搭建json与对象之间相互转换的桥梁。pom.xml中加入

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.8.5</version>

</dependency>

# Obj → Json

Person person = new Person("a",1,"b")

Gson gson = new Gson();

gson.toJson(person);//对象转为json，返回字符串，{"name":"a","age":1,"email":"b"}

# List → Json

List<Person> persons = new ArrayList<>();

persons.add(new Person("a",1,"b"));

persons.add(new Person("b",2,"b"));

gson.toJson(persons);//list转json，返回字符串，[{"name":"a","age":1,"email":"b"},{"name":"b","age":2,"email":"b"}]

# Json → obj

String str = "{"name":"a","age":1,"email":"b"}";

Person person = gson.fromJson(str, Person.class);

# Json → List

String str = " [{"name":"a","age":1,"email":"b"},{"name":"b","age":2,"email":"b"}]"

List<Person> reList = gson.fromJson(str,new TypeToken <List<Person>>(){}.getType());

# JsonParser

解析json字符串，返回jsonElement

JsonParser parser = new JsonParser();

JsonElement element = parser.parse(result);

# JsonElement

Json元素，可获取属性

# JsonObject

Json对象，从jsonElement中获取

# @SerializedName(value="...",alternate="...")

该注解的alternate属性能指定字段在JSON中对应的字段名称，value可忽略不写。

# @Expose(deserialize = true,serialize = true)

该注解用于限定是否序列化（serialize）/反序列化（deserialize）pojo类中某属性。