

Université Paris Diderot - Paris 7

Master 1 Informatique, Programmation de composants mobiles.

23 juin 2016

Durée : 2h Tous les documents papier autorisés. Documentation android autorisée. Les transparents du cours autorisés.

Toutes les autres ressources web (forums de développeurs etc.) interdites

Le sujet comporte 4 pages.

Consignes

La documentation se trouve localement dans

`/opt/android-sdk-linux_x86-r8/docs`

Si vous utilisez votre ordinateur portable alors c'était à vous d'installer la documentation localement sur votre machine avec Android Studio.

Après avoir fini chaque exercice compressez le répertoire source `VotreAppli/app/src/main` de votre application :

```
tar cvf VotreNom_VotrePrenom_Gr1Exoi.tar VotreAppli/app/src/main
gzip VotreNom_VotrePrenom_Gr1Exoi.tar
```

où *i* sera remplacé par le numéro de l'exercice et `VotreNom` c'est bien sûr votre nom. J'accepte aussi les archives zip. Par contre **je n'accepte pas les archives rar.**

Si vous travaillez sur une machine de l'ufr vous pouvez copier le fichier `VotreNom_VotrePrenom_Gr1Exoi.tar.gz` dans le répertoire `/ens/zielonka/ExamAndroidSession2`

Si vous utilisez votre propre ordinateur portable copiez le fichier `tar.gz` qui contient la solution de l'exercice dans `didel` (cours Android)

Travaux → ExamJuin2016

Avant de commencer les exercices récupérez depuis `didel`

Documents et liens → FichiersXMLExamRattrapage

ou à partir de répertoire

`/ens/zielonka/FichiersXMLExamRattrapage`

les fichiers xml.

Dans les exercices vous développerez deux applications.

Vos programmes seront testés avec un logiciel de détection de plagiat efficace.

Le dépôt sur `didel` est protégé donc c'est moi et vous même qui auront l'accès à votre programme et personne d'autre.

Si vous travaillez sur les machines de l'ufr c'est à vous de protéger votre répertoire de travail contre les visites non autorisées.

Application 1

Le but de l'application est de développer un programme qui aidera à apprendre des langues étrangères, ce sera juste un prototype très incomplet.

Exercice 1

L'application composée de deux activités.

L'activité principale c'est l'activité Choisir dont le layout vous trouverez dans `content_choisir.xml`.

L'activité Choisir est composé de : un Spinner, deux RadioButtons `fr` et `eng` groupés dans un `RadioGroup`, un `Button` bouton.

Le contenu de Spinner est alimenté à l'aide de deux tableaux dont le contenu se trouve dans le fichier ressources `array.xml` Les deux tableaux sont du même longueur, le premier tableau contient les mots anglais, la deuxième français de telle sorte que les mots à la même position dans le deux tableaux désignent la même couleur.

Votre activité doit initialement sélectionner (« checked ») le `RadioButton fr` et doit alimenter le Spinner avec les mots français.

A chaque moment quand l'utilisateur change la sélection de `RadioButton` le `Spinner` doit être alimenté avec le tableau qui correspond à la langue sélectionnée. Implémentez cette fonctionnalité.

Indication 1. Le plus commode est d'installer le listener

`RadioGroup.OnCheckedChangeListener`

sur le `RadioGroup` (méthode `setOnCheckedChangeListener`). Android active ce listener chaque fois quand l'utilisateur sélectionne un nouveau `RadioButton` appartenant à ce groupe. L'unique méthode du listener possède deux paramètres, le deuxième paramètre fournit le `id` du `RadioButton` sélectionné. (Comme toujours les ids de deux `RadioButtons` sont donnés dans le fichier `content_choisir.xml` donc il suffit de comparer le deuxième paramètre de la méthode avec `R.id.nom_id` approprié).

Indication 2. Pour alimenter un `Spinner` on procède comme pour alimenter une `ListView`, c'est-à-dire en utilisant un `ArrayAdapter`. Le fichier `ligne.xml` donne un layout pour une ligne (nécessaire pour construire un `ArrayAdapter`).

Exercice 2 Dans l'activité Choisir ajouter le code à exécuter quand l'utilisateur clique sur le bouton ENVOYER.

En appuyant sur le bouton l'utilisateur lance une nouvelle activité Comparer, le fichier layout `activity_comparer.xml`.

