



**Institut Universitaire de Technologie**

**Département Informatique**  
Site de Bourg-en-Bresse



android

## **Mini projet Application météo**

[lionel.buathier@univ-lyon1.fr](mailto:lionel.buathier@univ-lyon1.fr)

# Introduction

- ▶ On souhaite réaliser une application pour Smartphone qui permet d'informer l'utilisateur sur la météo d'une ville durant une semaine.
- ▶ Celle-ci sera basée sur le flux RSS de meteorologic
- ▶ Nous allons procéder par étapes :
  - 1. Charger le flux RSS
  - 2. Parser le flux et afficher les données « brutes » extraites dans une ListView
  - 3. Utiliser une classe spécifique pour les données, et les afficher
  - 4. Améliorer l'affichage en utilisant des pictogrammes
  - 5. Vérifier l'état de la connexion (wifi/3G)
  - 6. Afficher un écran d'accueil (splash screen) lors du lancement
  - 7. Ajouter un menu de gestion des préférences / favoris pour la ville à consulter
  - 8. Afficher la météo de plusieurs villes sur Google Maps
  - 9. Ajouter des news /alertes



# Qu'est ce qu'un flux RSS ?

- ▶ Le format « **RSS** » (*Really Simple Syndication*) permet de décrire de façon synthétique le contenu d'un site web, dans un fichier au format XML (ou json), afin de permettre son exploitation par des tiers.
- ▶ Le fichier RSS, appelé également **flux RSS**, **canal RSS** ou **fil RSS**, contenant les informations à diffuser, est maintenu à jour afin de constamment contenir les dernières informations à publier.
- ▶ Un fil RSS est un fichier contenant le titre de l'information, une courte description et un lien vers une page décrivant plus en détail l'information.

<http://www.commentcamarche.net/contents/1328-les-standards-rss>



# Le flux RSS de prévisions météo

- ▶ Le site de **openweathermap** permet de donner les prévisions à 7 jours pour toutes les communes de France.
- ▶ Deux liens sont disponibles (exemple pour Bourg) :

<http://openweathermap.org/forecast5>

Le flux par défaut est au format json.

Pour changer les unités : &units=metric

Ex de lien :

<http://api.openweathermap.org/data/2.5/forecast?q=Lyon,fr&units=metric&appid=...id..>



# 1. Connexion au serveur distant

- ▶ On s'intéressera au flux Json proposé par Info Climat et décrit ici :  
<http://openweathermap.org/forecast5>
- ▶ Mettre en place l'AsyncTask nécessaire pour la connexion et charger le flux RSS, comme dans le TP précédent.
- ▶ Attention, le flux étant parfois très volumineux, on ne peut pas le lire avec un seul 'readLine'. Dans ce cas, il peut être judicieux d'utiliser un StringBuilder pour concaténer la chaîne.

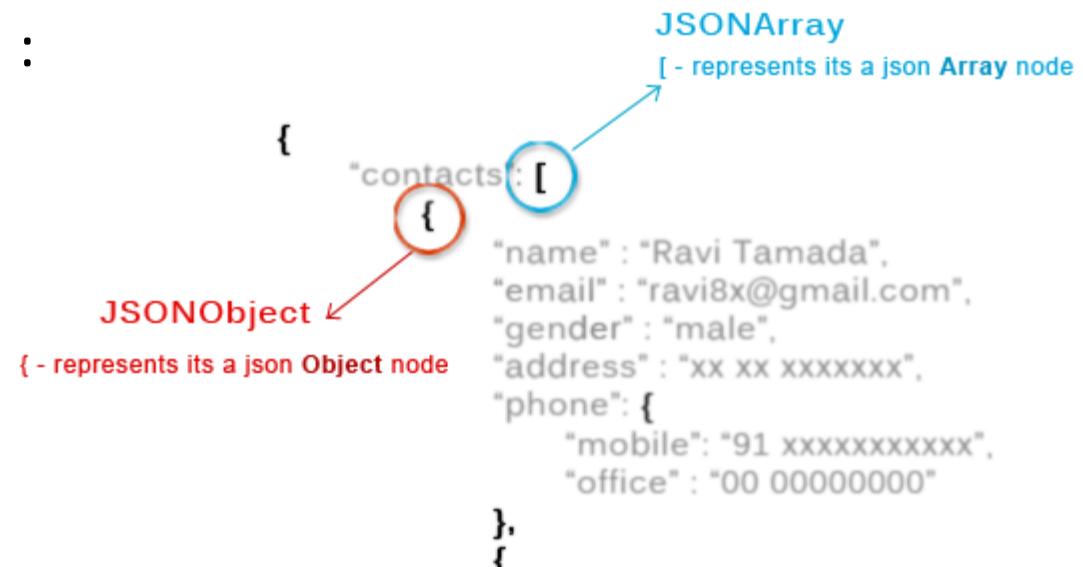


## 2. Parsage Json

- ▶ On peut parser le flux Json très simplement avec les classes :
  - JSONObject : lorsque le nœud commence par une accolade {
  - JSONArray : lorsque le nœud commence par un crochet [

JSON Structure                                   AndroidHive

- ▶ Exemple de fichier Json :



## 2. Parsage Json (suite)

- ▶ Dans un premier temps, vous afficherez les informations extraites dans le LogCat (avec la classe Log - cf. exemple ci-après).
- ▶ Ensuite, pour simplifier, vous pouvez commencer par mettre les données dans une chaîne et afficher son contenu dans une WebView avec la fonction loadData() ou dans un TextView.
- ▶ Ressources :

