

# Rapport de TP 5 : Vues à adaptateur en Java

Boutejdir Mohamed Amine - Mehdi Mohammed  
BTS DAI - 2ème Année

Janvier 2026

**Module :** Développement d'applications mobiles  
**Sujet :** ArrayAdapter, ListView, GridView et Spinner

# Table des matières

<b>1</b>	<b>Introduction et Objectifs</b>	<b>3</b>
<b>2</b>	<b>Conception de l'application</b>	<b>3</b>
2.1	L'entité Country . . . . .	3
2.2	L'Adaptateur Personnalisé . . . . .	3
<b>3</b>	<b>Captures d'écran de l'application</b>	<b>3</b>
<b>4</b>	<b>Réponses aux questions du TP</b>	<b>5</b>
<b>5</b>	<b>Conclusion</b>	<b>5</b>

# 1 Introduction et Objectifs

Ce travail pratique a pour but de maîtriser l'affichage de collections de données dans une application Android. [cite,start] Les principaux objectifs pédagogiques sont :

Comprendre le rôle d'un  `ArrayAdapter`.

Utiliser différentes vues :  `Spinner`,  `ListView` et  `GridView`.

Créer un adaptateur personnalisé pour afficher des objets complexes (Image + Texte).

Gérer les événements de sélection (Listeners).

## 2 Conception de l'application

### 2.1 L'entité Country

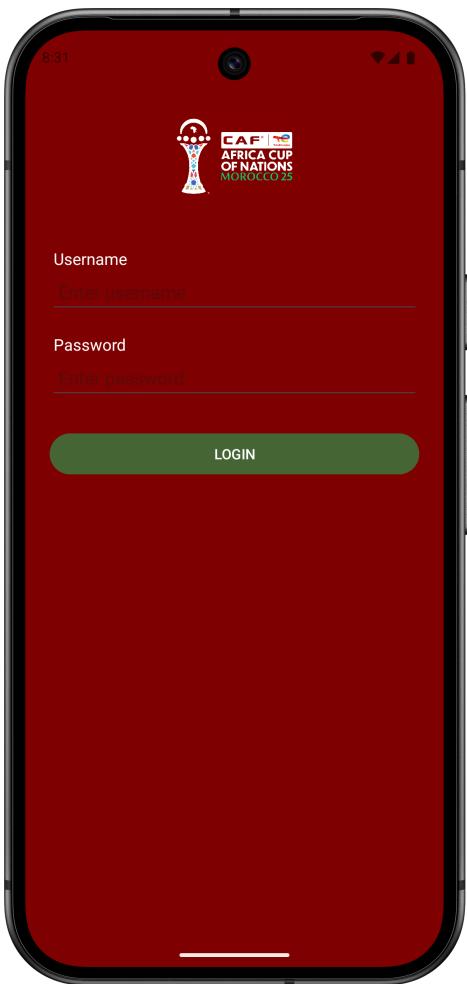
Nous avons créé une classe `Country` pour modéliser les données. Chaque pays possède un identifiant de ressource pour son drapeau (entier) et un nom (chaîne de caractères).

### 2.2 L'Adaptateur Personnalisé

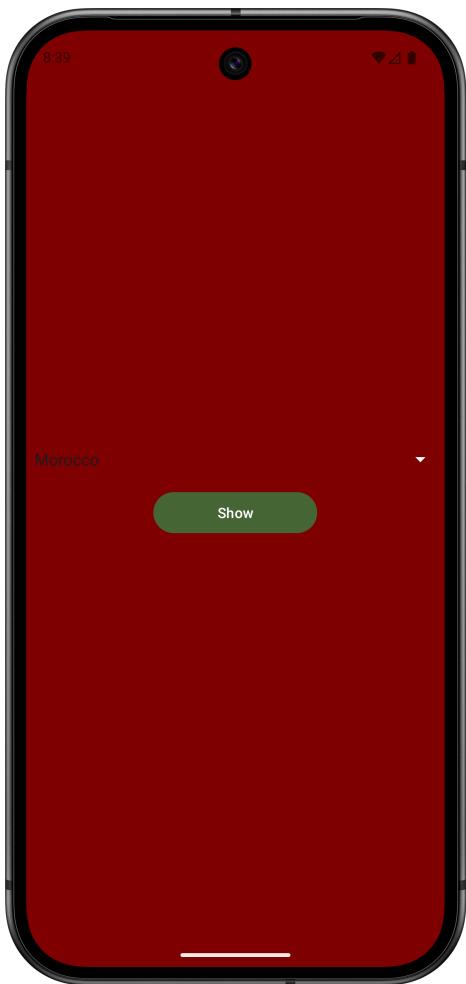
Pour afficher à la fois le drapeau et le nom du pays, nous avons étendu la classe  `ArrayAdapter` pour créer `CountryAdapter`. La méthode centrale est  `getView()`, qui permet de "gonfler" (inflate) le layout XML personnalisé pour chaque item de la liste.

