

TP°4 : Persister les données

Exercice 1 : Utilisation d'un singleton

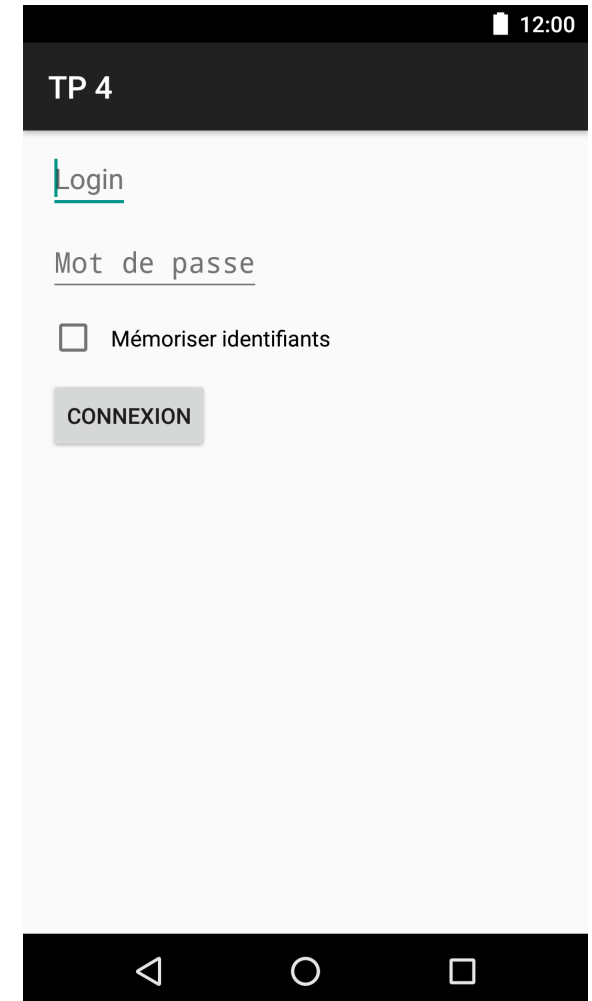
- Créer un Singleton UserSingleton permettant de garder en mémoire un objet User
- Dans l'Activity LoginActivity, au clic sur le bouton Connexion :
 - Créer un objet User avec les champs Login et Mot de passe
 - Affecter le User au UserSingleton
 - Lancer l'Activity HomeActivity
- Dans la HomeActivity, récupérer le User à partir du Singleton et mettre à jour la TextView R.id.login avec le login du User
- Tourner votre téléphone et vérifier qu'après rotation, la donnée est toujours affichée

Exercice 2 : Gestion singleton null

- Dans HomeActivity, avant d'affecter le login du User à la TextView, vérifier que le User du UserSingleton n'est pas null
- Si le User est null, relancer LoginActivity
- Aller dans les options développeurs du téléphone (Paramètres -> Options développeurs)
- En bas de la liste, appuyer sur Limite process arr-plan et sélectionner aucun. Avec cette option, toute app en background aura son process tué pour libérer de la mémoire. Cela permet de tester le comportement d'une app Garbage collectée rapidement.
- Retourner sur l'application TP_4 à partir des apps récentes et vérifier que vous retournez sur LoginActivity