[http://www.tutorialspoint.com/android/android\\_json\\_parser.htm](http://www.tutorialspoint.com/android/android_json_parser.htm)

[http://androidexample.com/JSON Parsing -  
Android Example/index.php?view=article\\_descripion&aid=71&aaid=95#](http://androidexample.com/JSON_Parsing_-_Android_Example/index.php?view=article_descripion&aid=71&aaid=95#)

<http://stackoverflow.com/questions/9605913/how-to-parse-json-in-android>

<http://www.androidhive.info/2012/01/android-json-parsing-tutorial/>

Méthode récursive :

<http://developer.android.com/reference/android/util/JsonReader.html>



### 3. Ajout d'une classe de données météo

- ▶ Maintenant que vous savez "parser" le fichier, vous pouvez :
  - stocker les données dans une classe spécifique
    - ⇒ sans oublier de mettre en place les assesseurs et mutateurs (getters et setters)
  - Afficher ces données dans une ListView, au moyen de votre propre Adapter (cf. TP5)



## 3bis. Rafraîchir une ListView dans une AsyncTask

### Main Activity

```
... void onCreate(...)  
List list = new ArrayList<Revision>();  
MyAdapter adapter =  
    new MyAdapter(this, list);  
  
...void myOnclick(View v) {  
    new MyAsyncTask().execute(list, adpater);  
}
```

### MyAsyncTask

```
...doInBackground(Object... params)  
list = (ArrayList<Revision>) params[0];  
myAdapter = (MyAdapter)params[1];  
  
// Lecture et parsage du flux  
// mise à jour de list  
  
...onPostExecute(String s){  
    myAdapter.notifyDataSetChanged();
```



## 4. Amélioration de l'affichage

- ▶ Amélioration de l'affichage (icônes, polices, etc.)
- ▶ Vous pouvez par exemple, utiliser une ListView avec un affichage simplifié pour une vision de la semaine
- ▶ Ensuite, lorsqu'on clique sur un item, proposer une nouvelle vue avec un affichage plus complet (effet de zoom) avec des pictogrammes.
- ▶ La légende des termes et codes utilisés pour définir le prévisions et les noms des pictogrammes associés est décrite ici :  
<http://openweathermap.org/weather-conditions>
- ▶ Vous pouvez utiliser un jeu d'images embarqué dans l'application, voir une police de caractère spécifique.



## 4bis. Rafraîchir une ListView dans une AsyncTask

### Main Activity

```
... void onCreate(...)  
List list = new ArrayList<Revision>();  
MyAdapter adapter =  
    new MyAdapter(this, list);  
  
...void myOnclick(View v) {  
    new MyAsyncTask().execute(list, adpater);  
}  
}
```

### MyAsyncTask

```
...doInBackground(Object... params)  
list = (ArrayList<Revision>) params[0];  
myAdapter = (MyAdapter)params[1];  
  
// Lecture et parsage du flux  
// mise à jour de list  
  
...onPostExecute(String s){  
    myAdapter.notifyDataSetChanged();
```



## 5. Test de l'état de la connexion wifi ou 3G

- ▶ Ajouter un test de l'état de la connexion wifi ou 3G qui évite que l'application plante lors d'une requête http.
- ▶ S'il n'y a pas de connexion, on le signale à l'utilisateur au moyen d'un Toast.

<https://developer.android.com/training/monitoring-device-state/connectivity-monitoring.html>

- ▶ On peut également afficher les dernières informations enregistrées dans un fichier local. Dans ce cas, l'enregistrement se fait depuis la méthode `onStop()`, lorsque l'application est mise en arrière plan ou arrêtée. On peut utiliser un `WriteObject` /`ReadObject` pour enregistrer une `List` ou ses dérivés (`ArrayList`, etc.) puisqu'elle implémente l'interface `Serializable`. Il faut bien sûr que les attributs et objets la constituant soient également `Serializable`



## 6. Ecran d'accueil (splash screen)

- ▶ Il suffit de lancer un Intent avec un délai. Profitez en pour le lancer pendant le chargement de vos données, comme dans l'exemple suivant :

<http://www.androidhive.info/2013/07/how-to-implement-android-splash-screen-2/>



## 7. Gestion du menu et des préférences

- ▶ Ajouter un menu de gestion des préférences (PreferenceActivity) et les mémoriser dans les SharedPreferences :
  - La ville,
  - Une personnalisation de l'affichage (le nombre de jours à afficher ou autre)

Ressource :

<http://www.ace-art.fr/wordpress/2010/07/20/tutoriel-android-partie-5-les-ecrans-de-preferences/>



## 7. Gestion du menu et des préférences

- ▶ Ajouter un menu de gestion des préférences (PreferenceActivity) et les mémoriser dans les SharedPreferences :
  - La ville,
  - Une personnalisation de l'affichage (le nombre de jours à afficher ou autre)

Ressource :

<http://www.ace-art.fr/wordpress/2010/07/20/tutoriel-android-partie-5-les-ecrans-de-preferences/>



## 9. Affichage la météo de plusieurs villes sur Google Maps

- ▶ L'idée est de personnaliser les icônes

## 10. Ajouter des news /alertes

Proposer une nouvelle activité permettant d'afficher le contenu d'un flux RSS d'actualité de météo.



## Evaluation -Rendus

- ▶ L'évaluation se fera lors de la dernière séance sous forme d'une démonstration de votre application et sur la qualité de votre code (10 minutes)
- ▶ Vous déposerez sous spiral un zip portant vos noms, prénoms et votre groupe et contenant votre projet complet.
  - Exemple : PICASSO-Pablo-MATISSE-Henri-S4G2.zip

