

# 深圳大学实验报告

课程名称： 中间件技术

实验项目名称： 实验 3 应用 JSON 数据传输与交换编程

学院： 计算机与软件学院

专业： 软件工程

指导教师： 毛斐巧

报告人： 郑彦薇 学号： 2020151022 班级： 软件工程 01 班

实验时间： 2022 年 11 月 1、15 日（周二）

实验报告提交时间： 2022/11/1

教务部制

## 一、实验目的与要求：

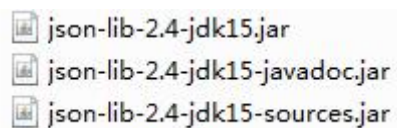
通过实例小项目的实践，直观的感受、体会和学用 JSON 数据交换与编程技术。具体包括：

- (1) 学习和掌握 JSON 语法。
- (2) 学习和掌握 JSON 数据交换编程技术。
- (3) 学习和掌握 Json-lib 工具包的使用，使用类库进行 JSON 解析

## 二、实验内容

内容：解决以下应用问题

0. 编程环境的准备：使用 Json-lib 工具进行 JSON 解析，首先要下载和在编程项目中导入该工具包和一些其它的支撑 JAR 包。Json-lib 工具下载地址有 <http://sourceforge.net/projects/json-lib/?source=directory>，下载后有如下三个 jar 文件：



其中，将第一个 Jar 包导入项目即可用其中的 API。具体做法是：在项目中建立 libs 文件夹，将第一个包复制进去，然后在 Java Build Path 里面 Add External JARs…。第二个包是文档，直接右键解压缩之后就会得到 html 文档，入口是 index.html，当然也可以利用工具（jd2chm.exe）将其转换为 chm 格式这样查看更方便。第三个包是源码，感兴趣或者以后碰到问题了可以看看。

导入了上面的包之后，运行程序会抛出运行时异常：Exception in thread "main" java.lang.NoClassDefFoundError。这是因为还要添加另外的包，如下：



这些包可以在网上自行搜索下载，也可以直接采用课程资源中提供的这些包（在 BB 或课程 QQ 群中有上传）。需要把上述包全部加入 libs 文件夹，并且把它们添加进 Build Path 中。

1. 编程实现对 JavaScript 对象属性的访问。要求创建 html 文件，使用初始化的方式创建 JavaScript 对象，并访问对象的属性。两个对象分别为：

对象 1： { 'name': 'Luffy', 'age': 17 }

对象 2： { 'name': 'Luffy', 'address': { 'city': 'Beijing', 'street': 'dzs', 'room': 17 } }

将上述两个对象分别改为描述自己的信息（即你本人的姓名、年龄，宿舍地址），运行

html 页面，点击对应链接可以分别查看两个对象的属性。可以参考讲义中 json01.html。

2. 编程实现对 JavaScript 数据中对象元素的访问。要求创建 html 文件，遵循数组的格式要求，使用方括号开头和结尾描述数组，数组中的元素有对象类型。例如：

数组 students: [{‘stuNo’:20170031, ‘name’:’何立立’}, {‘stuNo’:20170032, ‘name’:’赵多多’}]

对象 score: {‘stuNo’:20170031, ‘name’:’何立立’, ‘scores’:[{‘课程’:’中间件技术’, ‘成绩’:90}, {‘课程’:’数据结构’, ‘成绩’:86}, {‘课程’:’计算机网络’, ‘成绩’:80}]}

将上述 students 数组和 score 对象分别改为描述自己和另两位同学的信息（即 students 描述你和另两位同学的学号和姓名信息，score 对象描述你自己对应三门课的成绩信息，运行 html 页面，点击对应链接可以显示和查看你的学号和姓名，以及你的成绩信息。可以参考讲义中 json02.html。

3. 编程在符合 JSON 格式要求的字符串和 JSON 对象（以及 JSON 数组）之间互相转换。要求创建 html 文件，借助于原生的 eval 方法或者原生对象 JSON.parse(str)方法。

在 html 文件中定义一个脚本方法 jsStrToObj( )，在该方法中实现将字符串’{‘name’:’Luffy’, ‘age’:17}’转换为 JSON 对象，访问该对象的 name 属性值并以网页消息弹出显示，要求在 html 文件中点击对应“[JSON 字符串转换为对象](#)”的文字链接就可以调用 jsStrToObj( )方法，运行并弹出显示访问到的 name 属性值。

