

1 Terminer le TP précédent

Prenez d'abord le temps de terminer le TP précédent si ce n'est pas déjà fait. Pour vous aider, voici quelques indices (pour une des solutions possibles, il en existe d'autres) :

- On crée l'adaptateur associé à la `ListView`. (`extends BaseAdapter implements ListAdapter`).
- La méthode `getView` de cet adaptateur est surchargée pour retourner une vue représentant la totalité d'un élément de liste.
- On utilise pour cela l'outil appelé `LayoutInflater` qui permet de régénérer une `View` à partir du layout construit en XML.
- C'est également dans cette méthode que l'on associe les adaptateurs des composants subordonnés.
- Une `Listener` (`implements AdapterView.OnItemClickListener`) est créé pour gérer la liste des étudiants.
- La méthode `onItemSelected` de cet objet est surchargée pour signaler au modèle tout changement opéré par l'utilisateur.

2 Gestion de la date.

On désire associer chaque liste d'attribution avec la date de la séance.

1. Ajouter une zone d'affichage de Date et un bouton côte à côte en haut de l'écran (Figure 1.

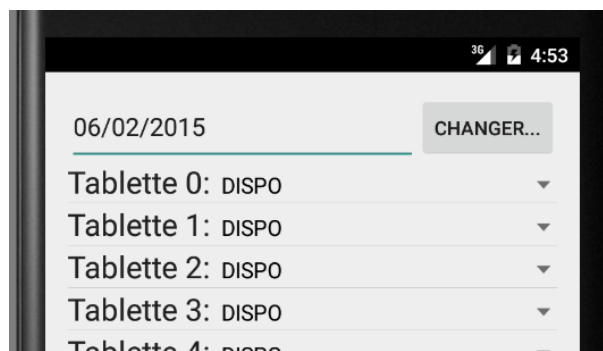


FIGURE 1 – Affichage de la date

2. La date sera initialisée (à l'a création) à la date du jour.
3. Le bouton permettra de modifier cette date (nous verrons comment plus tard).
4. Créer un second Layout qui comportera (Fig 2 :
 - Un `DatePicker` (Contrôle permettant de choisir une date).
 - Un bouton de validation (OK)
 - Un bouton d'annulation (ABANDON)
5. Inspirez-vous du cours pour permettre la modification de la date.
6. Intégrez la date dans le modèle.

3 Base de données

Vous pouvez constater que toutes les données sont perdues lorsque l'activité principale de l'application est détruite (par exemple par utilisation de la touche *retour* sur l'écran principal ; D'autre part, l'application ne mémorise que les réservations d'une seule date. Pour résoudre ces deux problèmes, nous allons utiliser une base de données comportant deux tables :

- table `jours`(colonne jour de type Date)
- table `resa`(colonne `id` (entière, clé primaire, colonne jour (date), colonne étudiant (chaîne)

3.1 Préparation

Écrivez selon le modèle vu en cours la classe `SQLConnect` qui génère la bases avec ses deux tables vides.



FIGURE 2 – Activité change date

3.2 Conception

Décrivez en fonction des événements suivant les opérations à réaliser sur la base.

1. Démarrage initial de l'application
2. Démarrage ultérieur de l'application
3. Fermeture de l'application
4. Changement de date sur l'activité (vers une nouvelle date)
5. Changement de date sur l'activité (vers une date déjà traitée)

3.3 Déduisez-en les opérations à réaliser sur la base de donnée