# Lecture d'un fichier JSON

(source Wikipédia)
**JSON** (*JavaScript Object Notation*) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. Créé par Douglas Crockford entre 2002 et 2005, il est décrit par la RFC 7159 de l'IETF.

Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci.

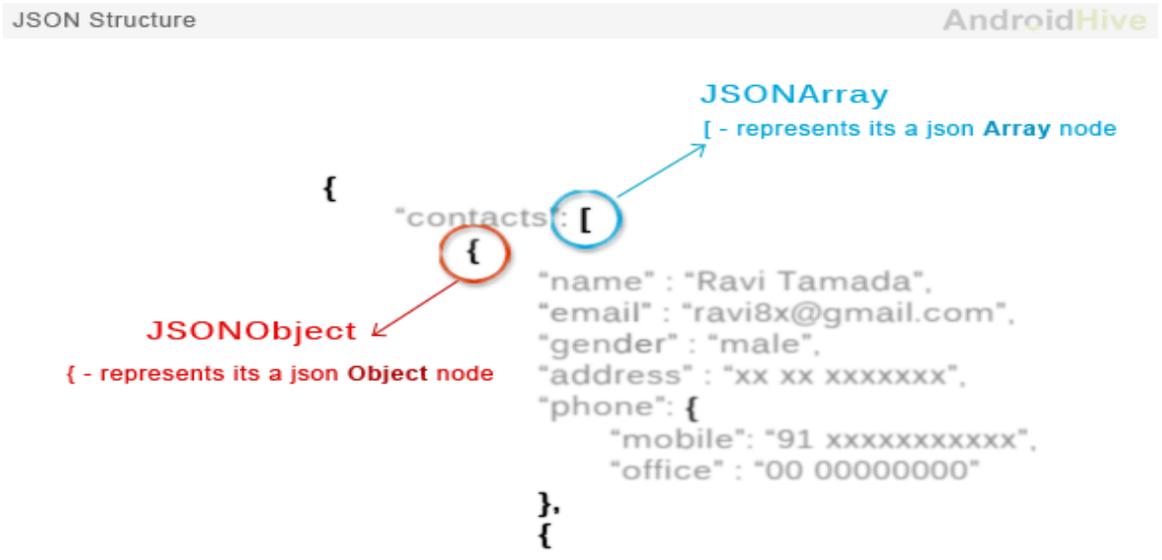
Un document JSON ne comprend que deux types d'éléments structurels :

•des ensembles de paires nom / valeur ;
•des listes ordonnées de valeurs.
Ces mêmes éléments représentent trois types de données :

•des objets ;
•des tableaux ;
•des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou null.

Exemple : soit le fichier JSON suivant

```
{
      "contacts": [
 {
  "id": 200,
  "name": "Ravi Tamada",
  "email": "ravi@gmail.com",
  "address": "xx-xx-xxxx,x - street, x - country",
  "gender" : "male",
  "phone": {
   "mobile": "+91 0000000000",
   "home": "00 000000",
   "office": "00 000000"
  }
 },
 {
  "id": 201,
  "name": "Johnny Depp",
  "email": "johnny_depp@gmail.com",
  "address": "xx-xx-xxxx,x - street, x - country",
  "gender" : "male",
  "phone": {
   "mobile": "+91 0000000000",
   "home": "00 000000",
   "office": "00 000000"
  }
 },
```




Si le nœud JSON commence par **[** , on utilise la méthode **getJSONArray(),** cette méthode va construire un ArrayList d'objet .
Si le nœud JSON commence par **{** , on utilise la méthode **getJSONObjet(),** cette méthode va retourner un objet JSON

```
try {
   // Create the root JSONObject from the JSON string.
   JSONObject  jsonRootObject = new JSONObject(myjsonstring);
   //Get the instance of JSONArray that contains JSONObjects
   JSONArray jsonArray = jsonRootObject.optJSONArray("contacts");
   //Log.i("tag1", "longueur "+jsonArray.length());
   //Iterate the jsonArray and print the info of JSONObjects
   for(int i=0; i < jsonArray.length(); i++){
      JSONObject jsonObject = jsonArray.getJSONObject(i);
      int id = jsonObject.getInt("id");
      String nom =jsonObject.getString("name");
      JSONObject phone = jsonObject.getJSONObject("phone");
      String mobile = phone.getString("mobile");
      Log.i("contacts", id +"  "+ nom+"  "+mobile);
      Contact c = new Contact(id,nom,mobile);
      lesContacts.add(c);

   }
}
catch (JSONException e) {e.printStackTrace();}
```

Ce code va permettre de récupérer l'id, le nom et le mobile de chaque contact.

