Utopic Village

Spécifications techniques

2012

Guillaume

Exod Creation

11/06/2012

# Sommaire

Table des matières

[Sommaire 1](#_Toc328145127)

[Activités 2](#_Toc328145128)

[Afficher une vue : 2](#_Toc328145129)

[Asynchrone 3](#_Toc328145130)

[Parsing de données 3](#_Toc328145131)

[Adapter 3](#_Toc328145132)

[Application 3](#_Toc328145133)

[Stockage 3](#_Toc328145134)

[Overlay 3](#_Toc328145135)

[Thread et service 4](#_Toc328145136)

[Transaction 4](#_Toc328145137)

# Activités

Ce sont les contrôleurs de l’application, elles servent à afficher la vue et les informations qu’elle contient.

La hiérarchie est la suivante :

* MasterActivity, qui contient les méthodes globales à toute l’activité :
  + Affichage du spinner
  + Gestion des évènements entrants (appel, coupure,…)
  + Gestion du démarrage d’une activité
  + Gestion de la navigation
  + Encapsulation de la vue (c’est cette méthode qui va permettre d’emboité les vues provenant des activités filles)
* HeaderActivity (optionnel)
  + Affichage de la barre d’option
  + Affichage de la barre de recherche
  + Va permettre d’afficher et d’initialiser les informations personnelles dans la barre du haut
  + Possède aussi une méthode permettant d’arrêter le service de géolocalisation lorsque cette activité est détruite
* TabMenuActivity (optionnel)
  + Va permettre l’affichage du menu principal
  + Le constructeur permet de mettre en avant l’onglet courant
  + Les autres méthodes sont les écouteurs d’évènement
* OtherActivity
  + Ce sont les activités spécifiques (pour leur fonctionnement voir les commentaires)

## Afficher une vue :

Pour une nouvelle vue, il suffit de créer son layout via un xml, de faire étendre votre nouvelle activité de MasterActivity si vous ne souhaité rien n’y ajouter, HeaderActivity si vous souhaitez juste ajouter l’en-tête, ou TabMenuActivity si vous souhaitez y ajouter le menu (déconseillé).

Ensuite vous n’avez plus qu’à appeler le constructeur parent grâce à super.oncreate(savedInstanceState,xxx), ou xxx est la référence vers votre layout.

# Asynchrone

Le package asynchrone, c’est un package contenant un ensemble de classe dérivant d’Asynctask. Elles permettent de faire des requêtes webService asynchrones.

Elles se composent généralement de trois méthodes : doInBackground, onPreExecute, onPostExecute (pour le déclanchement de c’est méthodes voir la documentation de l’AsyncTask).

onPostExecute : Va appeler la méthode de l’activité qui permet de masquer la vue et d’afficher à la place le spinner.

doInBackground : Procède au traitement de la requête et au parsing des informations

onPreExecute : Va appeler la méthode de l’activité qui permet de supprimer le spinner et de réafficher la vue (en lui passant des paramètres au besoin)

## Parsing de données

Les classes asynchrones font fréquemment appel à la classe de parsing de donnée. Cette classe permet de crée un User ou une Help à partir d’un objet JSON. Ce traitement est assuré par une méthode static de la class ParsingUtil

## Mise en erreur

Lorsque le serveur ne réponse pas ou bien renvoie une erreur, la méthode de connexion préviens l’application qui va ensuite se charger de passer sur la vue YourAskingHelp et affiche un message d’erreur dans une alerte.

# Adapter

Il existe deux adapters actuellement, un pour afficher les informations sur les utilisateurs et un pour afficher les informations sur les demande d’aide

# Application

La classe UtopicVillageApplication est une classe un peu spéciale, elle contient l’ensemble des éléments de l’application. Elle est en fait une classe dérivée d’Application.

## Stockage

Les instances des objets sont stocké dans la classe de stokage de UtopicVillageApplication, c’est objets deviennent donc de faite commun a toute l’application et sont donc persistants.

Il y a plusieurs type d’objet qui sont stocké : Hashtable (voir documentation java ) des User et des Help.

Les classe de ces deux derniers type sont dans le package entité.

# Overlay

Le package overlay défini les éléments utilisés pour la carte.

# Thread et service

C’est deux package sont utiliser pour la géolocalisation et pour la réception de notification. Le thread produit une requête toutes les minutes qui vérifie qu’il n’y a pas de nouvelles notifications pour l’utilisateur.

Le service quant à lui est un processus lié au contexte de l’application. Il va permettre de signaler à l’application tous changements d’état, ici il s’agit de la géolocalisation.

# Transaction

Le package transaction est anciennement le package qui servait à produire les requêtes. Il ne reste dans ce package que les requêtes pour trouver les aides ou vous êtes volontaires et celle ou vous êtes le participant.

# Coté serveur

Les webservices font appel à la base de données grâce à l’ORM doctrine. Ensuite l’objet est converti en array grâce à la méthode get\_object\_vars. Après avoir été converti en array il suffit de l’encoder grâce à la méthode json\_encode et de l’envoyer dans la réponse.