

## 目次

### JSONを読み込み、パースする

1. Jsonとは
2. JSONをパースしてみよう
3. APIからJSON形式のデータを取得する

**JSON**（ジェイソン、JavaScript Object Notation）は軽量なデータ記述言語の1つである。構文はJavaScriptにおけるオブジェクトの表記法をベースとしているが、**JSON**はJavaScript専用のデータ形式では決してなく、様々なソフトウェアやプログラミング言語間におけるデータの受け渡しに使えるよう設計されている。  
（参照） [wikipedia](#)

JSON は、null、真偽値、数値、文字列、配列、オブジェクト（連想配列）のデータを、文字列で表現できる。配列とオブジェクト（連想配列）は、ネスト（入れ子）構造で表現することができる。

JSONには下記6種類の「値」が存在する。

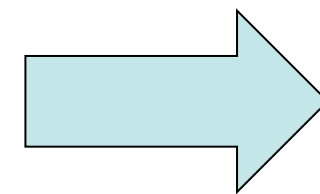
- 文字列値(string)：“abc”や“def”等、任意の情報をダブルクォートで囲んだ記述。
- 数値(number)：1や23.45等の数値。
- 真偽値(boolean)：true及びfalseという「yes/no的な意味合い」を表現する記述。
- ヌル値(null)：
- 配列値(array)：[]や[1]や[1,true,"hello"]等。
- オブジェクト値(object)：{}や{"foo":"bar"}や{"foo":"bar", "baz":null}。

＊シングルクォーテーション『`'`』でくくる事はできません。

# JSONをパースしてみよう

## JSONデータの文字列

```
{
  "name":"takeno",
  "email":"ggggg@gmail.com",
  "sex":1,
  "age":20,
  "address":"tokyo japan",
  "nationality":"JAPANESE",
  hobby:[
    "baseball",
    "music",
    "trip"
  ]
}
```



## Java側でのパース処理

//取得したJSONデータの文字列を代入しておく。

```
String responce_json = "{\"name\":\"takeno\",\"email\":\"ggggg@gmail.com\",
  \"sex\":1,\"age\":20,\"address\":\"tokyo japan\",\"nationality\":\"JAPANESE\",
  \"hobby\":[\"baseball\",\"music\",\"trip\"]}";
```

```
try {
  JSONObject jsonObj = new JSONObject(responce_json);
  JSONArray jsonHobbyArg = jsonObj.getJSONArray("hobby");
  String hobby1 = jsonHobbyArg.getString(0);
  String hobby2 = jsonHobbyArg.getString(1);
```

```
  String name = jsonObj .getString("name");
  String email = jsonObj .getString("email");
  int sex= jsonObj .getInt("sex");
  int age= jsonObj .getInt("age");
  String address = jsonObj .getString("address");
  v("hello_app",address+"::::");
```

```
}catch (JSONException e) {
  // TODO 自動生成された
  e.printStackTrace();
}
```

# APIからJSON形式のデータを取得する

## zip cloud

日本郵便のWebサイトで公開されている郵便番号データを再配信するサービス。

### <リクエスト例>

郵便番号で住所を取得するAPI

<http://zipcloud.ibsnet.co.jp/api/search?zipcode=1690072>

### <レスポンス例>

```
{
  "message": null,
  "results": [
    {
      "address1": "東京都",
      "address2": "新宿区",
      "address3": "大久保",
      "kana1": "トウキョウト",
      "kana2": "シンジ ュク",
      "kana3": "オオクボ ",
      "prefcode": "13",
      "zipcode": "1690072"
    }
  ],
  "status": 200
}
```

# APIからJSON形式のデータを取得する

## HeartRails Express

路線／駅名データ等の地理情報を、XML、JSON(P) 形式の 無料でご提供するAPIサービス

### <リクエスト例>

経度、緯度により最寄駅を取得

<http://express.heartrails.com/api/json?method=getStations&y=35.806625&x=139.160065>

# APIからJSON形式のデータを取得する

## <レスポンス例>

```
{
  "response":
    {"station":
      [
        {
          "x":139.164306,
          "next":"古里",
          "prev":"御嶽",
          "distance":"880m",
          "y":35.813764,
          "line":"JR青梅線",
          "postal":"1980102",
          "name":"川井",
          "prefecture":"東京都"
        },
        {
          "x":139.161549,
          "next":"御岳山",
          "prev":null,
          "distance":"1190m",
          "y":35.796032,
          "line":"御岳登山鉄道",
          "postal":"1980174",
          "name":"滝本",
          "prefecture":"東京都"}
      ]
    }
}
```

# APIからJSON形式のデータを取得する

## Google MAP API

### <リクエスト例>

経度、緯度を指定して、住所情報を取得する

<http://maps.googleapis.com/maps/api/geocode/json?latlng=35.806625,139.160065&language=ja&sensor=true>