在 html 文件中定义一个脚本方法 jsStrToArr( )，在该方法中实现将字符串’[{‘stuNo’:20170031, ‘name’:’何立立’}, {‘stuNo’:20170032, ‘name’:’赵多多’}]’转换为 JSON 数组，访问该数组第二个元素的 name 属性值，并以网页消息弹出显示，要求在 html 文件中点击对应“[JSON 字符串转换为 JSON 数组](#)”文字链接就可以调用 jsStrToArr( )方法，运行并弹出显示访问到的 name 属性值。

在 html 文件中定义一个脚本方法 jsObjToStr( )，在该方法中实现将第 2 题中的 JSON 对象 score 转换为 JSON 字符串，要求在 html 文件中点击对应“[JSON 对象转换为 JSON 字符串](#)”文字链接就可以调用 jsObjToStr( )方法，运行并弹出显示转换后的字符串内容。

4. 编程在 java 对象（以及 java 数组）与 JSON 字符串之间互相转换。要求创建 Java 实例类 Student，该类的属性有 name 和 age；方法有：各属性的 get 和 set 方法，以及将 name 和 age 属性连为一个字符串的 toString( )方法。

创建含 main( )方法的 JSONtoJavaTest 类，在该类中 javaObjToJSONstr( )方法，在该方法中实现将一个的 student 对象（姓名和年龄分别为方依依,19）转换为 JSON 字符串，并显示输出该字符串。在该类中定义 javaArrToJSONstr( )方法，在该方法中实现将一个含三个 student 对象元素（三个 student 对象的姓名和年龄请设置为你同宿舍的三个同学的信息）的 java 数组转换为 JSON 字符串，并显示输出该字符串。在该类中定义 jsonStrTojavaObj( )方法，在该方法中实现将 JSON 字符串’{‘name’:’方依依’, ‘age’:17}’转换为 Student 类对象，并显示输出该类对象。在该类中定义 jsonStrToJavaArry( )方法，在该方法中实现将 JSON 字符串’[{‘name’:’何立立’, ‘age’:21}, {‘name’:’赵多多’, ‘age’:22}]’转换为 java 数组，并显示输出该数组中的元素。在 JSONtoJavaTest 类的 main 方法中分别调用上述四个方法，进行测试，观察运行结果。

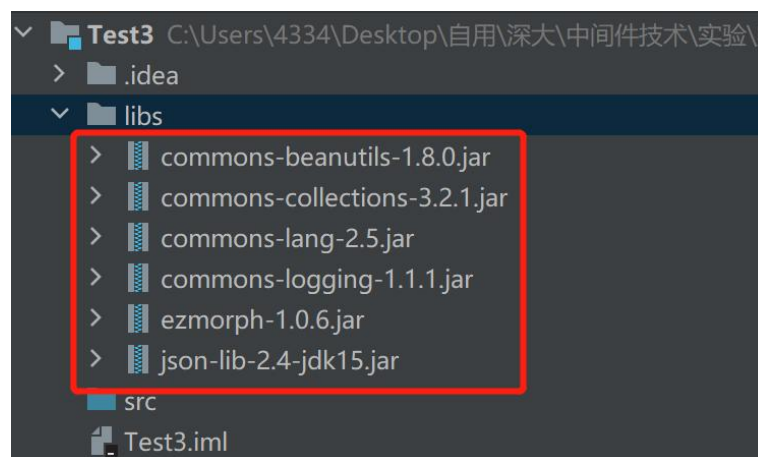
报告写作。要求：主要思路有明确的说明，重点代码有详细的注释，行文逻辑清晰可读性强，报告整体写作较为专业。 **(20 分)**

说明：

- (1) 本次实验课作业满分为 100 分，占总成绩的比例（待定）。
- (2) 报告正文：请在指定位置填写，本次实验**需要提交实验报告+源码+描述信息的 XML 文件**，其中，**源码和 XML 文件压缩成一个.rar(或.zip)文件提交**，实验报告**不要压缩进压缩包文件**。
- (3) 个人信息：**WORD 文件名中的“姓名”、“学号”，请改为你的姓名和学号**；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。
- (4) 提交方式：截至时间前，请在 Blackboard 平台中提交。
- (5) 发现雷同，所有雷同者该次作业记零分。
- (6) 期末考试阶段补交无效。

### 三、实验过程及结果

#### 0. 准备编程环境



#### 1. 源码、运行结果及设计思路简述。（20 分）

1.1. 在项目中新建 html 文件，使用初始化方法创建 JavaScript 对象，并访问对象的属性。编写代码及解释如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>json01.创建 JavaScript 对象并访问属性</title>
  <script type="text/javascript">
    function fl() {
      //使用 var 方法创建对象
      var obj1 = {};
      //添加属性
      obj1.name = "郑彦薇";
    }
  </script>
</head>
</html>
```

