## JSON是什么

* JavaScript Object Notation
* **JSON是一种传递对象的语法，对象可以是name/value对，数组和其他对象。**
* JAVAScript Object Notation是一种轻量级的数据交换格式
* 具有良好的可读和便于快速编写的特性。
* 业内主流技术为其提供了完整的解决方案（有点类似于正则表达式 ，获得了当今大部分语言的支持）
* JSON采用兼容性很高的文本格式，同时也具备类似于C语言体系的行为。 – Json.org
* JSON作为数据是目前网络中主流的数据传输格式之一，应用十分广泛，说是使用率达到99%一点也不勉强

### JSON主要部分

**JSON里面的数据是以一种键值对的方式存在 .**

**数据类型：**

**嵌套对象、数组、字符串、数字、布尔值或空值。**

**根据JSON的官方定义，键，加"",值，如果是字符串，就加""，其他不加。**

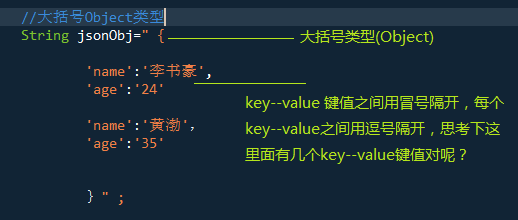
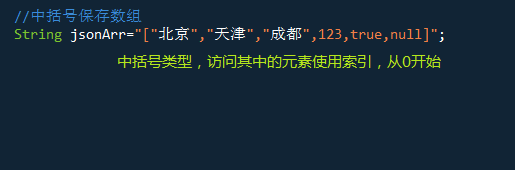
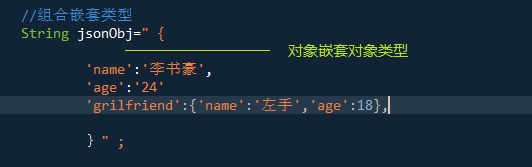
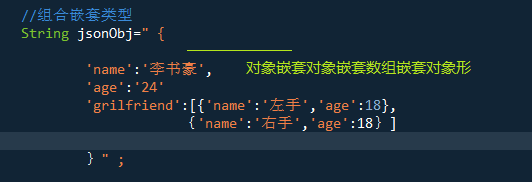
**花括弧，方括弧，冒号和逗号**  
花括弧表示一个“容器”  
方括号装载数组  
名称和值用冒号隔开  
数组元素通过逗号隔开

**JSON跟XML不一样，因为：**

XML文件格式复杂，比较占宽带，服务器端与客户端解析xml话费较多的资源和时间.  
XML里在元素的开始和结尾处有尖括号和标签名：JSON使用花括号，而且只在数据的开始和结束时使用。  
JSON更简练，毫无疑问更适合人类书写，也许也能让我们更快速的阅读。  
JSON可以在JavaScript里简单的传递到eval()方法里使用  
JSON里有数组{每个元素没有自己的名称}  
在XML里你可以对一个元素使用任意想要的名称，在JSON里你不能使用Javascript里的保留字

**XML方式：**  
取回一个XML文件  
循环它，从中提取值  
处理这些值，等　  
**JSON方式：**  
取回JSON字符串。  
‘eval’ JSON数据

### JSON基本语法与图例

* Object(对象类型)
  + 用{ }包含一系列无序的key–Value键值对表示，其中Key和Value之间用冒号分割，每个key-value之间用逗号分割。
  + 比如：
* Array(数组类型)   
  + 使用[ ]包含所有元素，每个元素用逗号分隔，元素可以是任意的值
  + 比如：
* 组合形   
  +   
    

### 组合类型的解析

问什么要解析JSON数据?

* 因为从服务器获取到的JSON数据都是二进制的,客户端无法直接使用,就需要将二进制的JSON数据转换成客户端可以直接使用的OC对象.
* 注意 : JSON形式的字符串客户端也无法直接使用.还是需要将其转换成OC对象.
* **序列化：** 在向服务器发送数据之前，将NSArry/NSDictionary等OC对象转换成二进制的过程.
* **反序列化：** 在从服务器接收到数据之后，将二进制数据转换成NSArray/NSDictionary等OC对象的过程.
* JSON解析也叫做JSON反序列化

｛[ {},{} ]｝–>思路：第一步看到大括号：JSONObject去解,创建好对应里面的属性的值；第二步看到中括号：JSONArray去解析 对应创建好对应里面的属性值；第三步中括号里面的{}，又是一个JSONObject。思路大致如此，强调注意的是，设置的属性值setXxx getXxx必须要和去获得值名称一致，不然会出错！！

Json数据：

 {

     'desc': 'OK',

     'status': 1000,

     'data': {

         'wendu': '20',

         'ganmao': '相对于今天将会出现大幅度降温，易发生感冒，请注意适当增加衣服，加强自我防护避免感冒。',

         'forecast': [

             {

                 'fengxiang': '北风',

                 'fengli': '3-4级',

                 'high': '高温 27℃',

                 'type': '中雨',

                 'low': '低温 19℃',

                 'date': '6日星期五'

             },

             {

                 'fengxiang': '北风',

                 'fengli': '微风级',

                 'high': '高温 23℃',

                 'type': '大雨',

                 'low': '低温 17℃',

                 'date': '7日星期六'

             },

             {

                 'fengxiang': '北风',

                 'fengli': '微风级',

                 'high': '高温 26℃',

                 'type': '小雨',

                 'low': '低温 17℃',

                 'date': '8日星期天'

