Utopic Village

Spécifications techniques

2012

Guillaume

Exod Creation

11/06/2012

# Sommaire

Table des matières

[Sommaire 1](#_Toc327275101)

[Scénarios Utilisateurs 2](#_Toc327275102)

[Scénarios de modération 2](#_Toc327275103)

[Scénario d’administration 2](#_Toc327275104)

[Scénarios d’animateur de réseau sociaux 3](#_Toc327275105)

# 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