```

        obj1.age = "20";
        //html 页面显示对象信息
        alert("名字:"+obj1.name+"\n"+"年龄:"+obj1.age);
    }
    function f2(){
        var obj2 = {};
        obj2.name = "郑彦薇";
        //obj2 对象中的 address 属性也是一个对象
        obj2.address = {};
        obj2.address.city = "深圳";
        obj2.address.street = "聚翰";
        obj2.address.room = "1203";
        alert("名字:"+obj2.name+"\n"+"宿舍地址:"+obj2.address.city+obj2.address.street+obj2.address.room);
    }
</script>
</head>
<body>
    <!--html 页面中点击不同连接查看对象属性-->
    <a href = "javascript:;" onclick = "f1();">查看个人信息</a>
    <br>
    <a href = "javascript:;" onclick="f2();">查看宿舍地址</a>
</body>
</html>

```

1.2. 运行上述 html 文件，得到结果如下所示：



点击“查看个人信息”，访问第一个对象属性：



点击“查看宿舍地址”，访问第二个对象属性：



### 1.3. 设计思路的简述

Script 中：在 html 文件中首先创建两个 function 类，分别在两个 function 中创建对象，添加对象的属性，使用 “.” 访问对象属性并将对象属性按照一定格式进行输出。其中宿舍地址属性在 obj2 类中作为一个 address 类存在。

Body 中：加两个访问连接分别访问 script 中创建的两个类，点击相应连接按照类中定义的输出格式在 html 页面中查看对象属性。

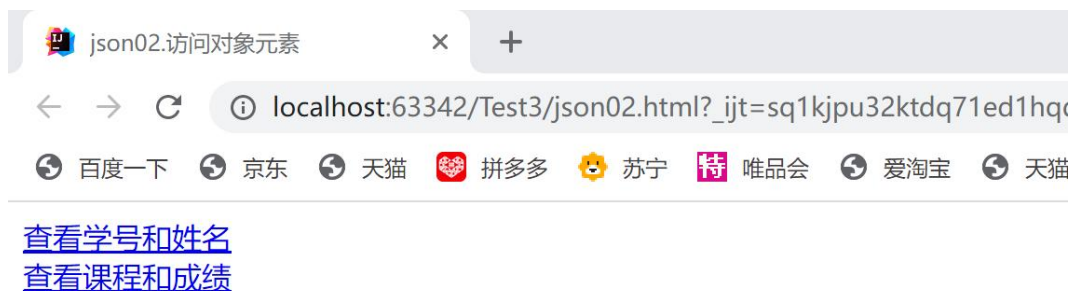
## 2. 源码、运行结果及设计思路简述（20 分）

2.1. 按照数组格式创建数组对象并添加属性值，然后按照一定格式输出。编写代码及解释如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>json02.访问对象元素</title>
  <script type="text/javascript">
    function f1(){
      //数组格式创建对象并设置属性
      var students=[{'stuNo': 2020151022, 'name': '郑彦薇'},
        {'stuNo': 2021280224, 'name': '小蓝'},
        {'stuNo': 2022152022, 'name': '小郑'}
      ];
      //按照一定格式输出对象属性
      alert("学号:"+students[0].stuNo+"\n"+"名字:"+students[0].name);
    }
    function f2(){
      var score={'stuNo':2020151022, 'name': '郑彦薇',
        'scores':[{'课程': '中间件技术', '成绩': 95},
          {'课程': '数据结构', '成绩': 90},
          {'课程': '计算机网络', '成绩': 90}]
        }
      alert("名字:"+score.name+"\n"+"课程:"+score.scores[0].课程+" 取得成绩:"+score.scores[0].成绩+"\n"
        +"课程:"+score.scores[1].课程+" 取得成绩:"+score.scores[1].成绩+"\n"
        +"课程:"+score.scores[2].课程+" 取得成绩:"+score.scores[2].成绩)
    }
  </script>
</head>
<body>
  <!--创建 JavaScript 对象并查看数据元素-->
  <a href="javascript:" onclick="f1();">查看学号和姓名</a>
  <br>
```

```
<a href="javascript:" onclick="f2();">查看课程和成绩</a>
</body>
</html>
```

2.2. 运行上述 html 文件，得到结果如下图所示：



点击“查看学号和姓名”，访问第一个对象属性：



点击“查看课程和成绩”，访问第二个对象属性：



### 2.3. 设计思路简述

首先是定义一个数组，数组元素为对象，定义的同时添加对象属性。然后通过数组下标访问第一个对象中的属性值，即我的个人信息。

同理定义 score 类，score 类中有属性 scores，属性 scores 为一个数组，数组元素为包含“课程”和“成绩”两个属性的对象。然后同样通过对象中访问属性的方式以及数组中通过下标访问元素的方式对数据进行输出。

