## Lyrics : Affichage des paroles

### Préambule

Il est difficile de résister à l’envie de fredonner lorsque l’on écoute ses chansons préférées mais passer à l’étape supérieure, chanter, est souvent un pallier difficile à franchir pour plusieurs raisons, notamment la non-connaissance des paroles. C’est pourquoi nous avons décidés d’inclure dans notre application cette formidable fonctionnalité qui permettra aux amateurs et aux professionnels de profiter de leurs chansons au maximum.

### Fonctionnement

Après de nombreuses recherches sur la possibilité de streamer les paroles d’une chanson, notre attention s’est portée sur le site LyricsWiki (http://lyrics.wikia.com/Main\_Page), projet collaboratif gratuit qui regroupe à l’heure actuelle les paroles de dizaine de milliers de chansons. LyricsWiki est aussi le seul site proposant une API qui permet d’interroger directement sa base de données sans avoir à payer un forfait ou de quelconques autres frais.

Cette API est extrêmement facile d’utilisation :

1. Une requête contenant le nom de l’artiste, le nom de la chanson ou les deux est envoyée. Cette requête prend simplement la forme d’une adresse url, à laquelle on peut ajouter 3 paramètres :
   * **artist** pour le nom de l’artiste,
   * **song** pour le nom de la chanson et enfin
   * **fmt** pour le format de retour avec au choix :
     + **text**: format texte classique sans aucun formatage,
     + **html :** formatage dédié aux navigateurs web,
     + **xml :** Séparation en balise, requiert alors un parser pour récupérer les informations,
     + **json :** envoi d’un objet json.
2. Une réponse est reçue contenant les informations relatives au résultat trouvé. Plusieurs cas sont alors à discerner :
   * La recherche porte uniquement sur un artiste, l’API retourne alors la liste des chansons disponibles pour cet artiste s’il existe.
   * La recherche porte uniquement sur une chanson, l’API ne retourne que la documentation.
   * La recherche porte sur un couple artiste/chanson : Si la chanson existe, un échantillon des paroles est affiché ainsi qu’un lien pour en afficher la totalité, sinon une page d’erreur est renvoyée.

[screen4.png] Exemples de requête et de réponses associées

Le principal défaut de cette API est qu’elle n’affiche qu’une portion de la totalité des paroles (environ 1/7ème) à cause de problèmes de licences. En effet les principales maisons de disques qui règnent sur l’industrie ont interdits la publication des paroles en intégralité à travers des applications pour la principale raison que cela ne leur permettait pas d’avoir un total contrôle de l’utilisation de ces paroles.

(<http://groups.google.com/group/lyricwiki-api/browse_thread/thread/733ccd919d654040?pli=1>)

La solution qui a alors été trouvée par les fondateurs de LyricsWiki est de proposer un échantillon des paroles accompagné d’un lien vers la page web contenant l’intégralité de celles-ci. Heureusement, à l’heure des smartphones en permanence connectés à internet, cet obstacle a été relativement aisé à franchir, chose que nous aurions difficilement pu accomplir quelques années auparavant.

### Implémentation

#### Partie graphique

La partie graphique est définie sous format xml dans le fichier de configuration main.xml. Sachant que la quantité d’information à afficher allait être importante, j’ai opté pour contenir toutes les parties graphiques de cette fonctionnalité dans un **ScrollView**, une classe fille de FrameLayout permettant un défilement vertical de son contenu.

Cette ScrollView est composée de deux TableLayout, permettant ainsi d’organiser le contenu sous forme de lignes et de colonnes. Ces deux TableLayout alterneront entre l’était **Visible** et l’état **Invisible** selon la page à afficher :

* Le premier TableLayout contient la **WebView,** une vue permettant l’affichage d’un navigateur web **dans** notre application sans ouvrir le navigateur par défaut du téléphone. Son état par défaut est invisible et il ne sera appelé que lorsque l’utilisateur décidera d’afficher les paroles sur le site de LyricsWiki. Cette vue est accompagnée de deux icônes pour revenir à la recherche de paroles et quitter la fonctionnalité.
* Le second TableLayout est un peu plus fourni, il comporte une **TextView** où s’afficheront les paroles, un Button permettant l’ouverture et l’affichage du premier TableLayout mentionné plus haut (en cachant celui-ci). La deuxième partie se compose des champs de recherche (Artiste et Chanson) avec le Button lançant cette recherche.

