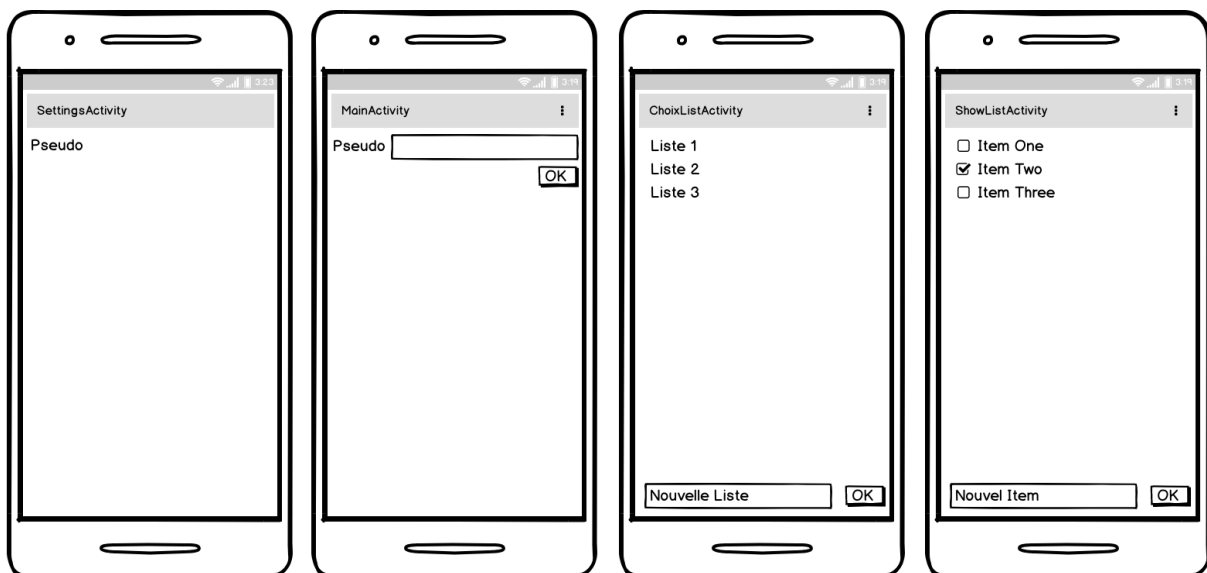


Électif PMR - Séquence 2

Objectifs

- Présentation du protocole HTTP, entêtes HTTP, requêtes GET & POST, chaînes de requêtes
 - Cf. RFC HTTP : <https://tools.ietf.org/html/rfc2616>
- API REST & bonnes pratiques
 - <https://blog.octo.com/designer-une-api-rest/>
 - <https://blog.ineat-conseil.fr/2013/01/restful-authentication/>
- API todo-api, utilitaire postman, requêtes PUT & DELETE
- Consommation d'une API Rest depuis Android dans votre application TODO
- Intent implicites

Storyboard



Travail demandé

Nous souhaitons améliorer l'application développée à la séquence 1 en récupérant les listes et les items des utilisateurs depuis une API Rest. Pour réaliser les requêtes HTTP, vous pourrez utiliser au choix :

- Des asyncTasks et l'objet `URLConnection`
 - Vous pourrez utiliser le code sur moodle présent dans le fichier `verifReseau.txt`
- La librairie Retrofit
- La librairie Volley

Vous utiliserez l'API todo-api, dont les sources sont disponibles sur moodle, et qui est disponible à l'adresse : <http://tomnab.fr/todo-api/>

La documentation de cette API est disponible à l'adresse :

<https://documenter.getpostman.com/view/375205/S1TYVGta?version=latest>

Voici les modifications à mettre en oeuvre dans vos précédentes activités :

- SettingsActivity : sauvegarder et permettre l'édition de l'url de base de l'API
- MainActivity :
 - Vérifier l'accès au réseau, et activer le bouton OK si le terminal utilisé peut accéder au réseau
 - Ajouter un champ mot de passe
 - Lors du clic sur le bouton OK, demander une identification auprès de l'API en lui envoyant pseudo/mot de passe
 - Récupérer le token d'identification sous forme d'un hash et le stocker dans les préférences de l'application
 - Vous renverrez ce token à chaque requête à l'API, dans un header "hash: <token>", ou dans une chaîne de requête
- ChoixListActivity : Récupérer les listes d'items associés à l'utilisateur connecté, et les afficher dans l'activité.
 - Lors du clic sur une des listes, passer à l'activité suivante en lui fournissant l'identifiant de la liste sélectionnée.
- ShowListActivity : Récupérer les items associés à la liste sélectionnée et les afficher dans l'activité.
 - Permettre de cocher/décocher chaque item,
 - Permettre l'ajout d'un nouvel item dans cette liste

Améliorations facultatives :

- Permettre l'ajout d'une nouvelle liste pour l'utilisateur connecté
- Connecter automatiquement l'utilisateur sans passer par l'activité de connexion en stockant pseudo/passe dans les préférences; ajouter un menu "déconnexion" dans toutes les activités, qui fera revenir à l'activité de connexion
- Améliorer les items des listes pour leur associer des liens, permettre de cliquer sur un item pour afficher le lien dans un navigateur au choix de l'utilisateur à l'aide d'un intent implicite..

Organisation

- Développement en kotlin
- Vous pouvez réaliser ce travail en binômes (éventuellement, à cheval sur les deux groupes de TD, pas forcément le même binôme que pour la séquence 1)
- Vous rendrez ce TP avant le début de la séquence 3, ainsi qu'un CR au format pdf faisant un bilan du travail réalisé (introduction, analyse, conclusion, perspectives, bibliographie...)
- Les dépôts se feront sur github, dans une branche différente de celle utilisée pour la séquence 1
- Plagiat entre les groupes **non autorisé**