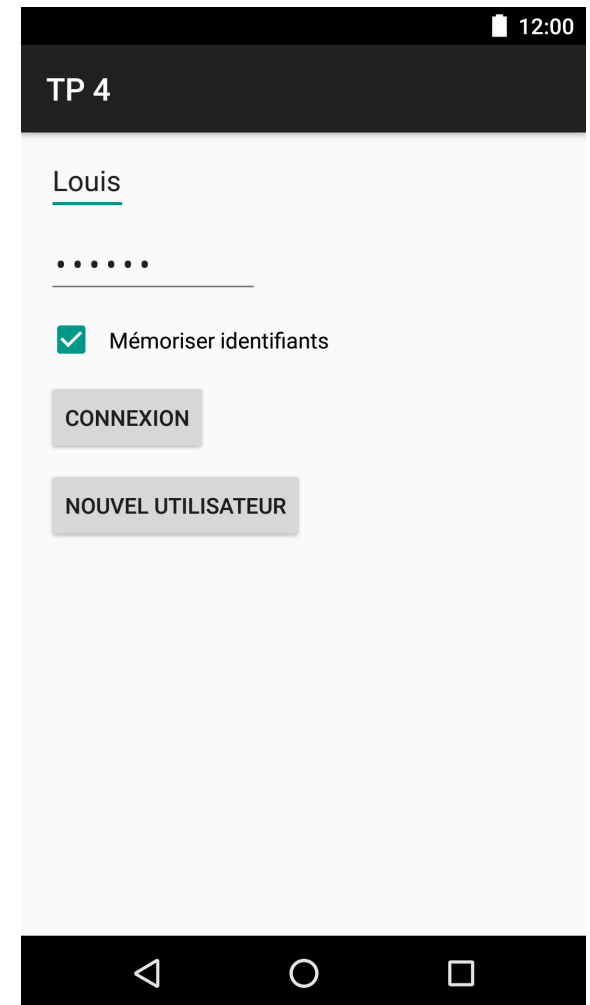
Exercice 3 : Utilisation des SharedPreferences

- Afficher la checkbox « Mémoriser identifiants » sur l'écran LoginActivity
- Modifier le listener du bouton Connexion pour ajouter :
 - Si la checkbox est cochée, mémoriser le login et le mot de passe dans des SharedPreferences en utilisant les clés PREFERENCE_LOGIN et PREFERENCE_PASSWORD avant de rediriger sur la HomeActivity
 - Si la checkbox n'est pas cochée, réinitialiser les éventuelles données stockées dans les clés
- Au lancement de l'Activity LoginActivity, utiliser les SharedPreferences pour pré-remplir les champs si les clés PREFERENCE_LOGIN et PREFERENCE_PASSWORD existent
- Vérifier que l'implémentation fonctionne et que les identifiants ont bien été sauvegardés en tuant le process de l'application à partir des apps récentes



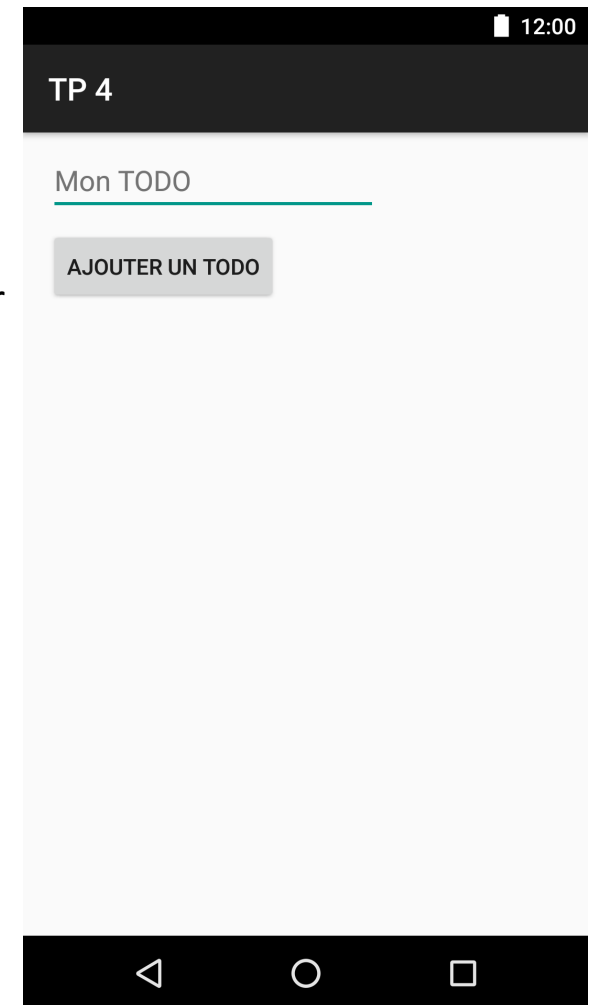
Exercice 4 : Utilisation des SavedInstanceStates

