

TP n°2

Application de petites annonces

Présentation

L'application à réaliser est une application de petites annonces. Les fonctionnalités de base sont les suivantes :

- Écran d'accueil avec boutons pour lancer les divers écrans de l'application (à faire évoluer plus tard)
- Visualisation d'une annonce (prise au hasard pour commencer)
- Saisie et dépôt d'une annonce
- Liste des annonces avec, pour chaque annonce, la possibilité de voir le détail et de contacter le vendeur (email/tél.)

On pourra ajouter ensuite :

- Avoir un profil local pour ne pas saisir son email/tél. à chaque dépôt d'annonce
- Sauvegarder localement des annonces sur l'appareil
- Prendre une photo avec l'appareil pour illustrer une annonce lors du dépôt

Divers détails seront précisés au fil de ce document.

Démarrage

Créer une Activity qui, pour commencer, servira à lancer les diverses parties de l'application. On pourra commencer avec 4 boutons comme sur le schéma ci-contre.

Créer un Intent pour que le clic sur "Voir annonce" emmène l'utilisateur sur une autre Activity qui permet de visualiser une annonce.

Modèle d'une annonce

Créer une classe Java qui modélise une annonce. Pour commencer, une annonce sera composée de :

- id : l'identifiant de l'annonce (une chaîne de caractères)
- titre
- description
- prix : un entier
- pseudo : le pseudo du déposant
- emailContact
- telContact
- ville
- cp : le code postal de la ville
- images : une liste d'URLs des images illustrant l'annonce. La liste peut être vide.
- date : la date de dépôt de l'annonce



Visualisation d'une annonce

1- Visualisation sans requête et sans image

Créer une Activity pour visualiser une annonce, par exemple comme sur le schéma ci-contre. La zone grisée représente la place qui sera prise par l'image plus tard.

Vous êtes libre de disposer les éléments de l'annonce comme vous le voulez.

2- Requête HTTP et Mock API

L'API de gestion des annonces sera mise en ligne ultérieurement.

Pour pallier à son absence, une « mock API » sera utilisée. Il s'agit d'un fichier JSON qui contient des données au format qui sera utilisé par l'API. La réponse de l'API au format JSON est constituée de :

- success : valeur true/false
- response : les données transmises. Contient le message d'erreur si success est false.

On pourra utiliser les URLs suivantes pour le développement en cours.

Réponse de type erreur :

<https://ensweb.users.info.unicaen.fr/android-api/mock-api/erreur.json>

Réponse avec les données d'une annonce :

<https://ensweb.users.info.unicaen.fr/android-api/mock-api/completeAd.json>

Intégrer la bibliothèque Volley ou OkHttp (au choix mais je recommande OkHttp) au projet et interroger la mock API pour obtenir les informations sur une annonce et l'afficher.

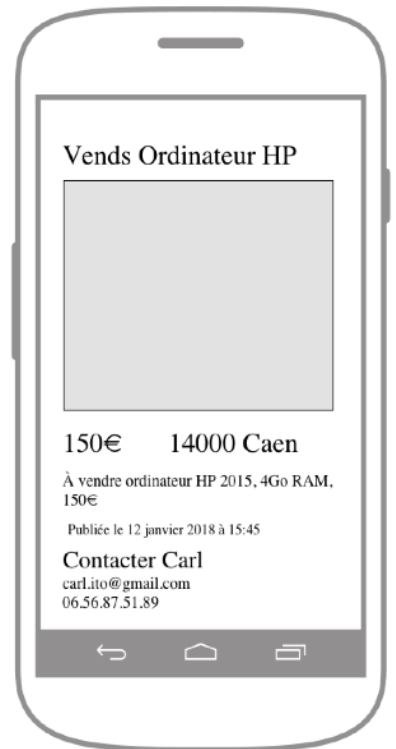
3- Affichage d'une image par défaut

La réponse completeAd.json contient une liste d'images vide. On voudrait alors qu'une image par défaut soit affichée, comme sur le schéma ci-contre.

Choisir une image par défaut et l'ajouter au projet dans le dossier res/ contenant les images. Attention, pour les images avec Android **le format PNG est recommandé.**

Intégrer la bibliothèque **Picasso** qui permet de facilement gérer l'affichage des images : <http://square.github.io/picasso/>

Utiliser Picasso pour afficher l'image par défaut dans la cas où la liste des images de l'annonce est vide.



4- Affichage d'une image de l'annonce

Pour obtenir une annonce avec une liste d'images, utiliser la mock API à l'URL suivante : <https://ensweb.users.info.unicaen.fr/android-api/mock-api/completeAdWithImages.json>

Toujours en utilisant Picasso, choisir au hasard une des images illustrant l'annonce pour l'afficher.

