**一、简介**

org.json是Java常用的Json解析工具，主要提供JSONObject和JSONArray类，现在就各个类的使用解释如下。

**二、准备**

1.使用org.json之前，我们应该先从该网址https://github.com/douglascrockford/JSON-java下载org.json源码，并将源码其加入到Eclipse中，即可调用。

2.查看相关的API文档，访问：https://github.com/douglascrockford/JSON-java。

**三、讲解：**

****1.JSONObject：****

是一个无序的键/值对集合。

它的表现形式是一个包裹在花括号的字符串，键和值之间使用冒号隔开，键值和键值之间使用逗号隔开。

内在形式是一个使用get()和opt()方法通过键来访问值，和使用put()方法通过键来添加或者替代值的对象。

值可以是任何这些类型：Boolean,JSONArray,JSONObject,Number和String，或者JOSONObject.NULL对象。

代码演示如下：

public static void jsonObjectTest() {

JSONObject jsonobj = new JSONObject("{'name':'xiazdong','age':20}");

String name = jsonobj.getString("name");

int age = jsonobj.getInt("age");

System.out.println("name = " + name + ",age = " + age);

}

注：JSONObject有很多optXXX方法，比如optBoolean, optString, optInt...

他们的意思是，如果这个jsonObject有这个属性，则返回这个属性，否则返回一个默认值。

****2.JSONArray：****

是一个有序的序列值。

它的表现形式是一个包裹在方括号的字符串，值和值之间使用逗号隔开。

内在形式是一个使用get()和opt()方法通过索引来访问值，和使用put()方法来添加或修改值的对象。

值可以是任何这些类型：Boolean,JSONArray,JSONObject,Number,和String，或者JSONObject.NULL对象。

代码演示如下：

public static void jsonArrayTest() {

JSONArray jsonarray = new JSONArray("[{'name':'xiazdong','age':20},{'name':'xzdong','age':15}]");

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

JSONObject jsonobj = jsonarray.getJSONObject(i);

String name = jsonobj.getString("name");

int age = jsonobj.getInt("age");

System.out.println("name = " + name + ",age = " + age);

}

}

嵌套的JSONObject和JSONArray代码演示如下：

public static void jsonObjectAndArrayTest() {

String jsonstring = "{'name':'xiazdong','age':20,'book':['book1','book2']}";

JSONObject jsonobj = new JSONObject(jsonstring);

String name = jsonobj.getString("name");

System.out.println("name" + ":" + name);

int age = jsonobj.getInt("age");

System.out.println("age" + ":" + age);

JSONArray jsonarray = jsonobj.getJSONArray("book");

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

String book = jsonarray.getString(i);

System.out.println("book" + i + ":" + book);

}

}

****3.JSONStringer：****

是一个用于快速构造JSON文本的工具

JSONWriter的子类

bject():开始一个对象，即添加{；enObject():结束一个对象，即添加}

 array():开始一个数组，即添加[； endArray():结束一个数组，即添加]

key():表示添加一个key；value():表示添加一个value

代码演示如下：

public static void jsonStringerTest() {

JSONStringer stringer = new JSONStringer();

stringer.object().key("name").value("xiazdong").key("age").value(20).endObject();

System.out.println(stringer);

}

负载的JSON格式写演示（PrintWriter+JSONStringer可以写入JSON文件）：

public static void jsonStringerTest2() throws FileNotFoundException {

JSONStringer jsonStringer = new JSONStringer();

JSONObject obj6 = new JSONObject();

obj6.put("title", "book1").put("price", "$11");

JSONObject obj3 = new JSONObject();

obj3.put("book", obj6);

obj3.put("author", new JSONObject().put("name", "author-1"));

JSONObject obj5 = new JSONObject();

obj5.put("title", "book2").put("price", "$22");

JSONObject obj4 = new JSONObject();

obj4.put("book", obj5);

obj4.put("author", new JSONObject().put("name", "author-2"));

JSONArray obj2 = new JSONArray();

obj2.put(obj3).put(obj4);

JSONObject obj1 = new JSONObject();

obj1.put("title", "BOOK");

obj1.put("signing", obj2);

jsonStringer.object().key("session").value(obj1).endObject();

System.out.println(jsonStringer.toString());

PrintWriter out = new PrintWriter(new FileOutputStream("1.txt"));

out.println(jsonStringer.toString());

out.close();

}

**4.JSONTokener**

它和JSONObject和JSONArray的构造函数一起使用，用于解析JSON源字符串

代码演示如下（JSONObject+JSONTokener能够获取JSON格式文本对象）：

public static void JSONTokenerTest() throws FileNotFoundException {

JSONObject jsonobj = new JSONObject(new JSONTokener(new FileReader(new File("1.txt"))));

System.out.println(jsonobj.getJSONObject("session").getJSONArray("signing").getJSONObject(1).getJSONObject("book").getString("title"));

}

注意：在Java中，JSON格式的字符串最好用单引号表示