
IFT1155 – Projet – 6 février 2019**« Flux RSS »****Mohamed Lokbani**

Équipes : le travail peut-être fait en binôme, dans ce cas vous ne remettez qu'un travail par équipe.**Remise :** la remise doit-être effectuée par voie électronique.**Conseils :** n'attendez pas le dernier jour avant la remise pour engager votre travail. Vous n'aurez pas le temps nécessaire pour le réaliser.

But : ce projet a pour but de vous faire pratiquer les différentes notions apprises pour réaliser une application Android.**Énoncé :** un site web d'informations est constamment mis à jour en fonction des informations du jour. Au lieu de visiter plusieurs fois par jour le site web en question pour avoir les dernières nouvelles, on aimerait bien disposer d'un moyen plus simple pour être informé. Une solution consiste à regrouper ces informations dans un ensemble et diffuser cet ensemble à travers un mécanisme simple et pratique. Ceci va nous permettre de nous tenir au courant des mises à jour qui nous intéresse.

Le moyen simple pour diffuser ces mises à jour est l'utilisation d'un agrégateur (flux) RSS. Un flux RSS n'est qu'un format de fichier obéissant à certaines règles syntaxiques. Ce flux est généré le plus souvent de manière automatique dès la mise à jour du site web.

Le producteur du flux décide du contenu. Il contient généralement un titre, une courte ou une longue description, une ou plusieurs images, des liens hypertextes. Le lien renvoie, le plus souvent, vers la version longue de l'article sur le site web en question.

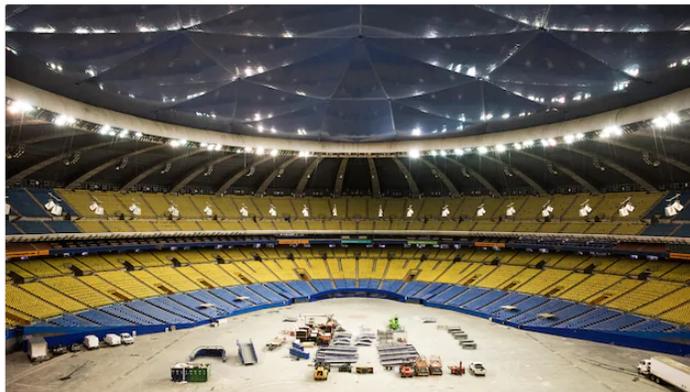
Voici un exemple d'un flux RSS d'un article de Radio-Canada sur le toit du stade olympique. L'article contient un titre, la date de publication, une courte description, une image et un lien vers la version longue de l'article.

```
<item>
  <title>
    Un nouveau toit pour le stade olympique, mais pas avant 2024
  </title>
  <link>
    http://ici.radio-canada.ca/nouvelle/1151200/toit-stade-olympique-montreal-hiver-toile-dechirures-remplacement
  </link>
  <guid isPermaLink="true">
    http://ici.radio-canada.ca/nouvelle/1151200/toit-stade-olympique-montreal-hiver-toile-dechirures-remplacement
  </guid>
  <pubDate>Wed, 06 Feb 2019 11:38:55 GMT</pubDate>
  <description>
    <![CDATA[
      <p>Il faudra attendre près de deux années supplémentaires pour découvrir le nouveau toit du stade olympique de Montréal.</p>
    ]]>
  </description>
  <enclosure type="image/jpeg" length="0" url="https://images.radio-canada.ca/q_auto,w_635,h_357/v1/ici-info/16x9/stade-olympique-montreal-interieur-gradins.jpg"/>
</item>
```

Voici comment l'article apparaît dans un flux RSS :

Un nouveau toit pour le stade olympique, mais pas avant 2024

Radio-Canada | Nouvelles 5h // keep unread // hide



Il faudra attendre près de deux années supplémentaires pour découvrir le nouveau toit du stade olympique de Montréal.

VISIT WEBSITE

Si on clique sur le lien proposé, on est redirigé vers le site web de Radio-Canada afin d'avoir accès à la version longue de l'article.

On vous demande de réaliser une application Android qui va permettre de gérer un ensemble de flux RSS et lire leurs contenus.

Il est important que l'application soit extrêmement simple à utiliser. On ne veut pas que l'utilisateur passe 10 minutes à faire 999 clics pour arriver à faire ce qu'il veut. L'application doit lui permettre de le faire rapidement et simplement.

Exemple d'application : « feedly »

https://play.google.com/store/apps/details?id=com.devhd.feedly&hl=en_CA

Description de l'application : nous allons commencer par décrire les différentes fonctionnalités de l'application dans une représentation globale. Plus tard dans l'énoncé, nous allons fixer un calendrier pour la remise de ces fonctionnalités. Nous allons aussi proposer quelques fonctionnalités optionnelles.

Gestion des flux : voici les quelques fonctionnalités en rapport avec la gestion des flux.

- Fournir une interface permettant de s'abonner à un ou plusieurs flux. L'utilisateur dispose d'un lien RSS qu'il veut suivre. Il doit l'incorporer dans le lecteur de flux.
- Fournir une interface permettant de se désabonner d'un ou de plusieurs flux. L'utilisateur ne veut plus suivre un ou plusieurs flux. Il veut se désabonner.
- Fournir un outil de recherche dans le tas de messages reçus. La recherche peut-être faite sur le titre ou la description.

