CR : ToDoList



Table des matières

[I. Création des objets ItemTodo, ListeTodo et ProfilListeTodo 2](#_Toc10451281)

[II. Sérialisation/Désérialisation et stockage des données 3](#_Toc10451282)

[Sérialisation/désérialisation 3](#_Toc10451283)

[Stockage des données dans les SharedPreferences 3](#_Toc10451284)

[III. Développement de 2 RecyclerView 4](#_Toc10451285)

[MyAdapter 4](#_Toc10451286)

[MyAdapterShowList 4](#_Toc10451287)

[IV. Développement des activités : 5](#_Toc10451288)

[MainActivity : 5](#_Toc10451289)

[Création du Layout **activity\_main.xml** 5](#_Toc10451290)

[Fonctionnement de **MainActivity** 5](#_Toc10451291)

[ChoixListActivity 6](#_Toc10451292)

[Création du Layout **activity\_choix\_list.xml** 6](#_Toc10451293)

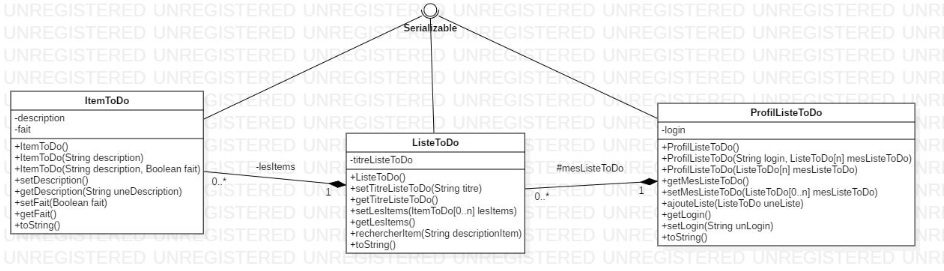
[Fonctionnement de **ChoixListActivity** 6](#_Toc10451294)

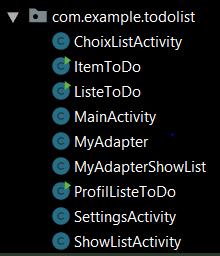
[ShowListActivity 7](#_Toc10451295)

[Création du Layout **activity\_show\_list.xml** 7](#_Toc10451296)

[Fonctionnement de **ShowListActivity** 7](#_Toc10451297)

# Création des objets ItemTodo, ListeTodo et ProfilListeTodo





# Sérialisation/Désérialisation et stockage des données

## Sérialisation/désérialisation

On utilise la librairie Gson pour sérialiser et désérialiser nos ProfilListeTodo :

// ############### Sérialisation/ Désérialisation ###################################  
// Json --> ProfilListeToDo  
public ProfilListeToDo deserialisation(String jsonProfilListe){  
 Gson gson = new Gson()**;** ProfilListeToDo profilListeToDo = gson.fromJson(jsonProfilListe**,**ProfilListeToDo.class)**;** return profilListeToDo**;**}  
  
// ProfilListeToDo --> Json  
public String serialisation(ProfilListeToDo profilListeToDo){  
 Gson gson = new Gson()**;** return gson.toJson(profilListeToDo)**;**}

## Stockage des données dans les SharedPreferences

On stocke la ProfilListeToDo (au format Json) dans les préférences partagées, la clé est le pseudo du profil.

// ############## Gestion des préférences partagées ################################  
 // Chargement des données  
 public ProfilListeToDo recupProfilListe(String pseudo){  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(this)**;** String jsonProfilListe = settings.getString(pseudo**,** "")**;** return deserialisation(jsonProfilListe)**;** }  
  
 // Sauvegarde des données  
 public void sauvegardeProfilListe(ProfilListeToDo newProfilListeToDo**,** String pseudo){  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(this)**;** String jsonProfilListe = serialisation(newProfilListeToDo)**;** SharedPreferences.Editor editor = settings.edit()**;**// editor.clear();  
 editor.putString(pseudo**,** jsonProfilListe)**;** editor.commit()**;** }

# Développement de 2 RecyclerView

## MyAdapter

* Layout pour les vues créées : ***liste\_todo.xml***
* Activité qui implémente de Recycler View : ***ChoixListActivity***
* Définition de la méthode **onItemClicked** ( int posListe )

Qui lance l’activité *ShowListActivity* en lui passant la liste cliquée via Intent explicite.

|  |  |
| --- | --- |
| **Liste\_todo.xml** | |
| TextView affichage titre liste | id : titreListe |

## MyAdapterShowList

* Layout pour les vues créées : ***liste\_item\_todo.xml***
* Activité qui implémente de Recycler View : ***ShowListActivity***
* Définition de la méthode **onItemClicked** ( int posItem, boolean fait )

Qui coche/décoche la checkox lors du clic lors du clic sur un item (et sauvegarde son état dans les préférences partagées).

