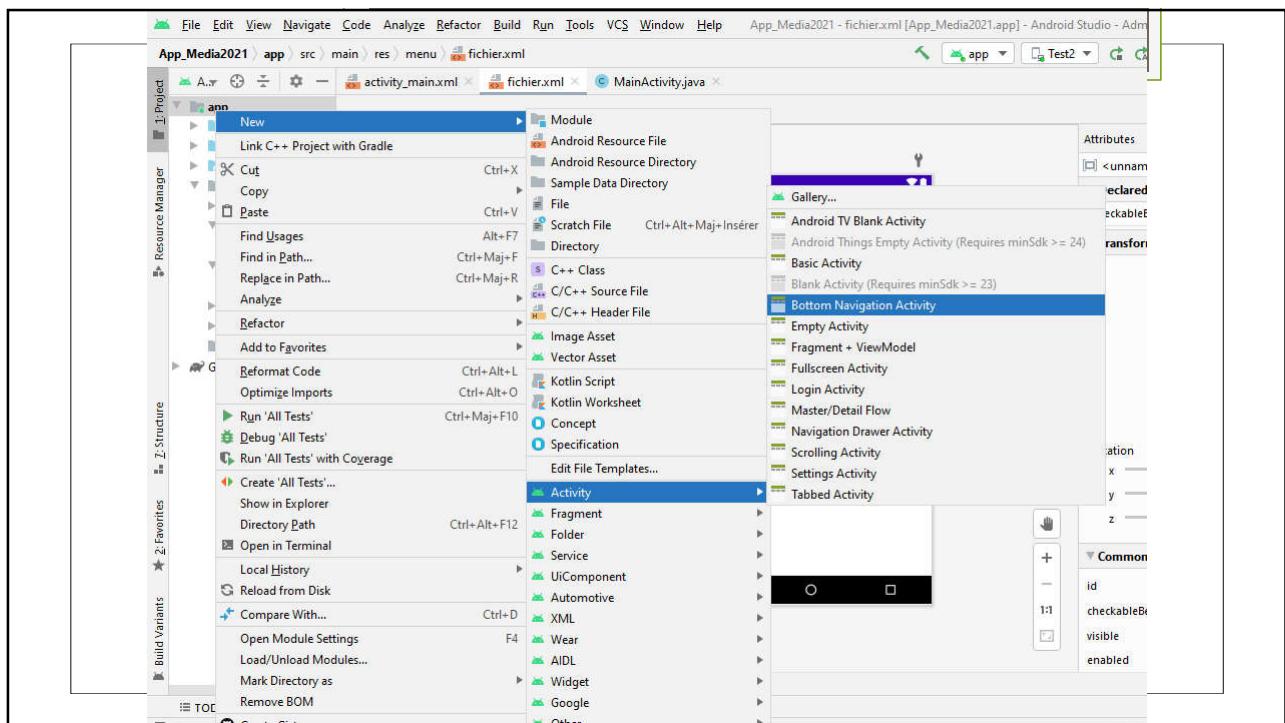
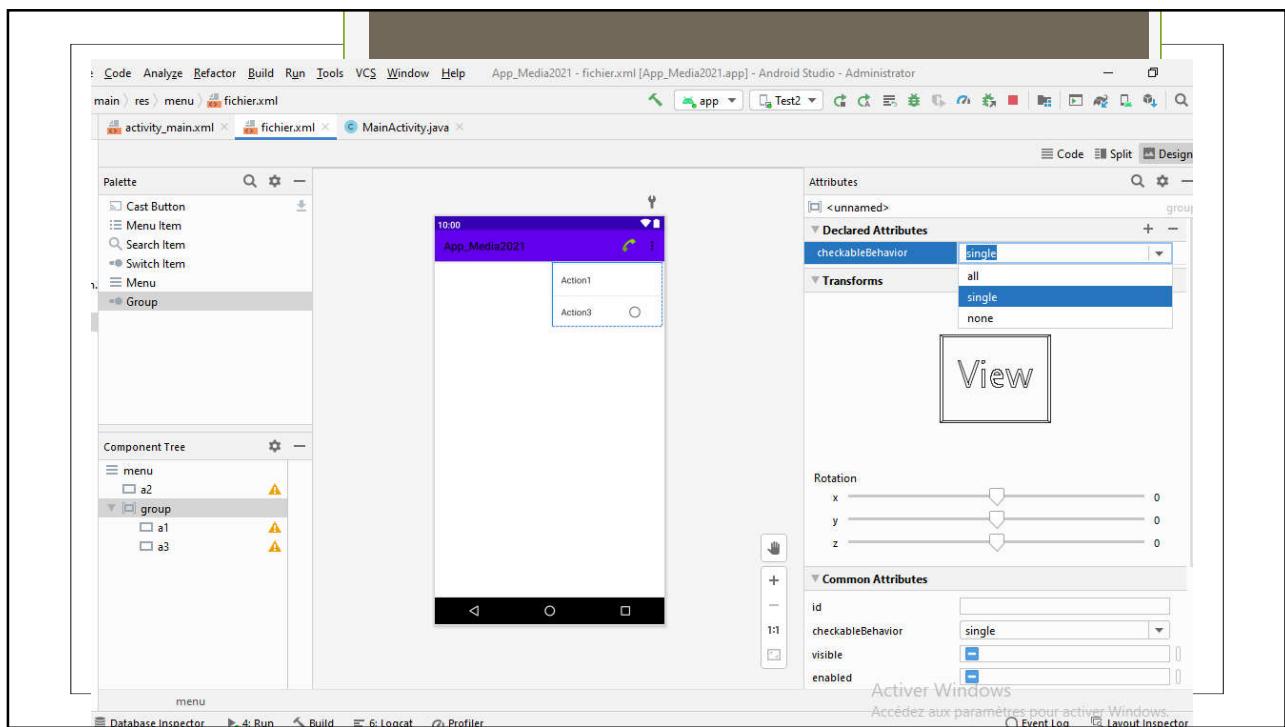
```

3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.Menu;
6
7 public class MainActivity extends AppCompatActivity {
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13
14     @Override
15     public boolean onCreateOptionsMenu(Menu menu) {
16         getMenuInflater().inflate(R.menu.fichier,menu);
17         return super.onCreateOptionsMenu(menu);
18     }
19 }
  