Les exceptions représentent le mécanisme de gestion des erreurs intégré au langage Java. Il se compose d'objets représentant les erreurs et d'un ensemble de trois mots clés qui permettent de détecter et de traiter ces erreurs (try, catch et finally , à voir en 2e année)

Avant de parser un fichier JSON, vous devez lire le contenu du fichier texte et mettre le tout dans une chaîne de caractères, pour la création d'un objet JSONObject à partir d une chaîne de caractère

Avec le jdk 1.6 la méthode est la suivante : ( le JDK 1.8 simplifie la lecture mais les fonctionnalités ne sont pour le moment pas disponible sur androidStudio)

```
StringBuffer sb = new StringBuffer();
BufferedReader br = null;
try {
   br = new BufferedReader(new InputStreamReader(getAssets().open("contacts.json")));
   String temp;
   while ((temp = br.readLine()) != null)
      sb.append(temp);
} catch (IOException e) {
   e.printStackTrace();
} finally {
   try {
      br.close(); // stop reading
   } catch (IOException e) {
      e.printStackTrace();
   }
}
String myjsonstring = sb.toString();
```

Les exceptions représentent le mécanisme de gestion des erreurs intégré au langage Java. Il se compose d'objets représentant les erreurs et d'un ensemble de trois mots clés qui permettent de détecter et de traiter ces erreurs (try, catch et finally , à voir en 2e année)

**getAssets().open("contacts.json")** : cela suppose sur AndroidStudio que vous avez créé un répertoire assets dans lequel vous avez copié un fichier country.json
**new → folder → assets folder**

**Travail à faire :**
1 – créer un nouveau projet contacts
2 – créer un répertoire assets dans lequel vous déposerez le fichier : contacts.json
3 – créer une classe Contact : id → int , nom → String , mobile → String
Générer les constructeurs , les getters et setters ( alt + inser)
4 - Dans votre class MainActivity , créer une méthode getLesContacts qui retournera la liste des contacts obtenue suite à la lecture du fichier : contacts.json

On déclare une liste de contacts :

```
ArrayList<Contact> lesContacts = new ArrayList<Contact>();
```

Pour chaque Contact on récupère son id, nom et mobile, on crée un objet Contact que l'on ajoute à la liste des contacts

```
private ArrayList<Personne> getLesContacts(){}
```

```
public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {
    ListView mListView;
    ArrayList<Contact> lesContacts = new ArrayList<Contact>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lesContacts = getLesContacts();
        mListView = (ListView) findViewById(R.id.listView);
        ArrayAdapter<Contact> adapter = new ArrayAdapter<Contact>(MainActivity.this,
                android.R.layout.simple_list_item_1, lesContacts);
        mListView.setAdapter(adapter);
        mListView.setOnItemClickListener(this);
    }
    private ArrayList<Contact> getLesContacts(){
        //lecture du fichier contacts.json pour mettre chaque ligne dans une chaine de caractère
        StringBuffer sb = new StringBuffer();
        BufferedReader br = null;
        try {
            br = new BufferedReader(new InputStreamReader(getAssets().open("contacts.json")));
            String temp;
            while ((temp = br.readLine()) != null)
                sb.append(temp);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                br.close(); // stop reading
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        String myjsonstring = sb.toString();
        try {
            // Create the root JSONObject from the JSON string.
            JSONObject  jsonRootObject = new JSONObject(myjsonstring);
            //Get the instance of JSONArray that contains JSONObjects
            JSONArray jsonArray = jsonRootObject.optJSONArray("contacts");
             //Iterate the jsonArray and print the info of JSONObjects
            for(int i=0; i < jsonArray.length(); i++){
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                int id = jsonObject.getInt("id");
                String nom =jsonObject.getString("name");
                JSONObject phone = jsonObject.getJSONObject("phone");
                String mobile = phone.getString("mobile");
                Log.i("contacts", id +"  "+ nom+"  "+mobile);
                Contact c = new Contact(id,nom,mobile);
                lesContacts.add(c);
            }
        }
        catch (JSONException e) {e.printStackTrace();}
        return lesContacts;
    }
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText(getApplicationContext(),"Bonjour : " +
                lesContacts.get(position).getNom(), Toast.LENGTH_SHORT).show();
    }
}
```