Json学习笔记

1. **Json数据格式：**
2. []：表示数组；
3. {}：表示对象；
4. “”：表示字段；
5. ： ：表示后面的数据是前面字段的值；

**eg：**

1）json文件中的数据：可以由对象、数组组成。本例子的json数据是一个数组，里面嵌套String、List类型的数据。

[  
 {  
 "name": "chris",  
 "age": "18",  
 "child": []  
 },  
 {  
 "name": "vicky",  
 "age": "17",  
 "child": [  
 {  
 "name": "emm",  
 "age": "1",  
 "like": ["读书", "游泳"]  
 }  
 ]  
 }  
]

2）将上面的json数据转换成java对象，对应java对象如下：

**//从json文件中加载数据，调用该方法，就可以获取json文件中的数据。**  
private List<**ParentDTO**> loadRecordsFromInitJsonFile() {  
 String jsonFilePath = "json文件路径";  
 try(InputStream inputStream = this.getClass().getResourceAsStream(jsonFilePath)) {  
 byte[] jsonContent = IOUtils.toByteArray(inputStream);  
 return JSONObject.parseArray(new String(jsonContent), **ParentDTO**.class);  
 } catch (IOException e) {  
 return Collections.*emptyList*();  
 }  
}

ParentDTO.java对象：**（此处必须有属性的set、get方法）**

public class **ParentDTO** {  
 private String name;  
 private String age;  
 private List<ChildDTO> childList;

}

ChildDTO.java**（此处必须有属性的set、get方法）**

public class **ChildDTO** {  
 private String name;  
 private String age;  
 private List<String> likeList;

}

1. 一般情况下，仅仅是解析前端回传的json格式的数据，是不需要上面那样——将json文件中的数据读取出来，映射到对应的java对象上。

**只需要按照json的行的属性名来解析数据即可。**

eg：使用fastJson解析

1. 生成json数据对象：

public class CreateJson {

**//生成json数据对象**  
 public JSONObject create() {  
 JSONObject object = new JSONObject();  
 object.put("name","chris");  
 object.put("age",2);  
 object.put("isMan",true);  
 List<Integer> integers = Arrays.*asList*(1,2,3);  
 object.put("others",integers);  
  **//以字符串的格式打印json数据**  
 System.*out*.println(object);  
 return object;  
 }

}

1. 解析生成给的json数据对象

public static void main(String[] args) {  
 CreateJson createJson = new CreateJson();  
  **//生成json数据：{"name":"chris","isMan":true,"age":2,"others":[1,2,3]}**  
 JSONObject jsonObject = createJson.create();  
 //获取name的数据  
 String s = jsonObject.**getString**("name");  
 //获取age的数据  
 int age = jsonObject.**getIntValue**("age");  
 //获取isMan的数据  
 boolean isMan = jsonObject.**getBooleanValue**("isMan");  
 //获取others的数据  
 List<Integer> integers = JSON.**parseArray**(jsonObject.**getJSONArray**("others").toJSONString(),Integer.class);  
}

1. **Json解析的四种方法：**

**注意：**使用这些方法的时候，若涉及到java对象，则该对象必须具有set、get、构造器方法

1. 传统方式

* 使用实体类生成json数据

public Person getPerson(){  
 return new Person("张三", "男", 25);  
}  
public void EntityToJson(){  
 JSONObject jsonObject = new JSONObject();  
 jsonObject.put("name", getPerson().getName());  
 jsonObject.put("sex", getPerson().getSex());  
 jsonObject.put("age", getPerson().getAge());  
 System.*out*.println(jsonObject.toString());  
}

* 使用string数据生成json数据

public void JsonToEntity(){  
 String jsonString = "{\"sex\":\"男\",\"name\":\"张三\",\"age\":25}";  
 JSONObject jsonObject = new JSONObject(jsonString);  
 Person person = new Person(jsonObject.get("name").toString(),

jsonObject.get("sex").toString(),

Integer.*valueOf*(jsonObject.get("age").toString()));  
 System.*out*.println(person.toString());  
}

1. 使用Jackson

public void EntityToJson() throws IOException {  
 Person person = new Person("张三", "男", 25);  
 **ObjectMapper** objectMapper = new ObjectMapper();  
 String jsonString = objectMapper.writeValueAsString(person);  
 System.*out*.println(jsonString);  
}

1. 使用GJson

public void EntityToJson(){  
 Person person = new Person("张三", "男", 25);  
 **Gson** gson = new Gson();  
 String jsonString = gson.toJson(person);  
 System.*out*.println(jsonString);  
}

1. **使用FastJson：**

* jar包（com.alibaba.fastjson）
* 常用的JSONObject、JSOn都是这里面的类

public void EntityToJson(){  
 Person person = new Person("张三", "男", 25);  
 Object jsonString = JSON.toJSON(person);  
 System.*out*.println(jsonString.toString());  
}