

# 复杂参数类型配置说明

1 配置公约

2 POJO配置协议说明

  2.1 协议格式

  2.2 协议字段：

3 不同参数类型配置示例

  3.1 方法参数全部为java基本类型包装类

  3.2 方法参数为简单类型List

  3.3 方法参数为简单POJO对象数组

  3.4 方法参数为简单POJO单对象

  3.5 方法参数为简单POJO List集合对象

  3.6 方法参数为复杂POJO 单对象

  3.5 方法参数为基本类型+复杂POJO单对象

  3.7 方法参数为复杂POJO Set集合对象

  3.8 方法参数为多个POJO对象（不支持）

4 POJO参数类型配置生成工具

!!!! 文档最后有复杂参数类型配置自动生成工具使用说明!!!!

## 1 配置公约

序号	公约	说明
1	支持参数类型	java基本类型包装类及类型数组； java集合类型（List、Set、Map）； 自定义JavaBean对象POJO、POJO数组、POJO集合类；
2	不支持参数类型	java基本类型（接口方法参数不使用基本类型， <b>使用包装类</b> ）
3	POJO类型使用	参数类型使用POJO类型，需要提前在类型管理中定义，然后才能选择
4	参数来源	支持URL获取和Http Body体获取，但每个方法中只能有一个参数选择从Body获取
5	Java基础类型集合参	例如：List<String> List<Integer>， <b>如果String类型 GET和POST</b>

数	请求都支持，如果是数字类型只能POST请求。原因：GET请求默认都是字符串。
---	--

## 2 POJO配置协议说明

### 2.1 协议格式

JSON格式

### 2.2 协议字段：

字段名	字段说明	备注
name	参数英文名	配置方法请求参数时，用于选择类型时筛选
desc	参数类型中文名	配置方法请求参数时，用于选择类型时筛选
type	类型class	填写POJO完整的类路径
collectionType	集合类型	当参数类型是集合POJO时，通过该字段指定是什么集合类 ( <b>不是集合POJO不用填</b> )
children	类型子属性POJO类型列表	适用POJO内部属于引用了另一个POJO复杂对象类型的场景
-name	子属性英文名（别填错了）	POJO内部属性的类型会通过此名称去和请求参数匹配参数值
-desc	子属性中文名	POJO内部属性的类型会通过此名称去和请求参数匹配参数值
-type	子属性类型class	POJO内部属性的类型完整class路径
-collectionType	子属性集合类型	POJO内部属性的类型会通过此名称去和请求参数匹配参数值 ( <b>Map类型需要填写，其它不需要</b> )
-mapping	POJO内部属性为Map类型的复杂参数类型映射	当POJO内部属性是Map集合类型，而且不同的Key对应不同的POJO对象时，可以通过该字段进行复杂场景配置

## 3 不同参数类型配置示例

### 3.1 方法参数全部为java基本类型包装类

```
1 Teacher saveTeacher(String name, String sku, Integer age);
```

**API基本信息**

*API名称: zhiyong.gateway.api.demo.saveTeacher (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名: com.zhiyong.gateway.facade.api.GatewayTestFacade (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述: 保存老师对象 (基本类型参数)	*RPC方法名: saveTeacher
*API分类: 配置Demo	*RPC版本: 1.0.0
*Api版本: 1.0	*登录检查: 不需要
*超时时间: 3000 (单位: 毫秒)	*权限检查: 不需要
*请求方式: GET	*限流次数: 99999 (TPS或QPS)

**API请求参数**

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例
1	name	老师姓名	java.lang.String	请求URL	必填	
2	sku	科目	java.lang.String	请求URL	必填	
3	age	年龄	java.lang.Integer	请求URL	必填	

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String, UserModel>>|UserModel[]等POJO及嵌套类型), 需要先到"参数类型"管理菜单配置类型, 然后再选择类型! !