#### Partie exécutive

Au niveau du code java, tout se passe dans la classe **AAP.java**.

Lorsque l’utilisateur appuie sur l’icône d’affichage des paroles, une première méthode est appelée s’occupant principalement de changer l’état de visibilité des différentes pages de l’application, elle s’occupe aussi d’initialiser les valeurs des champs de recherche à partir des informations fournies par la musique en cours de lecture, permettant ainsi une première recherche automatique.

Cette méthode passe ensuite la main à la méthode principale de la fonctionnalité : **setLyricsTextResult(String artist, String song, boolean estRetour)**

L’artiste et le titre de la chanson sont logiquement passés en paramètre car c’est à partir de ces variables que sera construite l’url de la requête :

* On part de la partie fixe de l’url : http://lyrics.wikia.com/api.php?
* On ajoute l’artiste si le champ n’est pas vide.
* On ajoute le titre de la chanson si le champ n’est pas vide.
* On doit maintenant choisir le format de réponse :
  + HTML si seul le champ artiste a été entré par l’utilisateur, on transférera alors directement l’url à la WebView pour afficher l’ensemble des chansons de l’artiste. L’utilisateur pourra ensuite sélectionner le titre qu’il souhaite.
  + XML sinon, ce format avec un formatage très facile d’utilisation nous permettra de récupérer les informations qui nous intéressent.

Dans le cas d’utilisation où le format de réponse est HTML, la méthode **openBrowser()** est appelée, rendant visible la WebView en lui faisant ouvrir l’url de recherche de l’artiste.

Il est d’ailleurs nécessaire de décrire brièvement cette méthode :

* Il faut activer manuellement la prise en compte du JavaScript dans la WebView grâce à la méthode .getSettings().setJavaScriptEnabled(true). Le JavaScript est utilisé par LyricsWiki dans la zone d’affichage des paroles, rendant obligatoire son activation.
* Par défaut, le « click » d’un lien dans la WebView entrainera l’ouverture du navigateur web par défaut du téléphone, sortant ainsi du cadre de l’application, c’est pourquoi il est nécessaire de forcer l’ouverture d’un nouveau lien dans une WebView à l’aide de la méthode .setWebViewClient(new WebViewClient()).

Il est important de noter que pour pouvoir accéder à internet via son application il est obligatoire d’ajouter une ligne d’autorisation dans le fichier **AndroidManifest.xml**:

<uses-permission android:name=*"android.permission.INTERNET"*/>

[screen9.png]Liste des chansons d’un artiste

Dans l’autre cas on récupère les informations du fichier xml de réponse grâce à un parser. Selon les informations récupérées on affiche ou non le bouton ouvrant le navigateur web (appel de la méthode openBrowser() comme au-dessus)

[screen10.png]Récupération des informations du fichier xml

### Rendu visuel

Une fois le menu **Fonctionnalités** déroulé, il est possible de sélectionner l’icône en forme de bulle qui représente l’ouverture des paroles.

[screen3.png] Icône « Lyrics » en surbrillance

On arrive alors directement à la page de recherche. Dès le chargement de la page, les paroles seront recherchées pour la chanson et l’artiste en cours de lecture et si des paroles sont trouvées, l’échantillon fourni par l’API sera alors visible ainsi qu’un bouton permettant à l’utilisateur d’ouvrir le navigateur web directement sur la page des paroles en question.

[screen5.png] Arrivée sur la page d’affichage des paroles

[screen6.png] Ouverture du navigateur web

[screen7.png] Scrolling jusqu’à l’affichage des paroles dans le navigateur

[screen8.png] Icône permettant le retour à la recherche ou la fermeture de la fenêtre des paroles