Dans `Intent` qui lance la nouvelle activité il faut passer deux extras :

- le premier extra, dont la clé est `mot`, est un entier – l'indice de l'élément qui est actuellement choisi dans `Spinner`,
- le deuxième extra, dont la clé est `lang`, est un `String` qui indique la langue différent de la langue du mot, donc si on envoie le mot français le deuxième extra aura la valeur "eng", si le mot envoyé est un mot anglais alors le deuxième extra sera "fr".

Le code de la nouvelle activité est à écrire dans l'exercice suivant, ici vous pouvez utiliser l'activité Comparer vide.

Exercice 3 Il reste à écrire le code de l'activité Comparer.

L'activité Comparer est composé d'un Spinner et d'un bouton VERIFIER Dans le Spinner on affiche les mots dans la langue indiquée par l'extra lang de l'intent qui a lancé l'activité. L'utilisateur choisit un mot dans le Spinner et appuie sur le bouton. Si ce mot est une traduction correcte du mot choisi auparavant dans l'activité Choisir, on affiche un toast BRAVO. Par contre si le mot sélectionné dans l'activité Comparer ne correspond pas au mot sélectionné dans l'activité Choisir on affichera le toast RATE.

Application 2

Les exercice suivants développent une nouvelle application dont le but est similaire que dans l'application précédente mais cette fois-ci nous utiliseront une base de donnée. Cette BD contient une seule table qui possède deux colonnes, fr et eng. Chaque ligne de la table donne un couple de mots, un mots anglais et un mot français. La base de données peut être implémentée par la classe DicoDB dont le code vous pouvez récupérer dans le fichier DicoDB.java. Cette classe est dérivée de SQLiteOpenHelper. Il ne vous reste qu'à implémenter la méthode onCreate de cette classe. (Vous n'êtes pas obligé d'utiliser le code java fourni, et si vous l'utilisez vous pouvez le modifier à votre guise.)

Exercice 4

L'activité principale de l'application contient trois boutons et un EditText. Le fichier layout : content_aprentissage.xml. Les trois boutons de l'activité principale servent à lancer trois nouvelles activités, respectivement : Ajouter, Afficher et Verifier.

Dans cet exercice on demande d'implémenter l'activité Ajouter lié au bouton AJOUTER de l'activité principale. Le but de l'activité Ajouter est d'alimenter la base de données. L'activité Ajouter est composée de deux EditText et d'un bouton, le fichier layout content_ajouter.xml. L'utilisateur met un couple de mots, un mot français l'autre anglais, dans les deux EditText et appuie sur le bouton. Les deux mots seront ajoutés dans la table de la BD. Après avoir ajouté le couple de mots dans la BD le contenu de deux EditText sera effacé.

Exercice 5 L'activité Afficher affiche le contenu de la BD dans une liste ListView de couples de mots, le fichier layout content_afficher.xml

Donc dans l'activité Afficher on lance la requête SELECT simple avec la condition WHERE vide et on affiche toutes les lignes récupérées par le curseur.

Le layout bd.xml c'est le layout d'une ligne d'affichage avec Adapter.

Rappel. Quand vous faites la requête SELECT sur un table il ne faut pas oublié la colonne _id qui est exigée par le curseur, donc il y aura trois colonnes : deux colonnes pour les mots anglais et français et la troisième rowid as _id

Indication. Comme Adapter de la liste on peut utiliser SimpleCursorAdapter et ignorer le fait que le constructeur est périmé (deprecated). Ou utiliser le curseur pour faire un parcours du résultat de SELECT, remplir un tableau ou une liste et utiliser ArrayAdapter.

Exercice 6 Le troisième bouton de l'activité principale est accompagné d'un EditText L'utilisateur entre un mot en français dans EditText, appuie sur le bouton et lance la dernière activité auxiliaire Verifier. Le mot fournit par l'utilisateur est envoyé comme un extra dans l'intent qui lance l'activité Verifier. Le fichier layout de l'activité Verifier : content_verifier.xml

L'activité Verifier est composée de : un TextView, un EditText et un bouton.

Dans TextView on affichera le mot français envoyé dans intent.

Ensuite l'utilisateur entre un mot anglais dans EditText, appuie sur le bouton et l'activité interroge la BD pour vérifier si le mot anglais correspond bien au mot français.

Si c'est le cas on affiche un toast BRAVO sinon un toast RATE.