## 3.源码、运行结果及设计思路简述（20 分）

3.1. 根据题目要求创建三个方法，分别使用 JSON 内置函数进行相应的转换。编写代码及解释如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/javascript;charset=utf-8">
  <title>json03.JSON 格式字符串和 JSON 对象之间的转换</title>
  <script type="text/javascript">
```

```

//将字符串转换为 JSON 对象
function jsStrToObj(){
    var str='{"name":"Luffy","age":17}';
    //使用 JSON.parse()方法进行转换
    var json = JSON.parse(str);
    console.log(json);
    alert(json.name);//访问 json 对象中的 name 属性
}
//将字符串转换为 JSON 数组
function jsStrToArr(){
    var str=[{"stuNo":20170031,"name":"何立立"}, {"stuNo":20170032,"name":"赵多多"}];
    //使用 JSON.parse()方法进行转换
    var arr = JSON.parse(str);
    alert(arr[1].name);//访问数组第二个元素的 name 属性值
}
//JSON 对象转换为字符串
function jsObjToStr(){
    var score={'stuNo':2020151022, 'name':'郑彦薇',
        'scores':[{'课程':'中间件技术','成绩':95},
            {'课程':'数据结构','成绩':90},
            {'课程':'计算机网络','成绩':90}]
    }
    //使用 JSON.stringify()方法进行转换
    var str=JSON.stringify(score);
    alert(str);//输出转换后的字符串
}
</script>
</head>
<body>
    <a href="javascript:" onclick="jsStrToObj();">JSON 字符串转换为对象</a>
    <br>
    <a href="javascript:" onclick="jsStrToArr();">JSON 字符串转换为 JSON 数组</a>
    <br>
    <a href="javascript:" onclick="jsObjToStr();">JSON 对象转换为 JSON 字符串</a>
</body>
</html>

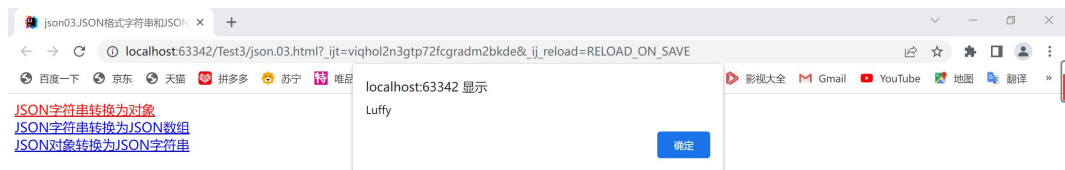
```

3.2. 运行上述 html 文件得到：





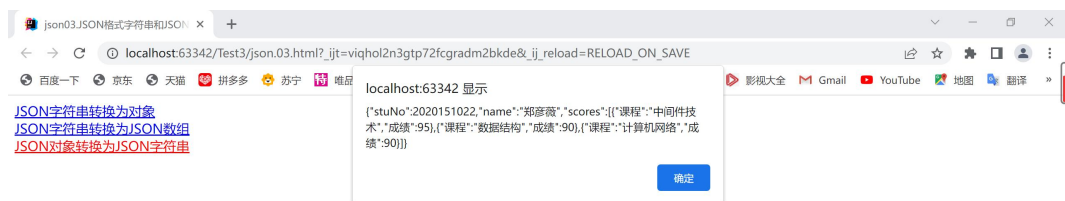
点击“JSON 字符串转换为对象”，输出对象中的 name 属性：



点击“JSON 字符串转换为 JSON 数组”，输出数组第二个元素中的 name 属性：



点击“JSON 对象转换为 JSON 字符串”，输出转换后的字符串：



### 3.3. 设计思路简述：

首先在 script 中定义待转换的字符串、对象，然后使用原生对象 JSON.parse() 方法和 JSON.stringify() 方法即可实现将字符串转换为对象或数组，将对象转换为字符串，最后再以正确格式进行访问输出。

## 4. 源码、运行结果及设计思路简述（20 分）

4.1. 根据类的属性要求和方法要求，编写 students.java 和 JSONtoJavaTest.java 代码及解释如下：

students.java:

```
import java.util.ArrayList;
import java.util.List;
import net.sf.json.JSONArray;
import net.sf.json.JSONObject;
```