             },

             {

                 'fengxiang': '南风',

                 'fengli': '微风级',

                 'high': '高温 27℃',

                 'type': '多云',

                 'low': '低温 15℃',

                 'date': '9日星期一'

             },

             {

                 'fengxiang': '南风',

                 'fengli': '微风级',

                 'high': '高温 29℃',

                 'type': '多云',

                 'low': '低温 16℃',

                 'date': '10日星期二'

             }

         ],

         'yesterday': {

             'fl': '微风',

             'fx': '北风',

             'high': '高温 33℃',

             'type': '阴',

             'low': '低温 22℃',

             'date': '5日星期四'

         },

         'aqi': '58',

         'city': '成都'

     }

 }

一:第一个大括号(JSONObject)｛”desc”: “status”: “data”｝

二：大括号里面有一个大括号(JSONObject)｛”wendu”: “20”, “ganmao”: “forecast”: “yesterday”: “aqi”: “city”: ｝

三：第二个大括号里面有两个对象 1.数组形（JSONArray） 2.对象形( JSONObject )

而数组形里面又套有数组的对象{}

Weather wea=new Weather();

// 首先看到的是一个{}所以用JSON Object来进行解析

// 获得外部的Weather

JSONObject obj = new JSONObject(json);

String desc = obj.getString("desc");

int status = obj.getInt("status");

wea.status=status;

wea.desc=desc;

// 获得内部Data的数据

JSONObject obj1 = obj.getJSONObject("data");

Data data=new Data();

data.wendu=obj1.getString("wendu");

data.ganmao=obj1.getString("ganmao");

data.aqi=obj1.getString("aqi");

data.city=obj1.getString("city");

wea.data=data;

List<Forecast> forecasts=new ArrayList<>();

// 获取forecast数组

JSONArray jArr = obj1.getJSONArray("forecast");

for (int i = 0; i < jArr.length(); i++) {

JSONObject obj2 = jArr.getJSONObject(i);

Forecast forecast=new Forecast();

forecast.date=obj2.getString("date");

forecast.fengxiang=obj2.getString("fengxiang");

forecast.high=obj2.getString("high");

forecast.low=obj2.getString("low");

forecast.fengli=obj2.getString("fengli");

forecast.type=obj2.getString("type");

forecasts.add(forecast);

}

data.forecast=forecasts;

JSONObject obj3 = obj1.getJSONObject("yesterday");

Yesterday yesterday=new Yesterday();

yesterday.fl=obj3.getString("fl");

yesterday.fx=obj3.getString("fx");

yesterday.high=obj3.getString("high");

yesterday.type=obj3.getString("type");

yesterday.low=obj3.getString("low");

yesterday.date=obj3.getString("date");

data.yesterday=yesterday;

// 输出字段

System.out.println(wea);

## 不用框架时将JavaBean转成JSON

### 导入开发包

* commons-io-2.0.1.jar
* commons-lang-2.5.jar
* commons-collections-3.1.jar
* commons-beanutils-1.7.0.jar
* ezmorph-1.0.3.jar
* json-lib-2.1-jdk15.jar

### 事例代码

package cn.itcast.javaee.js.bean2json;

import net.sf.json.JSONArray;

import java.util.\*;

/\*\*

\* 使用第三方工具，将JavaBean对象/List或Set或Map对象转成JSON

\* @author AdminTC

\*/

public class TestBean2Json {

private static void javabean2json() {

City city = new City(1,"广州");

JSONArray jSONArray = JSONArray.fromObject(city);

String jsonJAVA = jSONArray.toString();

System.out.println(jsonJAVA);

//[{"id":1,"name":"广州"}]

}

private static void list2json() {

List<City> cityList = new ArrayList<City>();

cityList.add(new City(1,"广州"));

cityList.add(new City(2,"珠海"));

JSONArray jSONArray = JSONArray.fromObject(cityList);

String jsonJAVA = jSONArray.toString();

System.out.println(jsonJAVA);

//[{"id":1,"name":"广州"},{"id":2,"name":"珠海"}]

}

private static void set2json() {

Set<City> citySet = new LinkedHashSet<City>();

citySet.add(new City(1,"广州"));

citySet.add(new City(2,"珠海"));

JSONArray jSONArray = JSONArray.fromObject(citySet);

String jsonJAVA = jSONArray.toString();

System.out.println(jsonJAVA);

//[{"id":1,"name":"广州"},{"id":2,"name":"珠海"}]

}

private static void javabeanlist2json() {

List<City> cityList = new ArrayList<City>();

cityList.add(new City(1,"中山"));

cityList.add(new City(2,"佛山"));

Province province = new Province(1,"广东",cityList);

JSONArray jSONArray = JSONArray.fromObject(province);

String jsonJAVA = jSONArray.toString();

System.out.println(jsonJAVA);

/\*

[

{

"id":1,

"name":"广东"

"cityList":[{"id":1,"name":"中山"},{"id":2,"name":"佛山"}],

}

]

\*/

}

private static void map2json() {

List<City> cityList = new ArrayList<City>();

cityList.add(new City(1,"中山"));

cityList.add(new City(2,"佛山"));

Map<String,Object> map = new LinkedHashMap<String,Object>();

map.put("total",cityList.size());//表示集合的长度

map.put("rows",cityList);//rows表示集合

JSONArray jSONArray = JSONArray.fromObject(map);

String jsonJAVA = jSONArray.toString();

System.out.println(jsonJAVA);

//[{"total":2,"rows":[{"id":1,"name":"中山"},{"id":2,"name":"佛山"}]}]

jsonJAVA = jsonJAVA.substring(1,jsonJAVA.length()-1);

System.out.println(jsonJAVA);

}

}