- Ajouter un bouton Nouvel utilisateur à l'écran LoginActivity et au clic lancer NewUserActivity
- Au clic sur le bouton Valider de NewUserActivity :
 - Créer un User à partir des champs
 - Passer ce User à l'Activity NewUserConfirmActivity
- Récupérer ce User dans le onCreate() de NewUserConfirmActivity à partir de l'intent
- Implémenter la méthode onSaveInstanceState de l'Activity pour sauvegarder le User
- Pré-remplir les TextViews dans la méthode onCreate() :
 - Si le bundle savedInstanceState est null, utiliser le User récupéré à partir de l'intent
 - Sinon, utiliser le User précédemment sauvegardé dans le onSaveInstanceState
- Tester l'implémentation en provoquant le Garbage Collect de l'application et en vérifiant que tous les champs sont bien remplis



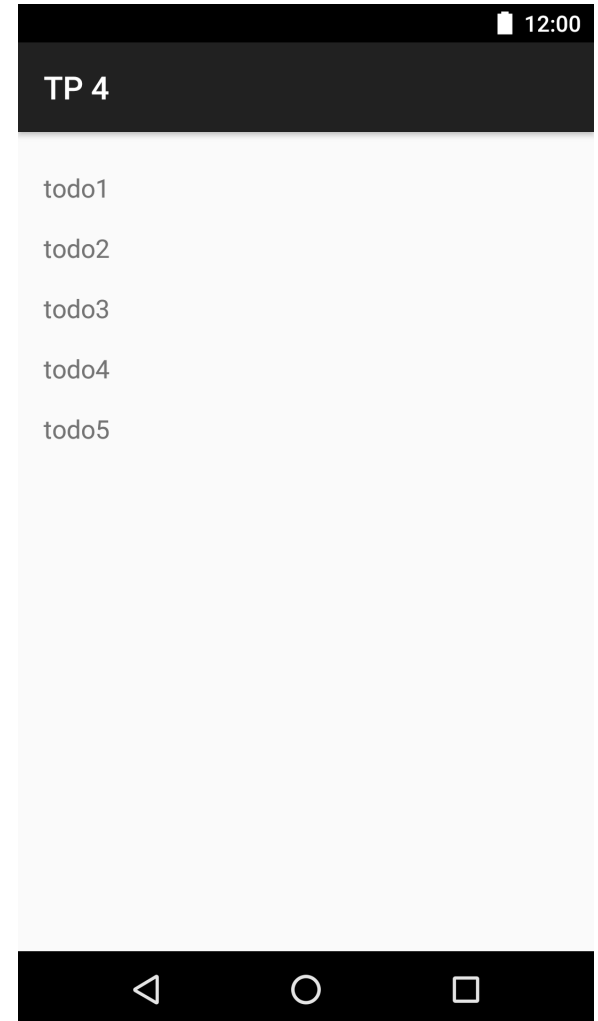
Exercice 5 : Ajouter une donnée dans une BDD

- Créer une base de données pour stocker des objets Todo (database/todo)
- Afficher le bouton Ajouter un Todo de la HomeActivity
- Au clic sur le bouton, lancer AddToDoActivity
- Dans l'Activity AddToDoActivity, au clic sur le bouton Ajouter un todo en exécutant une AsyncTask :
 - Passer le texte saisi dans l'EditText en paramètre de l'AsyncTask
 - Dans le onPreExecute(), afficher le ViewGroup R.id.loading du layout et rendre le bouton R.id.button disabled
 - Dans le doInBackground(), récupérer le texte passé en paramètre et l'insérer en base de données
 - Dans le onPostExecute, retourner sur HomeActivity