GET http://localhost:19002/router/rest?appKey=ABC123&session=1234&timestamp=2020-11-25 22:35:34&versic ...

appKey	ABC123	
session	1234	
timestamp	2020-11-25 22:35:34	
version	1.0	
sign	905C8EDC71831DA89F8DFD934C9CE215	
method	zhiyong.gateway.api.demo.saveTeacher	
tenantId	123	
name	张三	
sku	一建	
age	36	
Key	Value	Description

Body Cookies (1) Headers (5) Test Results Status: 200 OK Time: 6.48 s Size: 344 B Save Response ▾

Pretty Raw Preview Visualize JSON  

```

1 {
2   "requestId": "d51ccd78ef4b4336a36014f9794a1207",
3   "code": 200,
4   "msg": null,
5   "data": {
6     "name": "张三",
7     "id": null,
8     "sku": "一建",
9     ...
10    ...
11  }
12 }
```

## 3.2 方法参数为简单类型List

```
1 List<String> queryTeacher(List<String> teacherIds);
```

API基本信息

*API名称:	zhiyong.gateway.api.demo.queryTeacher (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名:	com.zhiyong.gateway.facade.api.GatewayTestFacade (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述:	测试demo-查询老师 (基本参数List)	*RPC方法名:	queryTeacher
*API分类:	配置Demo	*RPC版本:	1.0.0
*Api版本:	1.0	*登录检查:	不需要
*超时时间:	3000 (单位: 毫秒)	*权限检查:	不需要
*请求方式:	GET POST形式也可以	*限流次数:	99999 (TPS或QPS)

API请求参数

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例
1	teacherIds	老师ID	java.util.List	请求URL	必填	

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String,UserModel>>|UserModel[]等POJO及嵌套类型), 需要先到"参数类型"管理菜单配置类型, 然后再选择类型!

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> appKey	ABC123	
<input checked="" type="checkbox"/> session	1234	
<input checked="" type="checkbox"/> timestamp	2020-11-25 22:35:34	
<input checked="" type="checkbox"/> version	1.0	
<input checked="" type="checkbox"/> sign	905C8EDC71831DA89F8DFD934C9CE215	
<input checked="" type="checkbox"/> method	zhiyong.gateway.api.demo.queryTeacher	
<input checked="" type="checkbox"/> tenantId	123	
<input checked="" type="checkbox"/> teacherIds	1	
<input checked="" type="checkbox"/> teacherIds	2	
Key	Value	Description

Body Cookies (1) Headers (5) Test Results Status: 200 OK Time: 724 ms Size: 251 B Save Response ▾

Pretty Raw Preview Visualize JSON

```

1 {
2   "requestId": "108a172f6a794011a14a781e19f2cc1c",
3   "code": 200,
4   "msg": null,
5   "data": [
6     "1",
7     "2"
8   ]
9 }
```

### 3.3 方法参数为简单POJO对象数组

```

1 public class Teacher implements Serializable {
2     private static final long serialVersionUID = 80175249154340209
3     89L;
4     private Long id;
5     private String name;
6     private String sku;
7     private int age;
8 }
9 Teacher[] saveTeacher(Teacher[] teachers);

```

第一步：由于Teacher是自定义POJO对象，所以需要先到类型管理里面添加Teacher数组对象的定义：

致用网关平台 欢迎 admin 修改密码 安全退出

网关文档 参数类型 ×

高级查询

添加 修改 发布 下线

ID	类型名	类型描述	状态	类型结构定义	最后修改人	创建时间	创建人	最后修改时间	最后修改人
1	20 POJO-TestModel-2222	测试POJO类型	草稿	预览		2020/12/04 22:09:27	admin		
2	19 POJO-TestModel-1111	测试POJO类型	草稿	预览		2020/12/04 22:05:43	admin		
3	21 POJO-TestModel-Teacher[]	测试POJO类型-老师对象数组	已发布	预览	admin	2020/12/05 14:33:39	admin	2020/12/05 14:33:54	admin
4	18 POJO-AddInviteCodeRequest	创建租户邀请码	已发布	预览	admin	2020/12/04 18:51:47	test	2020/12/04 20:50:58	admin
5	17 POJO-TestModel	测试POJO	已发布	预览	admin	2020/12/03 23:51:46	admin	2020/12/04 20:50:48	admin
6	16 POJO-live-list	直播列表对象	已下线	预览	admin	2020/12/03 17:12:53	admin	2020/12/04 20:54:50	admin

```
▼ {  
    collectionType: "com.zhiyong.gateway.facade.model.Teacher[]",  
    name: "teachers",  
    type: "com.zhiyong.gateway.facade.model.Teacher",  
    desc: "Teacher对象数组"  
}
```

第二步：在API里面选择该类型，进行参数配置

**API基本信息**

*API名称: zhiyong.gateway.api.demo.saveTeacher (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名: com.zhiyong.gateway.facade.api.GatewayTestFacad (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述: 保存老师对象 (参数POJO数组)	*RPC方法名: saveTeacher
*API分类: 配置Demo	*RPC版本: 1.0.0
*API版本: 2.0	*登录检查: 不需要
*超时时间: 3000 (单位: 毫秒)	*权限检查: 不需要
*请求方式: POST	*限流次数: 99999 (TPS或QPS)

保存

**API请求参数**

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例	操作
1	teachers	老师对象列表	测试POJO类型-老师对象数组[POJO-Test]	请求Body	必填		

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String,UserModel>|UserModel[]等POJO及嵌套类型), 需要先到"参数类型"管理菜单配置类型, 然后再选择类型! !