```



```
@Override  
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
    int id = item.getItemId();  
    if (id == R.id.a1)  
    {  
        Toast.makeText(getApplicationContext(),"Bonjour",Toast.LENGTH_SHORT).show();  
    }  
    return super.onOptionsItemSelected(item);  
}
```





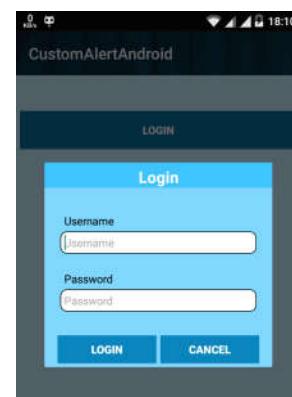
## WEBVIEW ET LES PERMISSIONS

Android System WebView est donc un mini-navigateur (basé sur Google Chrome) qui s'intègre aux diverses applications installées sur votre dispositif Android (smartphone, tablette).



## POP UP ACTIVITY

The popup window is a floating container that appears on top of the current activity.

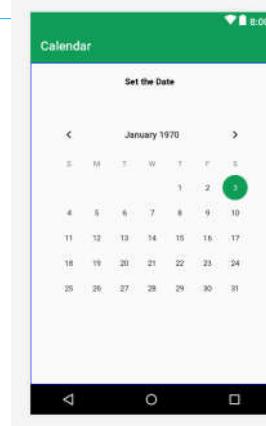


## ANDROID CALENDARVIEW

**CalendarView** is a widget that are used to display and select dates. However, you can configure range of dates to be shown in the calendar.

```
CalendarView calender;
calender = findViewById(R.id.calender);
```

**onSelectedDayChange:** In this method, we get the values of days, months and years that is selected by the user.

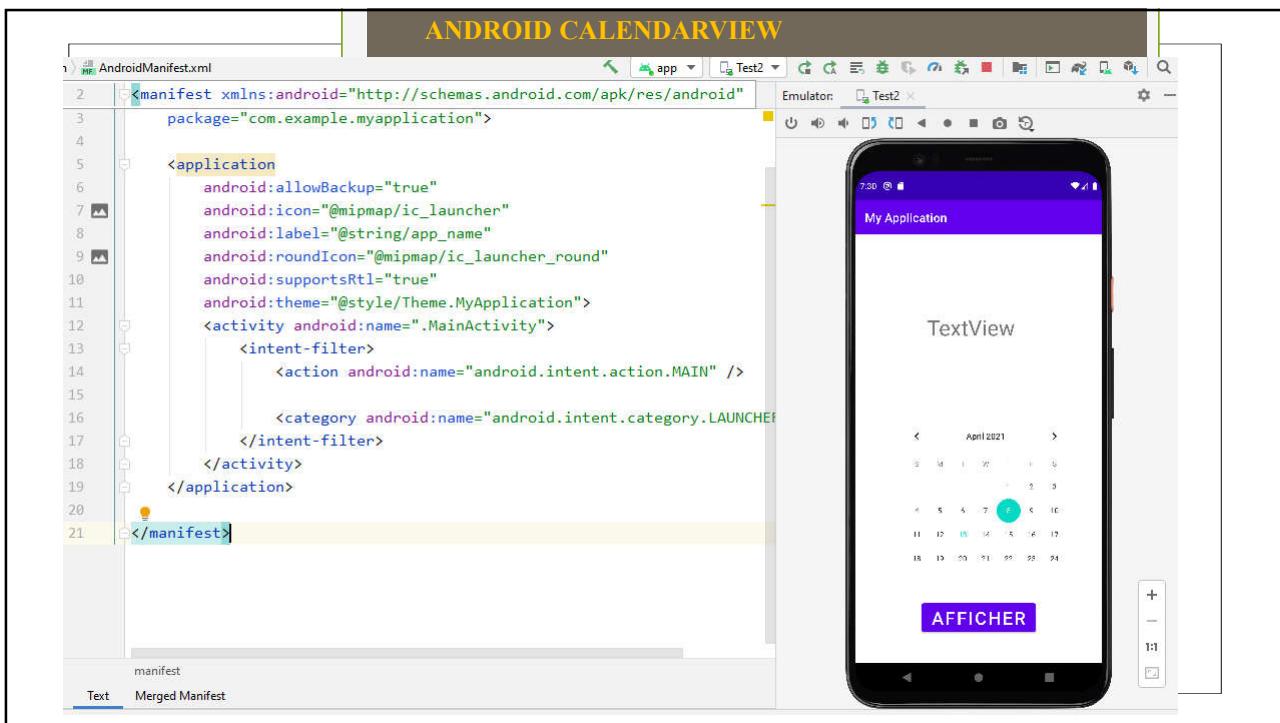
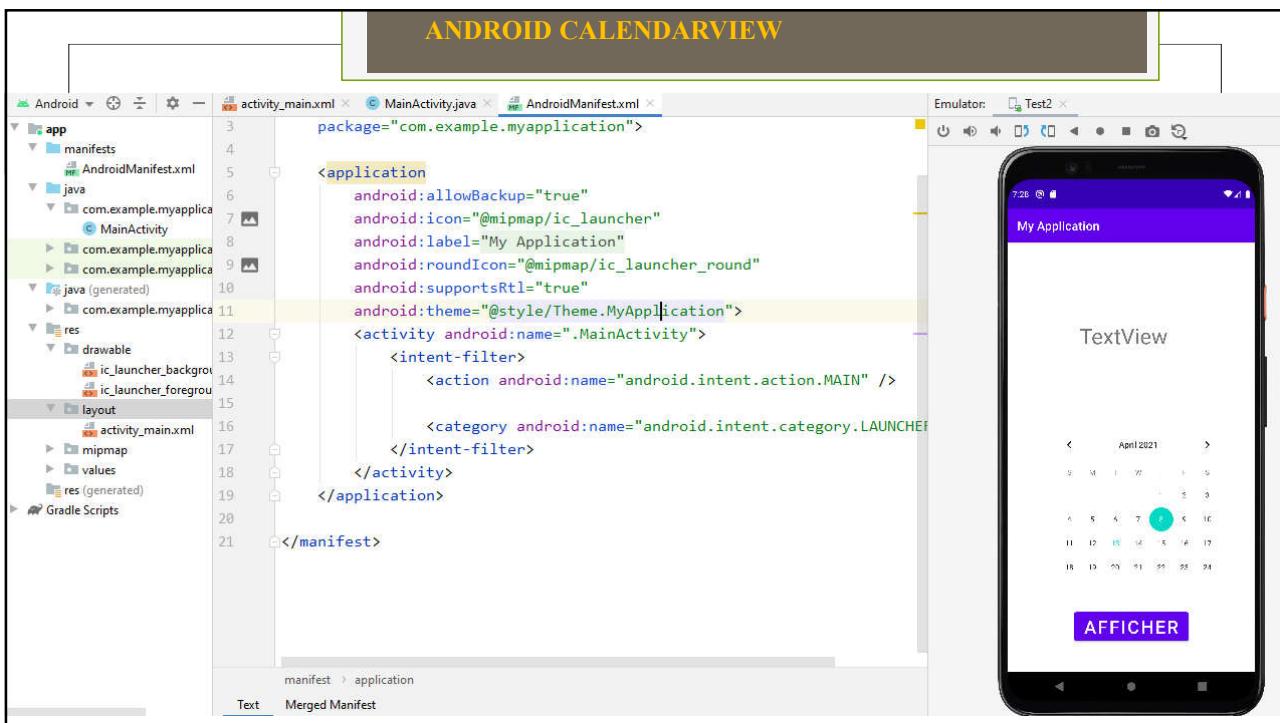


