

TPDEV – Projet
Réalisation d'une application Android
UPMCalendar
William Sergeant

Essentiellement responsable de la récupération et du parsing des évènements contenus dans le fichier .ics correspondant à un emploi du temps, j'ai également travaillé sur l'interface graphique.

La récupération des évènements se fait donc au travers d'URLs utilisées par l'application web déjà en place. Nous récupérons ainsi un fichier au format iCalendar que nous parsons afin de créer une collection d'évènements contenant des horaires, un lieu, une description, des répétitions, etc. Ces opérations ont ensuite été déplacées vers une tâche asynchrone par mon binôme, ce afin de ne pas bloquer l'affichage lors de la récupération.

La mise en cache des collections d'évènements ainsi créées se fait par le biais des *SharedPreferences*, elles seront ainsi chargées au prochain lancement de l'application.

Concernant la GUI, nous avons utilisé comme template de base un *Drawer*. Celui-ci a été repensé afin d'afficher des informations d'utilisation de base dans le volet tirable. La barre de menu contient des actions comme afficher les paramètres, aller directement au jour courant, changer le format d'affichage (1 jour / 3 jours / 7 jours). Un écran de sélection des masters à afficher sous forme liste *checkBox* apparaît en cliquant sur l'icône des paramètres.

Si nous n'utilisons qu'une seule vue pour afficher à la fois les écrans de calendrier et de paramètres, il serait bon à l'avenir d'effectuer une transition sur un modèle de fragments, ce qui serait à la fois plus propre et plus efficace en termes de ressources.

Un point intéressant du développement a été ici la nécessité de développer un *ArrayAdapter* personnalisé afin de répondre à nos besoins. Ainsi, une nouvelle classe fût créée afin de nous permettre d'afficher une liste de label et de maintenir un tableau d'états sous forme de booléens (affiché/non-affiché).

Concrètement la difficulté majeure de l'interface a été de réaliser que l'API *dayView* / *WeekView* des Agendas Android n'étaient pas disponibles dans le SDK. Ayant commencé à travailler avec une vue mensuelle qui elle est intégrée, nous avons dû chercher une solution alternative en cours de projet. Ceci nous empêche en partie de respecter la consigne « faire usage d'au moins une API Android ».