保存

The screenshot shows a Postman request configuration. The method is set to POST, and the URL is `http://localhost:19002/router/rest?appKey=ABC123&session=1234&timestamp=2020-11-25 22:35:34&version=1.0.0`. The 'Body' tab is selected, and the content type is set to JSON. The request body contains the following JSON:

```
[{"name": "张三", "sku": "一建", "age": 36}, {"name": "李四", "sku": "二建", "age": 39}]
```

The response status is 200 OK, and the response body is displayed in pretty JSON format:

```
{
  "requestId": "b8369a8db0c04271948462d2388fb334",
  "code": 200,
  "msg": null,
  "data": [
    {
      "name": "张三",
      "id": null,
      "sku": "一建",
      "class": "com.zhiyong.gateway.facade.model.Teacher",
      "age": 36
    }
  ]
}
```

### 3.4 方法参数为简单POJO单对象

```
1 public class Course implements Serializable {
2     private static final long serialVersionUID = -504495940219884
5281L;
```

```

3
4     private Long id;
5     private String name;
6     private String desc;
7     private int type;
8     private int sku;
9 }
10
11 Course saveCourse(Course courses);

```

第一步：由于Course是自定义POJO对象，所以需要先到类型管理里面添加Course数组对象的定义：

ID	类型名	类型描述	状态	类型结构定义	最后修改人	创建时间	创建人	最后修改时间	最后修改人
1	20 POJO-TestModel-2222	测试POJO类型	草稿	预览		2020/12/04 22:09:27	admin		
2	19 POJO-TestModel-1111	测试POJO类型	草稿	预览		2020/12/04 22:05:43	admin		
3	22 POJO-TestModel-Course	测试POJO类型-课程对象	已发布	预览	admin	2020/12/05 15:03:41	admin	2020/12/05 15:04:41	admin
4	21 POJO-TestModel-Teacher[]	测试POJO类型-老师对象数组	已发布	预览	admin	2020/12/05 14:33:39	admin	2020/12/05 14:33:54	admin
5	18 POJO-AddInviteCodeRequest	创建租户邀请码	已发布	预览	admin	2020/12/04 18:51:47	test	2020/12/04 20:50:58	admin
6	17 POJO-TestModel	测试POJO	已发布	预览	admin	2020/12/03 23:51:46	admin	2020/12/04 20:50:48	admin
7	16 POJO-live-list	直播列表对象	已下线	预览	admin	2020/12/03 17:12:53	admin	2020/12/04 20:54:50	admin

```

▼{
  name: "course",
  type: "com.zhiyong.gateway.facade.model.Course",
  desc: "课程对象"
}

```

第二步：在API里面选择该类型，进行参数配置

**API基本信息**

*API名称: zhiyong.gateway.api.demo.saveCourse (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名: com.zhiyong.gateway.facade.api.GatewayTestFacad (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述: 测试demo (简单POJO单对象)	*RPC方法名: saveCourse
*API分类: 配置Demo	*RPC版本: 1.0.0
*Api版本: 1.0	*登录检查: 不需要
*超时时间: 3000 (单位: 毫秒)	*权限检查: 不需要
*请求方式: POST	*限流次数: 99999 (TPS或QPS)

保存

**API请求参数**

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例	操作
1	course	课程对象	测试POJO类型-课程对象[POJO-TestMod]	请求Body	必填		

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String,UserModel>>|UserModel[]等POJO及嵌套类型), 需要先到"参数类型"管理菜单配置类型, 然后再选择类型!!