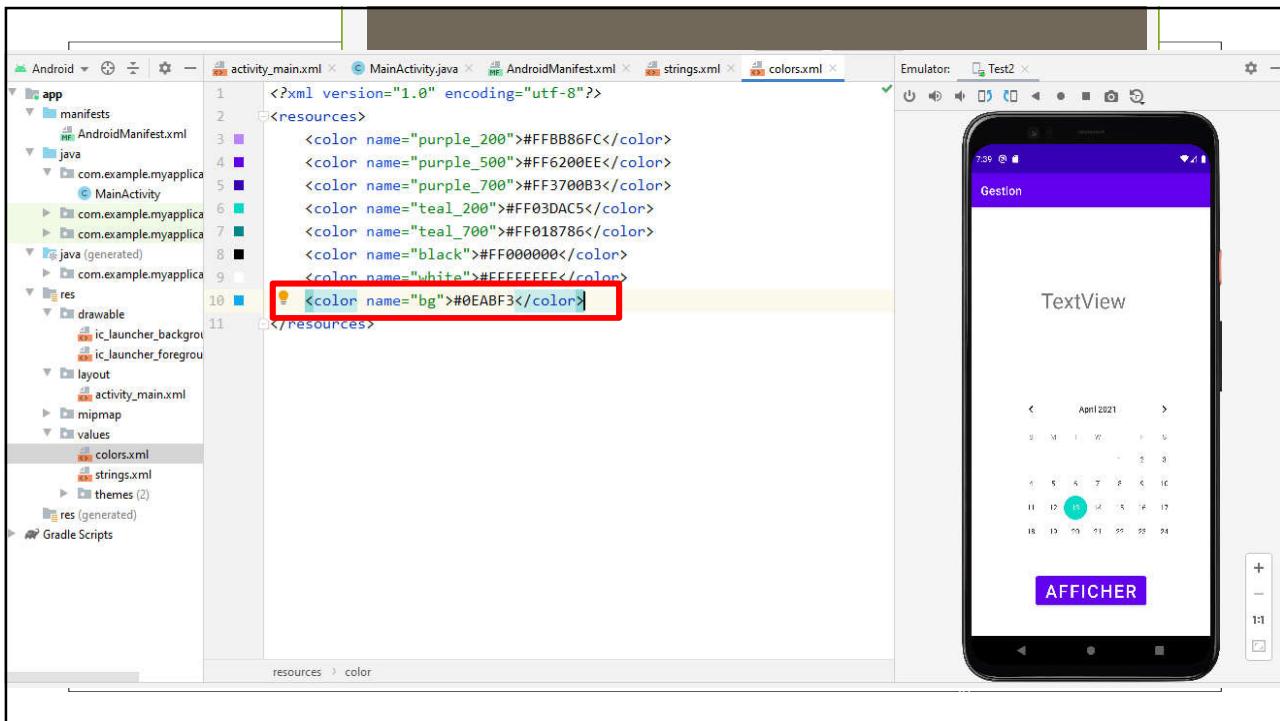
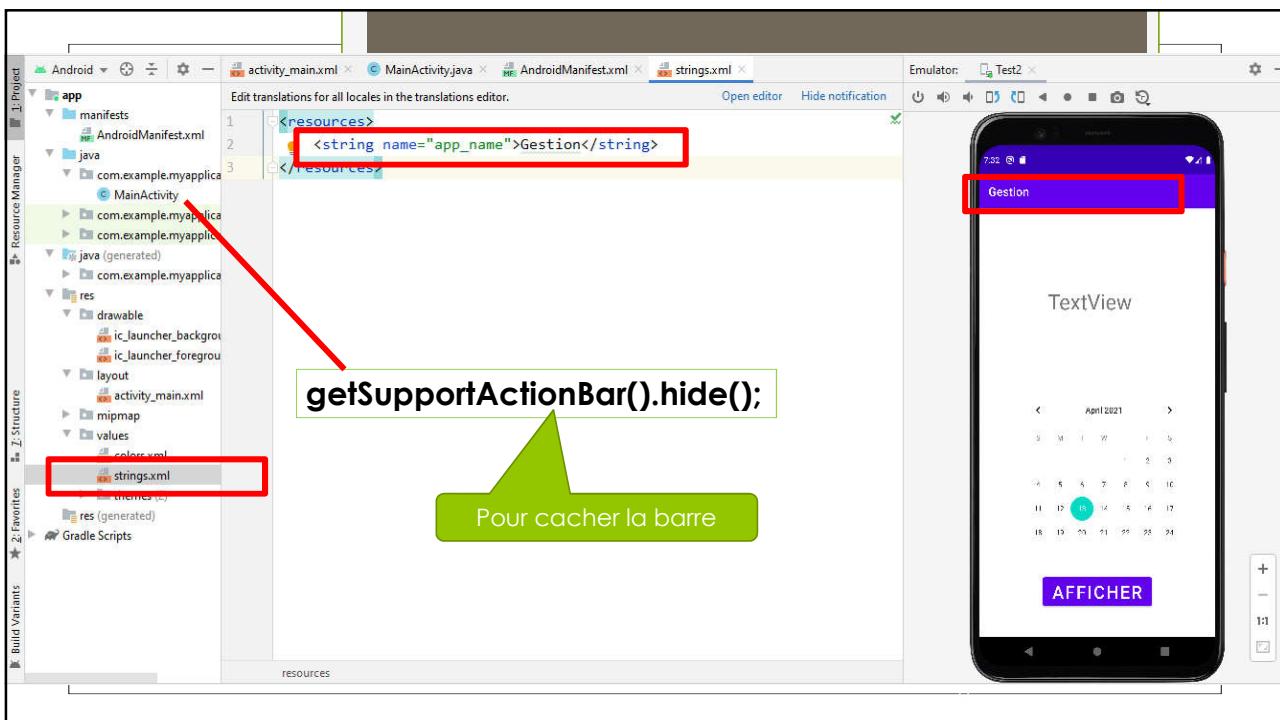
## ANDROID CALENDARVIEW

