# Annexe 8B – Exercices

# Exercice 1

En utilisant les données JSON sur le serveur suivant :

<https://api.jsonbin.io/v3/b/6374dac065b57a31e6b93755?meta=false>

Répondez aux questions suivantes en utilisant une requête GET de la librairie Volley :

1. Affichez le contenu du header du menu
2. Affichez le nombre d’éléments du tableau Items
3. Afficher le nombre d’éléments du tableau Items n’ayant pas d’attributs ( seulement null )
4. Afficher le nombre d’éléments du tableau Items n’ayant pas d’attribut «label »

# Exercice 2 – Gson

GSON et la magie de la réflexion Java ( Java Reflection )

Vous avez prouvé que vous êtes capables de récupérer des informations provenant d’un JSON mais ça peut demander de nombreux appels répétitifs dépendant de la profondeur de la structure; si on oublie quel niveau est un tableau et lequel est vide, on est pris avec des exceptions.

Il existe une manière plus simple de passer à travers les données JSON, soit la librairie GSON, Elle fonctionne comme suit :

On crée un modèle représentant les données qu’on espère trouver, sous forme de classes Java, ces classes auront les variables qui nous intéressent.

Ex : Soit le fichier JSON sur un serveur américain de météorologie :

**https://api.jsonbin.io/v3/b/6375d1732b3499323b022bbc?meta=false**

1. Examinons-le. Nous sommes intéressés aux prévisions journalières. On va donc créer un modèle de classes ressemblant à la structure du fichier JSON
2. Ajoutons les dépendances et permissions associées à Volley. Pour utiliser la librairie GSON on doit ajouter la dépendance suivante :

implementation **"com.google.code.gson:gson:2.8.5"**

1. Faire une requête StringRequest GET de Volley comme à l’habitude.
2. // séparer la grande String avec les morceaux de nos classes :

Gson gson = **new** GsonBuilder().create();  
ReponseMeteo wr = gson.fromJson(response, ReponseMeteo.**class** );

…on continue !