保存

POST http://localhost:19002/router/rest?appKey=ABC123&session=1234&timestamp=2020-11-25 22:35:34&version:1

Params ● Authorization Headers (10) Body (1) Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {"name": "课程A", "desc": "", "sku": 123, "type": 1}
```

Body Cookies (1) Headers (5) Test Results Status: 200 OK Time: 431 ms Size: 349 B Save Response ▾

Pretty Raw Preview Visualize JSON

```
1 {
  "requestId": "67751985c20e4ac3bbdd47a28413910b",
  "code": 200,
  "msg": null,
  "data": {
    "name": "课程A",
    "id": null,
    "type": 1,
    "sku": 123,
    "class": "com.zhiyong.gateway.facade.model.Course",
    "desc": ""
  }
}
```

### 3.5 方法参数为简单POJO List集合对象

```
1 public class Course implements Serializable {
2     private static final long serialVersionUID = -504495940219884
```

```

5281L;

3
4     private Long id;
5     private String name;
6     private String desc;
7     private int type;
8     private int sku;
9 }
10 List<Course> saveCourse(List<Course> courses);

```

第一步：由于Course是自定义POJO对象，所以需要先到类型管理里面添加Course List对象的定义：

ID	类型名	类型描述	状态	类型结构定义	最后修改人	创建时间	创建人	最后修改时间	最后修改人
1	20 POJO-TestModel-2222	测试POJO类型	草稿	预览		2020/12/04 22:09:27	admin		
2	19 POJO-TestModel-1111	测试POJO类型	草稿	预览		2020/12/04 22:05:43	admin		
3	23 POJO-TestModel-Course-list	测试POJO类型-课程对象list	已发布	预览	admin	2020/12/05 15:18:13	admin	2020/12/05 15:18:29	admin
4	22 POJO-TestModel-Course	测试POJO类型-课程对象	已发布	预览	admin	2020/12/05 15:03:41	admin	2020/12/05 15:04:41	admin
5	21 POJO-TestModel-Teacher[]	测试POJO类型-老师对象数组	已发布	预览	admin	2020/12/05 14:33:39	admin	2020/12/05 14:33:54	admin
6	18 POJO-AddInviteCodeRequest	创建租户邀请码	已发布	预览	admin	2020/12/04 18:51:47	test	2020/12/04 20:50:58	admin
7	17 POJO-TestModel	测试POJO	已发布	预览	admin	2020/12/03 23:51:46	admin	2020/12/04 20:50:48	admin
8	16 POJO-live-list	直播列表对象	已下线	预览	admin	2020/12/03 17:12:53	admin	2020/12/04 20:54:50	admin

```

▼{
  collectionType: "java.util.List",
  name: "testModel",
  type: "com.zhiyong.gateway.facade.model.Course",
  desc: "测试模型"
}

```

第二步：在API里面选择该类型，进行参数配置

**API基本信息**

*API名称:	zhiyong.gateway.api.demo.saveCourse (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名:	com.zhiyong.gateway.facade.api.GatewayTestFacad (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述:	测试demo (课程POJO对象List)	*RPC方法名:	saveCourse
*API分类:	配置Demo	*RPC版本:	1.0.0
*API版本:	2.0	*登录检查:	不需要
*超时时间:	3000 (单位: 毫秒)	*权限检查:	不需要
*请求方式:	POST	*限流次数:	99999 (TPS或QPS)

**API请求参数**

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例
1	courses	课程对象列表	测试POJO类型-课程对象list[POJO-TestModel-Course-list]	请求Body	必填	

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String,UserModel>>|UserModel[]|等POJO及嵌套类型), 需要先到“参数类型”管理菜单配置类型, 然后再选择类型! !

POST http://localhost:19002/router/rest?appKey=ABC123&session=1234&timestamp=2020-11-25 22:35:34&version: Send Save

Params Authorization Headers (10) Body JSON Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

1 [{"name": "课程A", "sku": 1001, "type": 1}, {"name": "课程B", "sku": 1002, "type": 1}]

Body Cookies (1) Headers (5) Test Results Status: 200 OK Time: 316 ms Size: 465 B Save Response

```

Pretty Raw Preview Visualize JSON
1 {
  "requestId": "fed18b9f4fa84765a1b68212d2eb1fa3",
  "code": 200,
  "msg": null,
  "data": [
    {
      "name": "课程A",
      "id": null,
      "type": 1,
      "sku": 1001,
      "class": "com.zhiyong.gateway.facade.model.Course",
      "desc": null
    },
    {
    }
  ]
}

```

### 3.6 方法参数为复杂POJO 单对象

```

1 public class Question implements Serializable {
2     private static final long serialVersionUID = 2515965040073080
3     345L;

```