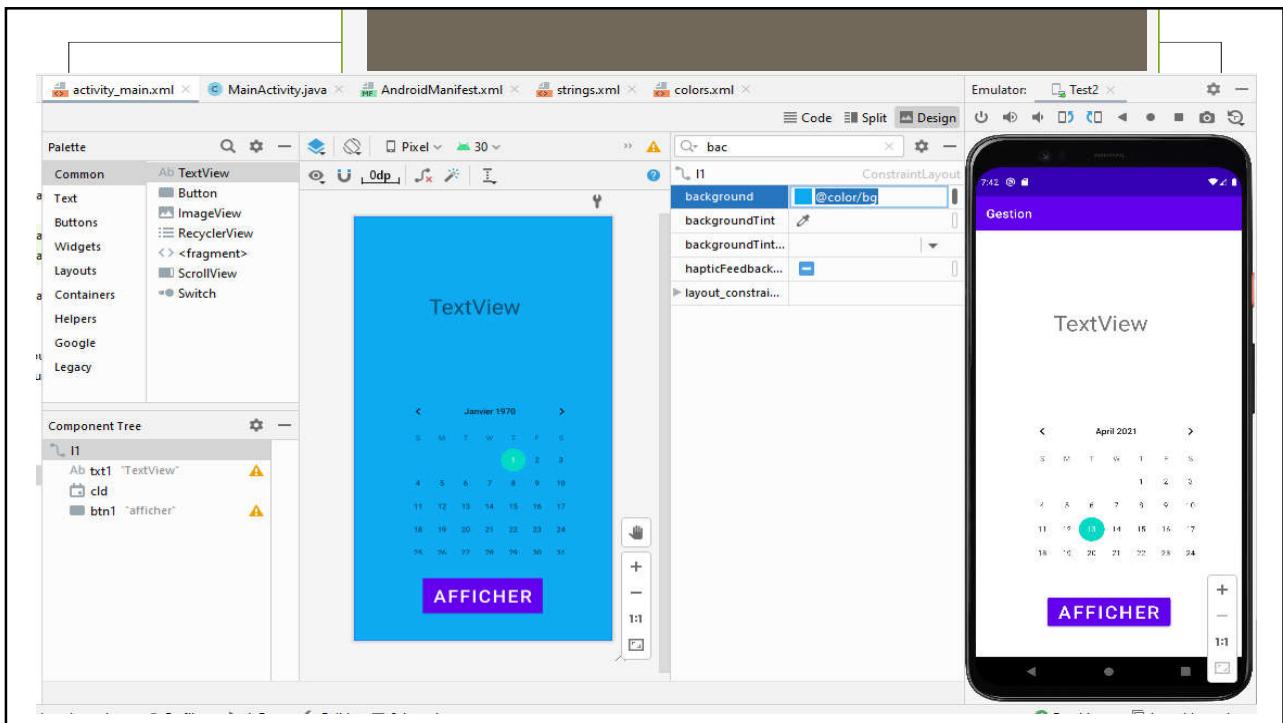
```
calender.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
    @Override
    public void onSelectedDayChange(@NonNull CalendarView calendarView,
        int i, int i1, int i2) {
        String Date
            = i2 + "-"
            + (i1+ 1) + "-" + i;

        // set this date in TextView for Display
        txt1.setText(Date);
    }
});
```