```

public class students{
    //属性 name 和 age
    private String name;
    private int age;
    //各个属性的 get 和 set 方法
    public String getName(){
        return name;
    }
    public void setName(String name){
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    //将 name 和 age 属性连为一个字符串的 toString()方法
    public String toString(){
        return this.name+this.age;
    }
}

```

JSONtoJavaTest.java:

```

import net.sf.json.JSONArray;
import net.sf.json.JSONObject;
import net.sf.json.processors.JsonVerifier;

import java.util.ArrayList;
import java.util.List;

public class JSONtoJavaTest {
    //定义 javaObjToJSONstr()方法， 将一个 student 对象转换为 JSON 字符串
    public static void javaObjToJSONstr() {
        students s = new students();
        s.setName("方依依");
        s.setAge(19);
        JSONObject jsonobj = JSONObject.fromObject(s); //后台封装数据
        String str = jsonobj.toString(); //调用对象的 toString 方法
        System.out.println("student 对象转换为 JSON 字符串:");
        System.out.println(str);
    }

    //定义 javaArrToJSONstr()方法， 将一个含 3 个 student 对象的 java 数组转换为 JSON 字符串
}

```

```

public static void javaArrToJSONstr(){
    List<students> s = new ArrayList<students>();//规定存放的类型
    students s1=new students();
    s1.setName("郑彦薇");
    s1.setAge(20);
    s.add(s1);//将对象 s1 放入 s 数组
    students s2=new students();
    s2.setName("魏晓欢");
    s2.setAge(19);
    s.add(s2);
    students s3=new students();
    s3.setName("王晓丽");
    s3.setAge(20);
    s.add(s3);
    //后台封装数组数据
    JSONArray jsonarr=JSONArray.fromObject(s);
    //调用 toString 方法将 java 数组转换为字符串
    String str=jsonarr.toString();
    System.out.println("Java 数组转换为 JSON 字符串:");
    System.out.println(str);
}

```

//定义 jsonStrTojavaObj()方法，将 JSON 字符串转换为 student 类对象

```

public static void jsonStrTojavaObj(){
    String str="{ 'name': '方依依', 'age': 17 }";
    JSONObject jsonObj=JSONObject.fromObject(str);
    students s=(students) JSONObject.toBean(jsonObj,students.class);//将 JSONObject 对象转
换为 java 对象
    System.out.println("JSON 字符串转换为 student 对象:");
    System.out.println(s);
}

```

//定义 jsonStrToJavaArray()方法，将 JSON 字符串转换为 java 数组

```

public static void jsonStrToJavaArray(){
    String str="[ { 'name': '何立立', 'age': 21 }, { 'name': '赵多多', 'age': 22 } ]";
    JSONArray jsonArr=JSONArray.fromObject(str);
    //定义转换类型为 List<students>
    List<students> s=(List<students>) JSONArray.toCollection(jsonArr,students.class);
    System.out.println("JSON 字符串转换为 java 数组:");
    System.out.println(s);
}

```

//main()方法

```

public static void main(String[] args) {

```

```

        jsonObjToJSONstr();
        javaArrToJSONstr();
        jsonStrTojavaObj();
        jsonStrToJavaArray();
    }
}

```

4.2. 运行 JSONtoJavaTest.java 文件，得到运行结果如下图所示：

```

student对象转换为JSON字符串：
{"age":19,"name":"方依依"}
Java数组转换为JSON字符串：
[{"age":20,"name":"郑彦薇"}, {"age":19,"name":"魏晓欢"}, {"age":20,"name":"王晓丽"}]
JSON字符串转换为student对象：
方依依17
JSON字符串转换为java数组：
[何立立21, 赵多多22]

```

4.3. 设计思路简述

首先根据题目要求，编写 students.java，包括两个属性 name 和 age，同时分别设置 get 方法和 set 方法。其次是对 JSONtoJavaTest.java 的编写，根据四个方法的具体要求，利用 JSON 内置函数，对数据进行定义、封装，并仿照第 3 题的转换方法进行转换，然后进行输出。

## 四、实验总结与体会

本次实验主要是进行 JSON 数据传输和交换编程。

JSON 对象和 JSON 字符串之间的转换是数据传输的关键，在 js 中，使用 JSON.parse() 方法和 JSON.stringify() 方法实现 json 与对象的转换；在 java 中，使用 fromJson() 和 toJson() 方法实现 json 与对象的转换。

通过本次实验，学习了如何进行 JSON 与数组、对象之间的相互转换与输出。与 xml 相比，JSON 具有长度更短，能够直接被 JavaScript 解释器解析，能够使用数组等优势。

## 五、成绩评定及评语

1.指导老师批阅意见:

2.成绩评定:

指导教师签字:毛斐巧  
2022 年    月    日