

The JSON format was originally specified by Douglas Crockford.

JSON: JavaScript Object Notation.

JSON is a lightweight data-interchange format

JSON is "self-describing" and easy to understand

JSON uses JavaScript syntax, but the JSON format is text only.
Text can be read and used as a data format by any programming language.

JSON is a syntax for storing and exchanging data.

JSON is text, written with JavaScript object notation.

When exchanging data between a browser and a server, the data can only be text.

JSON is text, and we can convert any object into JSON, and send JSON to the server.

We can also convert any JSON received from the server into JavaScript objects.

JSONObject & JSONArray

In a JSON file , square bracket ([]) represents a JSON array

In a JSON file, curly bracket ({}) represents a JSON object

A JSON object contains a key that is just a string. Pairs of key/value make up a JSON object

Each key has a value that could be string , integer or double e.t.c

```
String jsonString = readJsonObjectFromSomewhere();
try {
    JSONObject json = new JSONObject(jsonString);
} catch (Exception e) {
    e.printStackTrace();
}
```

```
public void writeJSON() {
    JSONObject object = new JSONObject();
    try {
        object.put("name", "Jack Hack");
        object.put("score", new Integer(200));
        object.put("current", new Double(152.32));
        object.put("nickname", "Hacker");
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
}
```

```
{
  "employee": {
    "name": "sachin",
    "salary": 56000,
    "married": true
  }
}
```

["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

```
{ "Employee" :
  [
    {"id":"101","name":"Sonoo Jaiswal","salary":"50000"},
    {"id":"102","name":"Vimal Jaiswal","salary":"60000"}
  ]
}
```

```
{
  "sys":
  {
    "country":"GB",
    "sunrise":1381107633,
    "sunset":1381149604
  },
  "weather":[
    {
      "id":711,
      "main":"Smoke",
      "description":"smoke",
      "icon":"50n"
    }
  ],

  "main":
  {
    "temp":304.15,
    "pressure":1009,
  }
}
```

JSONObject Parsing methods:

Below we define some important methods of JSONObject parsing which are mainly used for parsing the data from JSONObject.

- 1. get(String name):** This method is used to get the value from JSONObject. It returns the value of object type. We pass the String type key and it returns the value of Object type if exists otherwise it throws JSONException.
- 2. getBoolean(String name):** This method is used to get the Boolean value from JSONObject. We pass the String type key and it returns the value of Boolean type if exists otherwise it throws JSONException.
- 3. getDouble(String name):** This method is used to get the double type value from JSONObject. We pass the String type key and it returns the value in double type if exists otherwise it throws JSONException.
- 4. getInt(String name):** This method is used to get the int type value from JSONObject. We pass the string type key and it returns the value in int type if exists otherwise it throws JSONException.
- 5. getJSONArray(String name):** This method is used to get the JSONArray type value. We pass the String type key and it returns JSONArray if exists otherwise it throws JSONException.
- 6. getJSONObject(String name):** This method is used to get the JSONObject type value. We pass the String type key and it returns the JSONObject value if exists otherwise it throws JSONException.
- 7. getLong(String name):** This method is used to get the long type value from JSONObject. We pass the String type key and it returns the value in long type if exists otherwise it throws JSONException.
- 8. getString(String name):** This method is used to get the String type value from JSONObject. We pass the String type key and it returns the value in String type if exists otherwise it throws JSONException.
- 9. length():** This method is used to get the number of name/value mappings in this object.
- 10. keys():** This method is used to get the iterator of String names in the Object.
- 11. opt(String name):** This method is used to get the value from JSONObject. It returns the value of Object type. We pass the String type key and it returns the value of Object type if exists otherwise it returns null.
- 12. optBoolean(String name):** This method is used to get the Boolean value from JSONObject. We pass the String type key and it returns the value of Boolean type if exists otherwise it returns false.
- 13. optDouble(String name):** This method is used to get the double type value from JSONObject. We pass the String type key and it returns the value in double type if exists otherwise it returns NaN.
- 14. optInt(String name):** This method is used to get the int type value from JSONObject. We pass the string type key and it returns the value in int type if exists otherwise it returns 0.