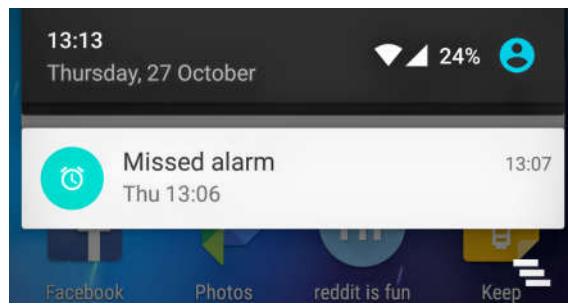


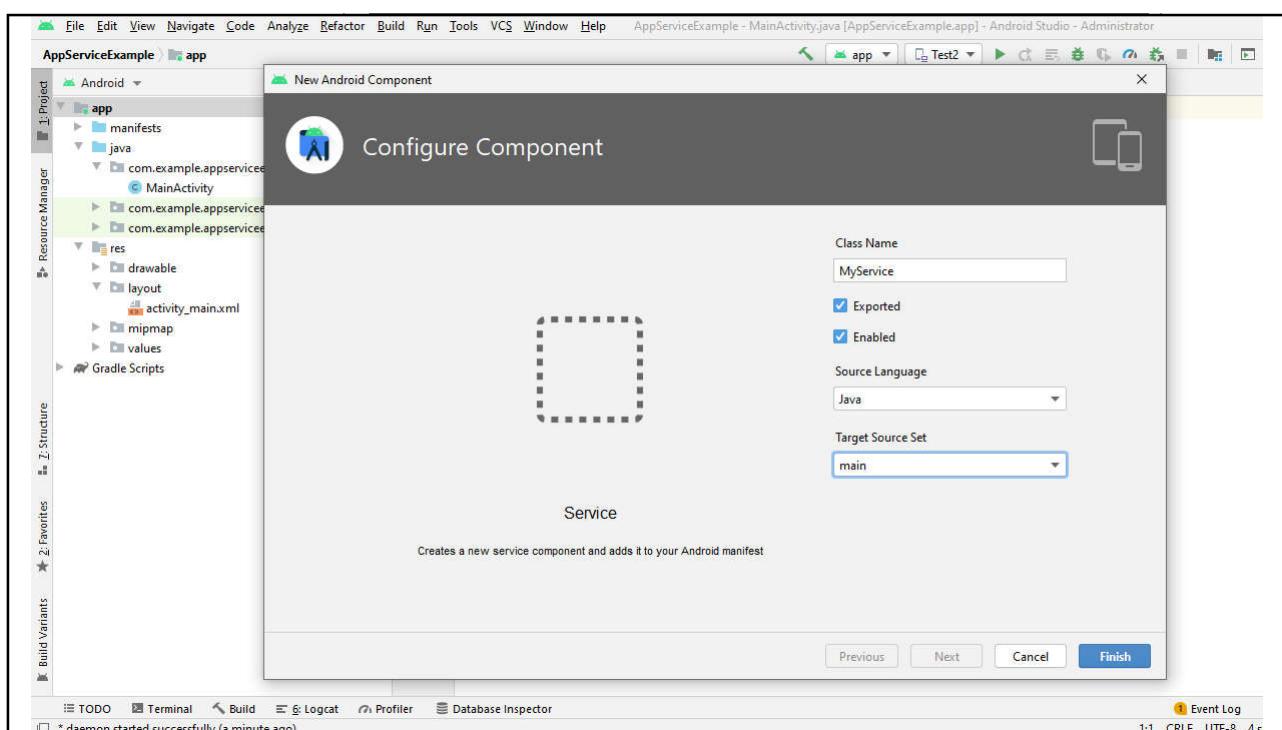
