使用目的

目前数据交互大多以 JSON 字符串为信息传输，主要格式有

{"age":"22","name":"李四"}

[{"age":"21","name":"张三"}]

最常见的应用场景是前后端对接，第三方平台文档对接，下面展示转换使用。

# 一、引入 jar

        <dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>1.1.31</version>

</dependency>

# 二、建立测试类对象

class Student {

private String name;

private String age;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAge() {

return age;

}

public void setAge(String age) {

this.age = age;

}

}

# 三、转换

## 1、对象转JSON

        Student stu1 = new Student();

stu1.setName("张三");

stu1.setAge("21");

String stu1Json = JSONObject.toJSONString(stu1);

输出：

{"age":"21","name":"张三"}

## 2、JSON转对象

        Student stu1to = JSON.parseObject(stu1Json, Student.class);

System.out.println("json 转对象：");

System.out.println(stu1to);

System.out.println(stu1to.getName());

System.out.println(stu1to.getAge());

输出：

json 转对象：

Student@2aae9190

张三

21

注：JSON键名要与对象属性名相对应

## 3、对象数组转JSON

        Student stu2 = new Student();

stu2.setName("李四");

stu2.setAge("22");

List<Student> list = new ArrayList<Student>();

list.add(stu1);

list.add(stu2);

String listJson = JSONObject.toJSONString(list);

System.out.println(listJson);

输出：

[{"age":"21","name":"张三"},{"age":"22","name":"李四"}]

## 4、JSON 转 对象数组

        List<Student> studentList = JSON.parseArray(listJson, Student.class);

for (Student student : studentList) {

System.out.println(student.getName());

}

输出：

json 数组格式转换对象

张三

李四

## 5、JSON多级组合，适用于请求文档传输参数

        JSONObject jsona = new JSONObject();

jsona.put("number", "1");

JSONObject jsonb = new JSONObject();

jsonb.put("listMap", list);

JSONObject jsonAll = new JSONObject();

jsonAll.put("jsona", jsona);

jsonAll.put("jsonb", jsonb);

String jsonAllStr =JSONObject.toJSONString(jsonAll);

System.out.println(jsonAllStr);

输出：

{"jsona":{"number":"1"},"jsonb":{"listMap":[{"age":"21","name":"张三"},{"age":"22","name":"李四"}]}}

## 6、多级 JSON 组合

### 6.1、只获取指定字段，接上方示例代码

String getJsona = JSON.parseObject(jsonAllStr).getString("jsona");

String strjsona = JSON.parseObject(getJsona, String.class); //指定获取 字段名对象信息，如果为单个String可不指定，这里作为实例写出

System.out.println("只拿jsona信息");

System.out.println(strjsona);

输出：

只拿jsona信息

{"number":"1"}

### 6.2、获取指定字段对象

String getJsonb = JSON.parseObject(jsonAllStr).getString("jsonb");

String getJsonbb = JSON.parseObject(getJsonb).getString("listMap");    //这里被二级包裹，所以要获取2次才能转换对象数组

List<Student> strjsonb = JSON.parseArray(getJsonbb, Student.class);

System.out.println("只拿jsonbb信息");

System.out.println(strjsonb);

输出：

只拿jsonbb信息

[Student@3d04a311, Student@7a46a697]