## 3 Captures d'écran de l'application

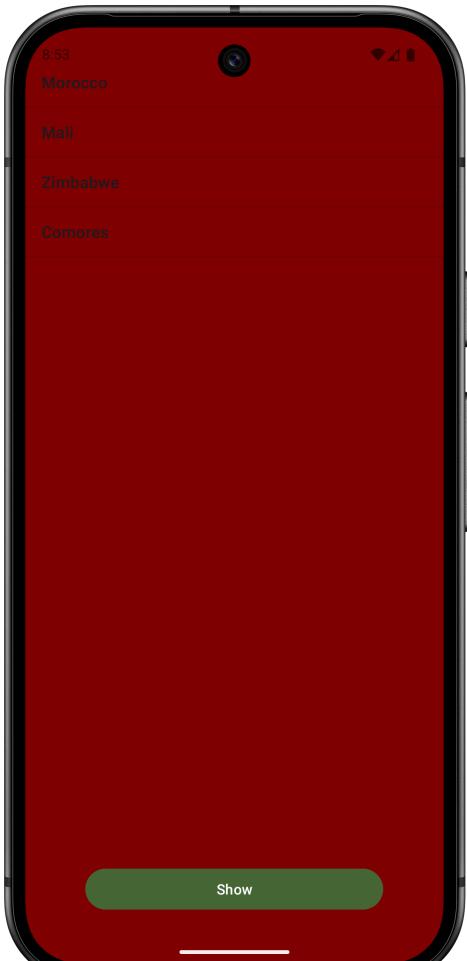
Cette section présente les différentes interfaces réalisées : la page de connexion, le  `Spinner`, la  `ListView` et la  `GridView` personnalisée.



(a) Écran de Login



(b) Sélection via Spinner



(c) ListView avec texte blanc



(d) GridView avec titre Groupe A

Comme illustré sur la figure 1d, l'interface GridView a été modernisée avec un titre "Groupe A" centré et une gestion des marges pour éviter le chevauchement avec la barre d'état.

## 4 Réponses aux questions du TP

### 1. Quel est le rôle de ArrayAdapter ?

L'ArrayAdapter sert d'intermédiaire entre la source de données (ici une `ArrayList`) et la vue (`ListView`, `Spinner`). Il convertit chaque objet de la liste en une vue affichable à l'écran.

### 2. Quelle est la différence entre Spinner et ListView ?

Le Spinner est une liste déroulante qui n'affiche qu'un seul élément à la fois (sélection compacte), tandis que la ListView affiche la liste complète verticalement avec une barre de défilement.

### 3. Que représente le paramètre position dans la méthode `getView()` ?

Le paramètre `position` correspond à l'index de l'élément dans la liste de données que l'adaptateur est en train de dessiner à l'instant T.

### 4. Pourquoi peut-on utiliser le même adaptateur pour plusieurs vues ?

Car l'adaptateur fournit des vues (Views) génériques. C'est le composant parent (`ListView` ou `GridView`) qui décide ensuite comment disposer ces vues (les empiler ou les mettre en grille).

## 5 Conclusion

Ce TP nous a permis de comprendre le cycle de vie des vues adaptatives sous Android. Nous avons appris à personnaliser l'affichage en utilisant des layouts XML spécifiques (`item_country.xml`) et à gérer l'interaction utilisateur via les Listeners.