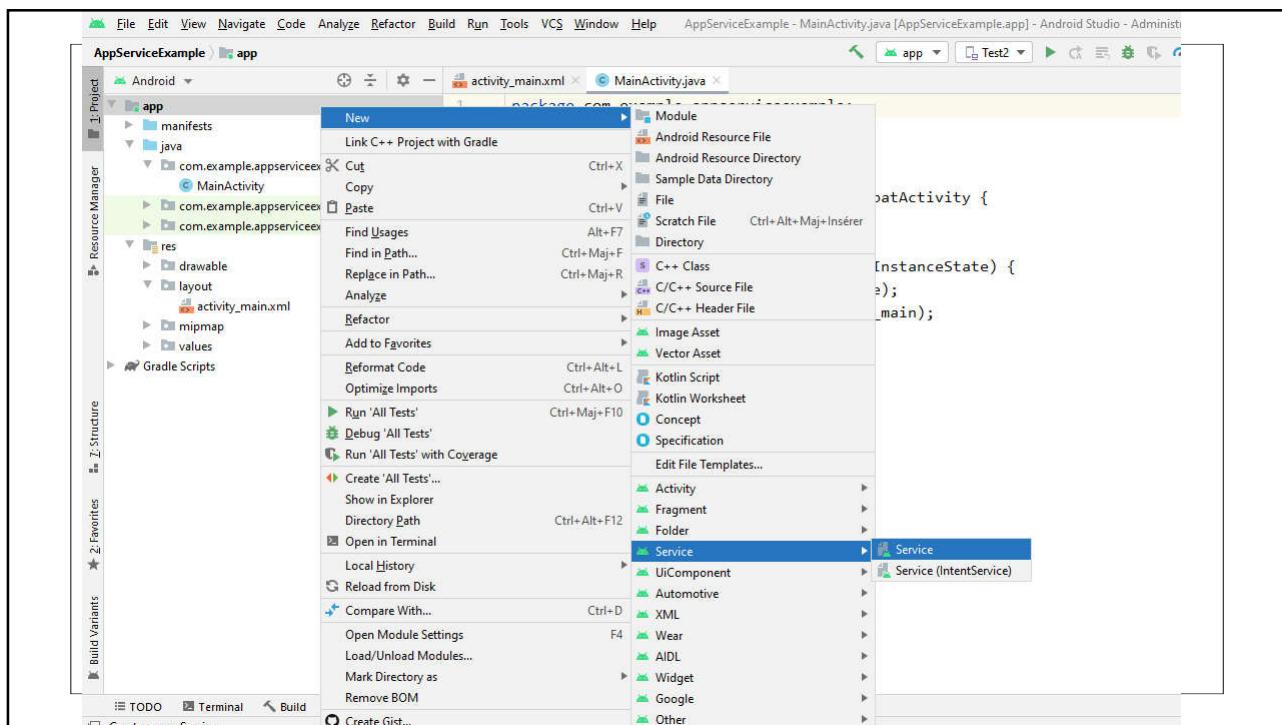