Pour les plus rapides

Gérer avec un *slider* l'affichage des images sur le détail d'une annonce (seulement s'il y en a plusieurs évidemment).

5- Contacter l'annonceur

Pour la vue de détails d'une annonce, ajouter la possibilité de contacter l'annonceur par email ou par téléphone en utilisant les Intent implicites.

Attention : pour que l'application ne plante pas lorsqu'on veut envoyer un email, il faut vérifier que l'appareil est capable de faire cette action.

Note : vous pouvez configurer l'application de mail sur l'émulateur.

Liste d'annonces

Créer une Activity qui utilise le RecyclerView pour visualiser une liste d'annonces

Vous êtes libre de disposer les éléments de l'annonce comme vous le voulez.

Mock API pour la liste d'annonces

L'API de gestion des annonces sera mise en ligne ultérieurement.

La « mock API » a été complétée pour pouvoir avoir une liste d'annonces. La propriété response contient alors une liste d'objets de type annonce au format JSON.

URL liste d'annonces: <https://ensweb.users.info.unicaen.fr/android-api/mock-api/liste.json>

Note : les données ont été générées automatiquement, les URLs des images sont des images hébergées chez Flickr. Les images sont pour l'instant au format JPG même si le PNG sera le format à préférer.

Ajouter la gestion du clic sur un item de la liste pour afficher le détail de l'annonce cliquée. Penser à réutiliser l'activité de visualisation du détail d'une annonce.

Les données de chaque annonce de la liste sont disponibles à l'URL (remplacer IDANNONCE par l'identifiant de l'annonce) :

<https://ensweb.users.info.unicaen.fr/android-api/mock-api/details/IDANNONCE.json>

Pour les plus rapides

Gérer l'affichage d'une image miniature dans chaque item de la liste des annonces, avec la possibilité de zoomer sur l'image.

Saisie du profil et enregistrement en Preferences

Réaliser une interface permettant à l'utilisateur de saisir son pseudo, son adresse email et son n° de téléphone et de les enregistrer avec les Preferences. Ces données seront alors utilisées plus tard lors du dépôt d'une annonce pour éviter de les saisir à chaque fois.

Mise en place de l'API

Mettre en place l'utilisation de l'API REST fournie et adapter les diverses parties de l'application.

Documentation de l'API <https://ensweb.users.info.unicaen.fr/isn/android/API-v1-documentation.pdf>

Dépôt d'une annonce

Créer une Activity pour pouvoir saisir une annonce. Le formulaire ne doit **pas** inclure le pseudo, email et téléphone qui sont enregistrés dans les Preferences de l'application et seront automatiquement repris pour l'envoi des données POST. Pour commencer vous pouvez coder en dur ces 3 données dans l'Activity mais ensuite ces données doivent venir des preferences.

La réponse du serveur doit servir pour afficher un message d'erreur ou de confirmation, et en cas de réussite il faut afficher l'annonce déposée.

Attention : Volley semble ne pas fonctionner comme sa documentation l'indique pour envoyer des requêtes de type JSONObject en méthode POST.

Utiliser OkHttp dans ce cas, ou si vous utilisez Volley alors effectuer une StringRequest en mode POST. La réponse du serveur est alors obtenue comme chaîne de caractères qui peut servir à instancier directement un JSONObject

Utilisation de Parcelable

Modifier la façon dont l'application passe de la liste des annonces aux détails d'une annonce donnée : implémenter l'interface Parcelable pour la classe qui modélise les annonces. Utiliser les extras de la classe Intent et pour passer directement l'objet annonce dont on veut afficher le détail à l'Activity qui affichera l'annonce. Utiliser cette méthode doit permettre d'éviter une nouvelle requête HTTP pour obtenir les détails d'une annonce.

Mise en place de menus

Mettre en place une App Bar qui est basée sur Toolbar. Les items de menu doivent être amenés à permettre la suppression de l'écran d'accueil avec les boutons. Utiliser les icônes adéquats pour les items du menu. Une fois le tout mis en place et testé, supprimer l'écran d'accueil avec les boutons et le remplacer par la liste des annonces.

Optionnel : mettre en place un *NavigationDrawer* pour y mettre en menu certaines actions (déterminer lesquelles) de l'application.

Pour les rapides

Mettre en oeuvre les éléments graphiques de l'application et personnaliser le « look'n'feel » et les interfaces utilisateur pour respecter les recommandations du Material Design.