15. `optJSONArray(String name)`: This method is used to get the JSONArray type value from JSONObject. We pass the String type key and it returns JSONArray if exists otherwise it returns null.

16. `optJSONObject(String name)`: This method is used to get the other JSONObject type value from JSONObject. We pass the String type key and it returns the JSONObject value if exists otherwise it returns null.

17. `optLong(String name)`: This method is used to get the long type value from JSONObject. We pass the String type key and it returns the value in long type if exists otherwise it returns 0.

18. `optString(String name)`: This method is used to get the String type value from JSONObject. We pass the String type key and it returns the value in String type if exists otherwise it returns empty("") string.

JSONArray Parsing methods:

Below we define some important methods of JSONArray parsing which are mainly used for parsing the data from JSONArray.

1. `get(int index)`: This method is used to get the value from JSONArray. It returns the value of object type. We pass the index and it returns the value of object type if exist otherwise it throws JSONException.

2. `getBoolean(int index)`: This method is used to get the Boolean value from JSONArray. We pass the index and it returns the value of Boolean type if exists otherwise it throws JSONException.

3. `getDouble(int index)`: This method is used to get the double type value from JSONArray. We pass the index and it returns the value in double type if exists otherwise it throws JSONException.

4. `getInt(int index)`: This method is used to get the int type value from JSONArray. We pass the index and it returns the value in int type if exists otherwise it throws JSONException.

5. `getJSONArray(int index)`: This method is used to get the JSONArray type value. We pass the index and it returns JSONArray if exists otherwise it throws JSONException.

6. `getJSONObject(int index)`: This method is used to get the JSONObject type value. We pass the index and it returns the JSONObject value if exists otherwise it throws JSONException.

7. `getLong(int index)`: This method is used to get the long type value from JSONArray. We pass the index and it returns the value in long type if exists otherwise it throws JSONException.

8. `getString(int index)`: This method is used to get the String type value from JSONArray. We pass the index and it returns the value in String type if exists otherwise it throws JSONException.

9. `length()`: This method is used to get the number of values in this Array.

10. `opt(int index)`: This method is used to get the value from JSONArray. It returns the value of Object type. We pass the index and it returns the value at index of Object type if exists otherwise it returns null.

11. `optBoolean(int index)`: This method is used to get the Boolean value from JSONArray. We pass the index and it returns the value of Boolean type if exists otherwise it returns false.

12. `optDouble(int index)`: This method is used to get the double type value from JSONArray. We pass the index and it returns the value in double type if exists otherwise it returns NaN.

13. `optInt(int index)`: This method is used to get the int type value from JSONArray. We pass the index and it returns the value in int type if exists otherwise it returns 0.

14. `optJSONArray(int index)`: This method is used to get the other JSONArray type value from JSONArray. We pass the index and it returns JSONArray if exists otherwise it returns null.

15. `optJSONObject(int index)`: This method is used to get the JSONObject type value from JSONArray. We pass the index and it returns the JSONObject value if exists otherwise it returns null.

16. `optLong(int index)`: This method is used to get the long type value from JSONArray. We pass the index and it returns the value in long type if exists otherwise it returns 0.

17. `optString(int index)`: This method is used to get the String type value from JSONArray. We pass the index and it returns the value in String type if exists otherwise it returns empty(“”) string.

```
String JSON_STRING = "{\"employee\":{\"name\":\"Abhishek  
Saini\",\"salary\":65000}}";  
    String name, salary;  
  
try {  
    // get JSONObject from JSON file  
    JSONObject obj = new JSONObject(JSON_STRING);  
    // fetch JSONObject named employee  
    JSONObject employee = obj.getJSONObject("employee");  
    // get employee name and salary  
    name = employee.getString("name");  
    salary = employee.getString("salary");  
  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
}
```