* Voie d’amélioration : le clic n’est pas pris en compte si l’on coche directement la checkbox … il faut cliquer sur l’item. Objectif : étendre le gestionnaire de clic à la checkbox.

|  |  |
| --- | --- |
| **Liste\_item\_todo.xml** | |
| TextView affichage titre liste | id : item |
| CheckBox fait/à faire | Id : cbFait |

# Développement des activités :

## MainActivity :

### Création du Layout **activity\_main.xml**

|  |  |
| --- | --- |
| **activity\_main.xml** | |
| Zone de texte entrée du pseudo | id : edtPseudo |
| Bouton ok | id : btnOk |
| Bouton reset | Id : btnReset |



### Fonctionnement de **MainActivity**

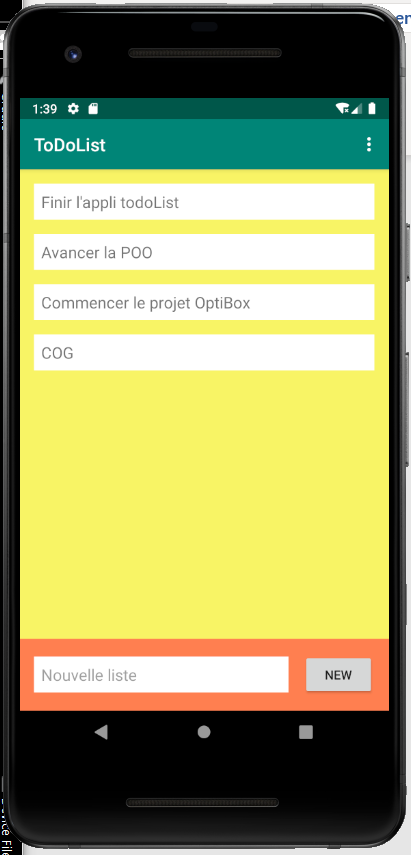
La *MainActivity* préremplit le dernier pseudo entré en le récupérant dans les préférences partagées.

L’utilisateur entre son pseudo, lorsqu’il clique sur « OK », l’activité enregistre le pseudo dans les préférences partagées et le transmet à l’activité *ChoixListActivity* via un intent explicite (et un bundle). L’activité *ChoixListActivity* se lance.

## ChoixListActivity

### Création du Layout **activity\_choix\_list.xml**

|  |  |
| --- | --- |
| **activity\_choix\_list.xml** | |
| Conteneur des listes | id : myRecyclerView |
| Zone de texte création de liste | Id : edtNewList |
| Bouton création de liste | Id : btnNewList |
| Recycler View | Id : myRecyclerView |



### Fonctionnement de **ChoixListActivity**

Au démarrage, *ChoixListActivity* récupère :

* les vues de *activity\_choix\_list.xml* via leur id.
* le pseudo envoyé par MainActivity via un Intent explicite

La vue essentielle est le RecyclerView (en rouge). Instancié durant le *onStart()*, il récupère les listes du ProfiListe associé au pseudo reçu et les affiche sous forme de liste. Le modèle de vue de chaque élément est défini dans un layout à part : *liste\_todo.xml.*

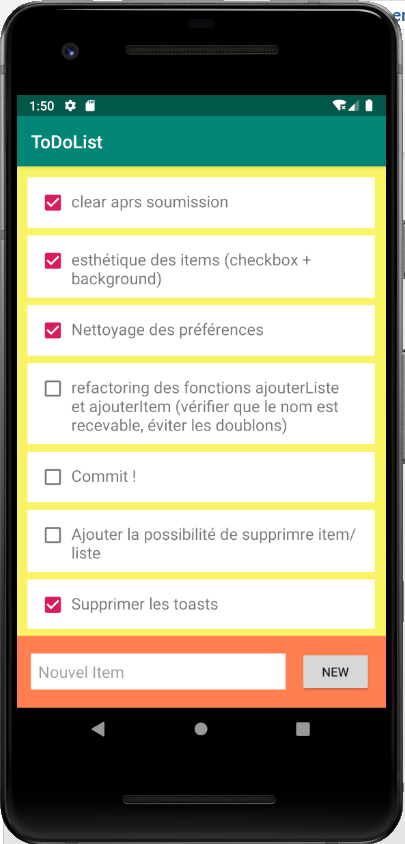
L’utilisateur peut ajouter une nouvelle liste en inscrivant son titre dans le TextView puis en cliquant sur « NEW ». La liste est ajoutée au ProfilListe et le RecyclerView est rappelé ce qui permet d’actualiser la vue. On réalise cela en assignant un gestionnaire de clic sur le bouton btnNewList.

Lors du clic sur un des titres de liste, un Intent explicite lance l’activité *ShowListActivity* en lui transmettant le pseudo et la position de l’item cliqué (ie le numéro de la liste).

## ShowListActivity

### Création du Layout **activity\_show\_list.xml**

|  |  |
| --- | --- |
| **activity\_show\_list.xml** | |
| Conteneur des items | id : conteneurItems |
| Zone de texte création d’item | Id : edtNewItem |
| Bouton création d’item | Id : btnNewItem |
| RecyclerView | Id : myRecyclerViewItem |



### Fonctionnement de **ShowListActivity**

Au démarrage, *ShowListActivity* récupère le pseudo et la position de la liste à afficher via l’Intent explicite envoyé par l’activité *ChoixListActivity.*

Grâce à ces données, l’activité récupère la liste des items correspondant à la liste en question, le RecyclerView est appelé et les affiche (en rouge) en utilisant le modèle de vue défini dans : *liste\_item\_todo.xml* (checkbox + description).

L’utilisateur peut ajouter un item de façon analogue à l’ajout de liste sur la liste précédente.

Au clic sur un item, celui-ci se coche ou se décoche et les données sont actualisées dans les préférences partagées.

Le RecyclerView est appelé à chaque modification du ProfilListe pour actualiser la vue en conséquence.