## LES SERVICES EN ANDROID STUDIO

A Service is an application component that can perform long-running operations in the background. It does not provide a user interface. Once started, a service might continue running for some time, even after the user switches to another application.

<https://developer.android.com/guide/components/services>





The screenshot shows the Android Studio interface with the project 'AppServiceExample' open. In the code editor, the file 'MyService.java' is displayed. The class definition is as follows:

```

1 package com.example.appserviceexample;
2
3 import android.app.Service;
4 import android.content.Intent;
5 import android.os.IBinder;
6
7 public class MyService extends Service {
8     public MyService() {
9     }
10
11     @Override
12     public IBinder onBind(Intent intent) {
13         // TODO: Return the communication channel to the service.
14         throw new UnsupportedOperationException("Not yet implemented");
15     }
16 }

```

A 'Select Methods to Override/Implement' dialog is open over the code editor. It lists methods from the 'android.app.Service' interface. The method 'onStartCommand(Intent intent, int flags, int startId)' is highlighted with a blue border. Other listed methods include onCreate(), onStart(), onStartCommand(), onStartCommand(), onDestroy(), onConfigurationChanged(), onLowMemory(), onTrimMemory(), onUnbind(), onRebind(), onTaskRemoved(), dump(), attachBaseContext(), getContextWrapper(), getBaseContext(), getAssets(), getResources(), getPackageManager(), getContentResolver(), getMainLooper(), getMainExecutor(), and getApplicationContext(). At the bottom of the dialog, there are 'OK' and 'Cancel' buttons, and a checked checkbox labeled 'Insert @Override'.

