

DAA – Labo 04

Tâches asynchrones et Coroutines

Galerie d'images

19/12/2023

Courbat Guillaume

Grunder Alice

1. Introduction

Nous avons principalement suivi les slides du cours et la documentation de kotlin pour faire ce laboratoire.

2. Conception du squelette de l'Activité

Notre code consiste d'une seule *Activité* muni d'une *RecyclerView* avec un *GridLayoutManager* et un *spanCount* de 3 et d'une *MaterialToolbar*. Nous avons un *Adapter* pour la *RecyclerView* et un *Worker* pour gerer le cache. Finalement nous avons une data class *Item* pour gerer les images de la galerie.

3. Mise en place de l'Adapter et des Coroutines

Notre *Adapter* est conçu pour recevoir une liste d'*Item* et une *CoroutineScope*. Nous lui donnons *lifecycleScope* dans la *MainActivity*. Ce scope est utilisé lorsque la vue est *Bind*, c'est à dire quand la vue doit être affichée. Le but étant d'appeler des fonctions *suspend* travaillant sur des *Contexts* différents.

Réponses aux questions :

1. Lorsqu'une vue est recyclée dans une *RecyclerView*, la méthode *onViewRecycled* est appelée pour cette vue. On l'override pour inclure l'interruption de l'éventuelle coroutine lié à cette vue.
2. Nous donnons le scope à utiliser à notre *Adapter* dans notre *MainActivity*. Du coup on peut interrompre toute les coroutine "enfant" dans le *onDestroy* de *MainActivity*, ce qui a pour effet d'interrompre celles de la *RecyclerView*.
3. Oui, *Dispatcher.IO* est spécialisé pour les téléchargements (opération *IO* comme son nom l'indique). Nous utilisons *Dispatcher.Default* pour, par exemple, décoder l'image une fois avoir obtenu son *ByteMap*
4. On ajoute un *onClickListner* au *ViewHolder* (il y a un exemple dans le code)

4. Nettoyage automatique du cache

Le cache se nettoie toute les 15 minutes grâce à un *Worker* qui est appelé toutes les 15 minutes. La fonction de nettoyage est aussi appelée lors du clic sur le bouton du menu de la bare d'outil.

Réponses aux questions :

1. On crée une fonction *onCacheCleared* dans notre *MainActivity* qui rafraichi la galerie. Dans notre *CleanCacheWorker* on appel cette fonction.
2. En ajoutant un tag avec *addTag* et en donnant un nom unique au *worker* avec *enqueueUniquePeriodicWork* on s'assure que la tâche n'est pas enregistrée plusieurs fois.