

1 JSON介绍

2 JSON在JavaScript中的使用

2.1 JSON的定义

2.2 JSON的转换

3 JSON在Java中的使用

3.1 JavaBean和JSON的相互转换

3.2 List和JSON的相互转换

3.3 Map和JSON的相互转换

1 JSON介绍

JSON (JavaScript Object Notation, JavaScript 对象表示法) 是一种轻量级的数据交换格式, 易于人阅读和编写, 也易于机器解析和生成。“轻量级”指的是与 XML 相比, JSON 解析速度更快; “数据交换”指的是客户端和服务端之间的业务数据传输。

JSON 采用完全独立于语言的文本格式, 而且很多语言都提供了对 JSON 的支持, 这就使得 JSON 成为理想的数据交换语言。

2 JSON在JavaScript中的使用

2.1 JSON的定义

JSON 由键值对组成, 并且被大括号包围。每个键用双引号引起来, 键和值之间用冒号分隔, 多组键值对之间用逗号分隔。其中键必须加双引号。

JSON 中允许的值: 数值、字符串、布尔值、`null`、对象、数组。

JSON 中的值不可以是以下数据类型之一: 函数、日期、`undefined`。

JSON 对象就是与 JavaScript 对象, 其定义和访问方式与 JavaScript 对象相同。

2.2 JSON的转换

JSON 有两种存在方式: 一种是以对象的形式存在, 称为 JSON 对象; 另一种是以字符串的形式存在, 称为 JSON 字符串。

一般在操作 JSON 中的数据时使用 JSON 对象, 在客户端和服务端之间进行数据交换时使用 JSON 字符串。

在 JavaScript 中有一个 `JSON` 对象, 这个对象可以用于 JSON 对象与 JSON 字符串之间的转换。

JSON 对象转换为 JSON 字符串: `JSON.stringify(jsonObj)`。

JSON 字符串转换为 JSON 对象: `JSON.parse(jsonString)`。

IE 7 及以下版本不支持 `JSON` 对象。为了解决兼容性问题, 可以引入外部 JavaScript 文件。

3 JSON在Java中的使用

在 Java 中使用 JSON 需要导入相关 jar 包, 这里使用 Gson 包。

3.1 JavaBean和JSON的相互转换

```
1 Person obj = ... // 创建对象
2 Gson gson = new Gson(); // 创建 Gson 对象
3 String jsonString = gson.toJson(obj); // 将对象转换为 JSON 字符串
4 Person obj1 = gson.fromJson(jsonString, Person.class); // 将 JSON 字符串转换为对象
```

3.2 List和JSON的相互转换

```
1 List<Person> objList = new ArrayList<>();
2 objList.add(new Person());
3
4 Gson gson = new Gson(); // 创建 Gson 对象
5 String listString = gson.toJson(objList); // 将 List 转换为 JSON 字符串
6
7 // 要想将 JSON 字符串转换为 List, 需要先创建一个类, 其中的泛型类型为要转换的类型
8 public class PersonListType extends TypeToken<List<Person>> {}
9 // 将 JSON 字符串转换为 List
10 List<Person> list = gson.fromJson(listString, new
    PersonListType().getType());
```

3.3 Map和JSON的相互转换

```
1 Map<Integer, Person> personMap = HashMap<>();
2 personMap.put(...);
3
4 Gson gson = new Gson(); // 创建 Gson 对象
5 String mapString = gson.toJson(personMap); // 将 Map 转换为 JSON 字符串
6
7 // 创建一个类
8 public class PersonMapType extends TypeToken<Map<Integer, Person>> {}
9 // 将 JSON 字符串转换为 Map
10 Map<Integer, Person> map = gson.fromJson(mapString, new
    PersonMapType().getType());
11
12 // 用匿名内部类进行转换
13 Map<Integer, Person> map2 = gson.fromJson(mapString,
14     new TypeToken<Map<Integer, Person>>()
        {}.getType());
```