```

3
4     private Long id;
5     private String indexNo;
6     private int answerNum;
7     private int optionType;
8     private int required;
9     private String desc;
10    private String[] imgUrls;
11    private String[] videoUrls;
12    private Map<String, List<Answer>> answers;
13    private List<List<Answer>> answerAll;
14 }
15
16 Question saveQuestion(Question question);

```

第一步：由于Question是自定义POJO对象，所以需要先到类型管理里面添加Question对象的定义：

ID	类型名	类型描述	状态	类型结构定义	最后修改人	创建时间	创建人	最后修改时间	最后修改人
1	20 POJO-TestModel-2222	测试POJO类型	草稿	预览		2020/12/04 22:09:27	admin		
2	19 POJO-TestModel-1111	测试POJO类型	草稿	预览		2020/12/04 22:05:43	admin		
3	24 POJO-TestModel-Question	测试POJO类型-问答对象（复杂对象）	已发布	预览	admin	2020/12/05 15:47:08	admin	2020/12/05 15:47:20	admin
4	23 POJO-TestModel-Course-list	测试POJO类型-课程对象list	已发布	预览	admin	2020/12/05 15:18:13	admin	2020/12/05 15:18:29	admin
5	22 POJO-TestModel-Course	测试POJO类型-课程对象	已发布	预览	admin	2020/12/05 15:03:41	admin	2020/12/05 15:04:41	admin
6	21 POJO-TestModel-Teacher[]	测试POJO类型-老师对象数组	已发布	预览	admin	2020/12/05 14:33:39	admin	2020/12/05 14:33:54	admin
7	18 POJO-AddInviteCodeRequest	创建租户邀请码	已发布	预览	admin	2020/12/04 18:51:47	test	2020/12/04 20:50:58	admin
8	17 POJO-TestModel	测试POJO	已发布	预览	admin	2020/12/03 23:51:46	admin	2020/12/04 20:50:48	admin
9	16 POJO-live-list	直播列表对象	已下线	预览	admin	2020/12/03 17:12:53	admin	2020/12/04 20:54:50	admin

```

    {
      children: [
        {
          collectionType: "java.util.Map",
          name: "answers",
          type: "com.zhiyong.gateway.facade.model.Answer",
          desc: "答案集合"
        },
        {
          name: "answerAll",
          type: "com.zhiyong.gateway.facade.model.Answer",
          desc: "全部答案"
        }
      ],
      name: "question",
      type: "com.zhiyong.gateway.facade.model.Question",
      desc: "问答对象"
    }
  }
}

```

## 第二步：在API里面选择该类型，进行参数配置

**API基本信息**

*API名称:	zhiyong.gateway.api.demo.saveQuestion (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名:	com.zhiyong.gateway.facade.api.GatewayTestFacad (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述:	测试demo-保存问题对象 (复杂POJO对象)	*RPC方法名:	saveQuestion
*API分类:	配置Demo	*RPC版本:	1.0.0
*Api版本:	1.0	*登录检查:	不需要
*超时时间:	3000 (单位: 毫秒)	*权限检查:	不需要
*请求方式:	POST	*限流次数:	99999 (TPS或QPS)

**API请求参数**

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例
1	question	问题对象	测试POJO类型-问答对象 (复杂对象) [POJO-TestModel]	请求Body	必填	

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String,UserModel>>|UserModel[]等POJO及嵌套类型), 需要先到"参数类型"管理菜单配置类型, 然后再选择类型!!

The screenshot shows a POST request in Postman. The URL is `http://localhost:19002/router/rest?appKey=ABC123&session=1234&timestamp=2020-11-25 22:35:34&version=1.0`. The request body is a JSON object:

```

1  {"indexNo": "1", "answerNum": 4, "optionType": 0, "required": 1, "imgUrls": ["/img/ASDFGHJKL", "/img/ASDFGHJKL"], "answers": {"simple": [{"indexNo": "1", "desc": "adadasdaa13131", "score": 5}, {"indexNo": "2", "desc": "erfghj13131", "score": 5}], "answerAll": [{"indexNo": "2", "desc": "erfghj13131", "score": 5}]}]

```

The response status is 200 OK, time 419 ms, size 891 B.

### 3.5 方法参数为基本类型+复杂POJO单对象

