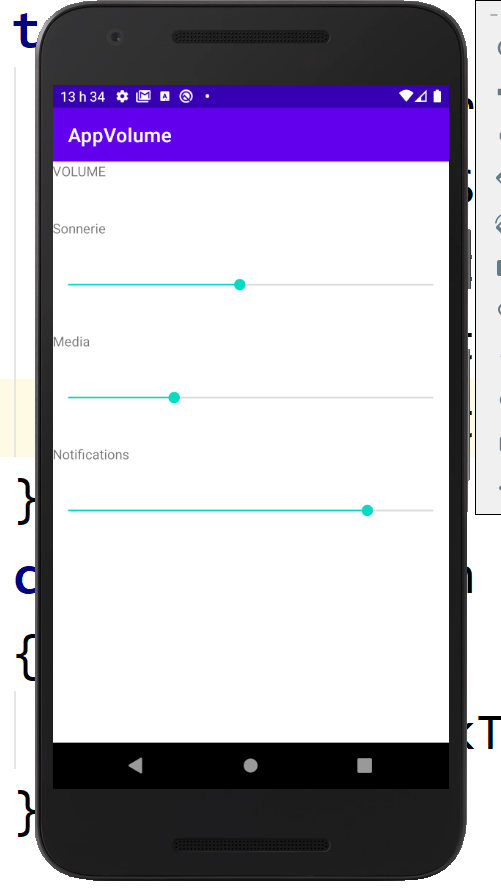
# Annexe 3C – Exercices ( Sérialisation, stockage partagé / importance du Buffer )

# Exercice 1

Créez une activité contenant 3 seekBars représentant des options de volume sur un téléphone :



À l'aide du processus de sérialisation, faites en sorte qu'on puisse retrouver les mêmes valeurs pour les SeekBars quand on redémarre l'app suite à une fermeture du bouton back, par exemple.

Est-ce qu'un SeekBar est sérialisable ?

# Exercice 2

Nous avons vu comment stocker des informations dans des fichiers situés en stockage interne. On peut également lire / écrire dans des fichiers se situant dans le stockage partagé de l’appareil, soit dans les dossiers Music, Images ou Téléchargements. La procédure cependant demande qu’on permette d’accéder à ces dossiers.

Utiliser le projet LireFichierMp3 présent dans LEA

But du projet : lire un fichier présent dans le dossier Musique pour prouver l’importance du Buffer.

1. Placer le fichier mp3 ( grand fichier ) dans le dossier storage/self/primary/Music de l’émulateur / de votre téléphone à l’aide du Device File Explorer ou le déposant directement sur l’émulateur. ( FERMER L'ÉMULATEUR APRÈS AVOIR AJOUTER LE .MP3 ET LE REDÉMARRER )
2. Regarder le [processus de permission](https://developer.android.com/training/data-storage/shared/media#request-permissions):

Scoped Storage / stockage étendu

* Sous le principe du stockage étendu ( scoped storage ) , plus besoin de demander des permissions dans les dossiers permis ( download, music… )
* On utilise le stockage étendu par opposition au stockage interne, visible que par l’app qu’on développe et que nous avons utilisé dans les annexes précédentes

1. Gestion de l’événement
2. Processus startActivityForResult, pourquoi ? Quel est le boomerang ?ouvre le fichier, ramène le fichier dans notre programme
3. Codez la méthode qui va lire un à un les octets du fichier mp3 et calculer le temps que ça prend avec ou sans tampon.

Quels sont les bons flux de données à utiliser ?FileInputStream et BufferedInputStream

Comment récupérer l’heure système ? System.*currentTimeMillis*();

Temps de lecture du fichier ( sans Buffer ) : 39500 millisec

Temps de lecture du fichier ( avec Buffer ) : 467 millisec