- Fournir un moyen pour masquer un message lu. Ceci va permettre d'avoir une vue globale des messages en attente de lecture.
- Fournir la possibilité de séparer la file des flux. Par exemple, un utilisateur s'abonne au flux de Radio-Canada et à celui de La Presse, on veut séparer les deux. Le contenu est séparé. **(Bonus)**
- Fournir la possibilité de regrouper les flux en catégories. On peut regrouper les flux « Radio-Canada » et « La Presse » dans la même catégorie « Information ». On peut regrouper des flux en rapport avec le sport dans la catégorie « Sport », etc. **(Bonus)**

Gestion de la configuration : il faut préserver la configuration de l'utilisateur, la liste des flux, et leurs contenus dans un endroit en dehors de l'application. Le but est de séparer les données de l'interface. S'il y a une mise à jour de l'interface, on ne veut pas que l'utilisateur soit forcé de fournir de nouveau la liste des flux, de reproduire les différentes catégories, etc.

Accès : l'accès à l'application doit-être sécurisé avec un nom d'utilisateur (login) et un mot de passe sous la forme d'un pavé numérique. C'est au premier lancement de l'application qu'on demande de compléter cette configuration. L'utilisateur peut décider plus tard de désactiver cette option dans le menu de configuration de l'application.

Bdd : vous allez utiliser une base de données (SQLite) pour manipuler toutes les informations relatives à cette application, y compris le log des transactions.

Sauvegarde : la base de données sera sauvegardée dans le nuage¹. Vous allez préserver la configuration de l'utilisateur. La sauvegarde doit-être cryptée. Quand l'utilisateur installe l'application sur un autre appareil, on peut lui proposer de télécharger cette configuration pour éviter de repartir de zéro. **(Bonus)**

Langues : l'application doit-être proposée au moins dans les deux langues (FR/EN). Vous pouvez ajouter d'autres langues si vous le souhaitez.

Appareils : l'application doit-être réalisée pour deux types d'appareils, Pixel 2 et Nexus 9.

Vues : l'application doit-être réalisée pour les deux vues, portrait et paysage. Le but ce n'est pas juste d'autoriser le changement de vue tout en gardant une représentation statique ! À quoi sert le changement de vue alors ? Voir une manière intelligente de profiter de l'espace supplémentaire. Nous allons limiter à 2 vues : une vue unique pour Pixel 2, et une vue unique pour Nexus 9. Si vous décidez de faire 4 vues distinctes, ça sera un bonus. Le bonus ne sera pas validé que si vraiment il y a **4 vues distinctes**. Un petit tour de passe-passe dans la représentation des vues ne donnera pas droit au bonus !

API : vous devez cibler **l'API du niveau 28**, qui incorpore les outils de Google. La version minimale est l'API 25 et la version maximale est l'API 28.

Il faudra laisser place à votre imagination et votre créativité pour la conception de ce travail.

1 Vous trouvez plus de détails sur cette page web: https://support.iro.umontreal.ca/doku.php?id=compte:page_web

Exigences

- TP01 : valider l'interface utilisateur

- réalisation de l'interface
- pour deux appareils
- pour deux orientations
- pour deux langues, dont le français
- valider l'accès à l'interface.
- les fonctions abonnement/désabonnement
- configuration

- TP02 : valider le projet au complet

- téléchargement des flux
- représentation des flux
- gestion de la configuration
- base de données
- journal des transactions
- recherche
- algorithmique complète (abonnement, désabonnement, etc.)

- Bonus

- séparation des flux/catégorisation des flux
- 4 vues distinctes
- sauvegarde cryptée dans le nuage et réinstallation sur un autre appareil

Remises : vous devez effectuer deux remises sur Studium aux dates suivantes

- lundi 18 mars 2019, pour la remise du TP01 dans le dossier de remise tp01.
- jeudi 18 avril 2019, pour la remise du TP02 dans le dossier de remise tp02.

Fichiers à remettre : pour chaque travail, vous devez remettre un rapport, une capture vidéo courte², une archive du projet au format « ZIP » le fichier Android Package (APK) et finalement la feuille de route³.

La section « Foire aux questions » sur le site web du cours⁴ contient plus de détails sur les informations à inclure dans un rapport.

2 Nous avons expliqué comment le faire en démo.

3 Le fichier est disponible sur le site web du projet.

4 Nous allons revenir sur le contenu d'un rapport dans une séance pratique. Pour plus de détails, voir cette page web:
<http://www.iro.umontreal.ca/~dift1155/cours/ift1155/communs/Autres/rapport.html>

Barèmes**- TP01 : 15 points**

Rapport : 3 points

Ajout des deux langues : 2 points

Validation sur deux appareils : 4 points

Accès à l'interface /Abonnement/Désabonnement/Configuration : 4 points

Avis global : 2 points

La présentation du travail réalisée **est obligatoire**. Elle va avoir lieu lors de la séance pratique du **vendredi 22 mars 2019 dès 18h30**.

- TP02 : 25 points

Rapport : 5 points

Téléchargement des flux : 4 points

Représentation des flux : 2 points

Gestion de la configuration : 2 points

Base de données : 4 points

Recherche : 4 points

Algorithmique complète (abonnement, désabonnement, etc.): 4 points

- Bonus : 3 points

- séparation des flux/catégorisation des flux

- 4 vues **vraiment** distinctes

- sauvegarde cryptée dans le nuage et réinstallation sur un autre appareil

Recommandations

Utiliser le gestionnaire de versions directement à partir d'Android Studio⁵.

Vos proches sont d'excellents bêta testeurs. Proposez-leur de tester votre application.

Mise à jour

25-01-2019 diffusion du projet en démo.

06-02-2019 diffusion de cet énoncé.

5 L'utilisation de svn a été décrite dans une séance de démo.