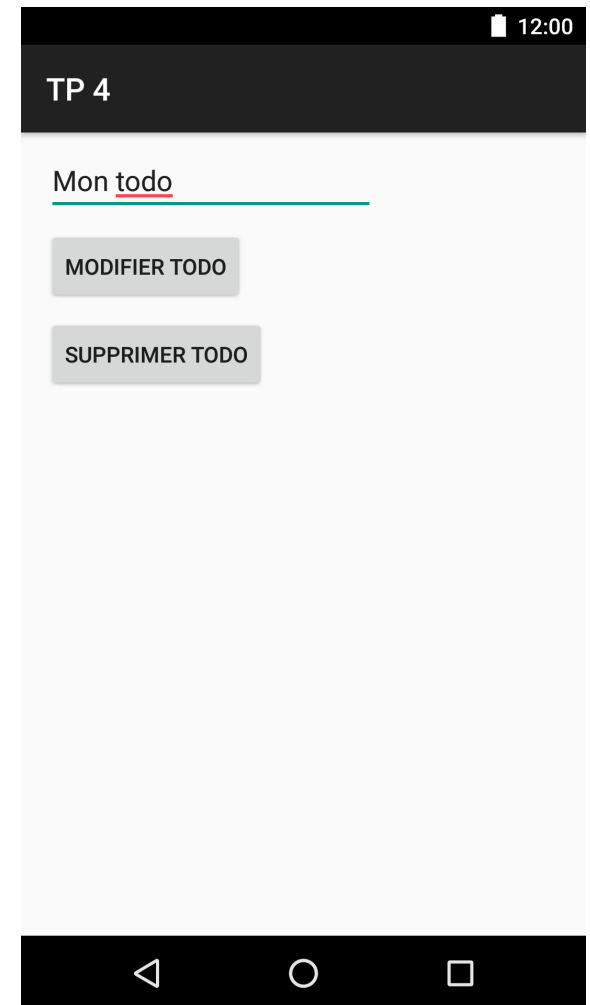
Exercice 6 : Lister les données d'une BDD

- Afficher le bouton Voir mes Todo sur la HomeActivity
- Au clic sur le bouton, lancer ToDoListActivity
- Dans l'Activity ToDoListActivity, récupérer et initialiser le RecyclerView du layout avec un LinearLayoutManager
- Lister les todos de la base de données et les affecter à la RecyclerView en exécutant une AsyncTask :
 - Dans le onPreExecute(), afficher la progressBar R.id.progress_bar
 - Dans le doInBackground(), récupérer la liste des todos et la retourner
 - Dans le onPostExecute, récupérer la liste retournée du doInBackground, initialiser l'adapter TodoAdapter et l'affecter à la RecyclerView



Exercice 7 : Modifier une donnée de la BDD

- Dans l'Activity ToDoListActivity, affecter un listener au clic d'un item de la liste en utilisant la méthode `setOnClickListener()` de l'adapter
- Au clic sur un item, lancer l'Activity EditToDoActivity en passant en intent le todo sélectionné
- Pré-remplir l'EditText avec le texte du todo passé en intent
- Au clic sur le bouton Modifier Todo, mettre à jour le todo et exécuter une AsyncTask :
 - Passer le todo mis à jour en paramètre de l'AsyncTask
 - Dans le `onPreExecute()`, afficher le ViewGroup `R.id.loading` du layout et rendre les boutons disabled
 - Dans le `doInBackground()`, récupérer le todo et le mettre à jour en base de données
 - Dans le `onPostExecute`, retourner sur HomeActivity
- Vérifier dans ToDoListActivity que votre todo a bien été mis à jour



Exercice 8 : Supprimer une donnée de la BDD

- Dans l'Activity EditToDoActivity, au clic sur le bouton Supprimer Todo, exécuter une AsyncTask :
 - Passer le todo à supprimer en paramètre de l'AsyncTask
 - Dans le onPreExecute(), afficher le ViewGroup R.id.loading du layout et rendre les boutons disabled
 - Dans le doInBackground(), récupérer le todo et le supprimer de la base de données
 - Dans le onPostExecute, retourner sur HomeActivity
- Vérifier dans ToDoListActivity que votre todo a bien été supprimé