```

1 public class Answer implements Serializable {
2     private static final long serialVersionUID = 2609403705968421
420L;
3
4     private Long id;
5     private String indexNo;
6     private String desc;
7     private int score;
8     private String imgUrl;
9     private String videoUrl;
10 }
11 List<Answer> updateAnswer(Long id, List<Answer> answers);

```

第一步：由于Answer是自定义POJO对象，所以需要先到类型管理里面添加Answer对象List的定义：

致用网关平台

欢迎 admin 修改密码 安全退出

网关文档 | 参数类型 | 添加类型 | 高级查询

ID	类型名	类型描述	状态	类型结构定义	最后修改人	创建时间	创建人	最后修改时间	最后修改人
1	20 POJO-TestModel-2222	测试POJO类型	草稿	预览	admin	2020/12/04 22:09:27			
2	19 POJO-TestModel-1111	测试POJO类型	草稿	预览	admin	2020/12/04 22:05:43			
3	25 POJO-TestModel-Answer-list	测试POJO类型-答案对象list	已发布	预览	admin	2020/12/05 16:08:40	admin	2020/12/05 16:08:56	admin
4	24 POJO-TestModel-Question	测试POJO类型-问答对象 (复杂对象)	已发布	预览	admin	2020/12/05 15:47:08	admin	2020/12/05 15:47:20	admin
5	23 POJO-TestModel-Course-list	测试POJO类型-课程对象list	已发布	预览	admin	2020/12/05 15:18:13	admin	2020/12/05 15:18:29	admin
6	22 POJO-TestModel-Course	测试POJO类型-课程对象	已发布	预览	admin	2020/12/05 15:03:41	admin	2020/12/05 15:04:41	admin
7	21 POJO-TestModel-Teacher[]	测试POJO类型-老师对象数组	已发布	预览	admin	2020/12/04 14:33:39	admin	2020/12/04 14:33:54	admin
8	18 POJO-AddInviteCodeRequest	创建租户邀请码	已发布	预览	admin	2020/12/04 18:51:47	test	2020/12/04 20:50:58	admin
9	17 POJO-TestModel	测试POJO	已发布	预览	admin	2020/12/03 23:51:46	admin	2020/12/04 20:50:48	admin
10	16 POJO-live-list	直播列表对象	已下线	预览	admin	2020/12/03 17:12:53	admin	2020/12/04 20:54:50	admin

```

    {
      collectionType: "java.util.List",
      name: "answers",
      type: "com.zhiyong.gateway.facade.model.Answer",
      desc: "答案对象list"
    }
  
```

## 第二步：在API里面选择该类型，进行参数配置

API基本信息

*API名称:	zhiyong.gateway.api.demo.updateAnswer (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名:	com.zhiyong.gateway.facade.api.GatewayTestFacade (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述:	测试demo-更新答案 (基本类型+复杂POJO对象的组)	*RPC方法名:	updateAnswer
*API分类:	配置Demo	*RPC版本:	1.0.0
*API版本:	1.0	*登录检查:	不需要
*超时时间:	3000 (单位: 毫秒)	*权限检查:	不需要
*请求方式:	POST	*限流次数:	99999 (TPS或QPS)

保存

API请求参数

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例	操作
1	id	问题ID	java.lang.Long	请求URL	必填		
2	answers	答案列表	测试POJO类型-答案对象list[POJO-TestM]	请求Body	必填		

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String, UserModel>>|UserModel[]|等POJO及嵌套类型) , 需要先到"参数类型"管理菜单配置类型, 然后再选择类型!!

保存

POST http://localhost:19002/router/rest?appKey=ABC123&session=1234&timestamp=2020-11-25 22:35:34&version:1.0.0

Params: Authorization: Bearer ABC123, Headers: Content-Type: application/json, Body: [{"indexNo": "2", "desc": "erfghj13131", "score": 5}], Response: Status: 200 OK, Time: 5.30 s, Size: 380 B

```

1 [{"indexNo": "2", "desc": "erfghj13131", "score": 5}]
  
```

```

1 {
2   "requestId": "854d0b5a4947403d9a6c8e4c1df45278",
3   "code": 200,
4   "msg": null,
5   "data": [
6     {
7       "imgUrl": null,
8       "score": 5,
9       "videoUrl": null,
10      "indexNo": "2",
11      "id": null,
12      "class": "com.zhiyong.gateway.facade.model.Answer",
13      "desc": "erfghj13131"
14    }
15  ]
  
```

### 3.7 方法参数为复杂POJO Set集合对象

```

1 public class Module implements Serializable {
2     private static final long serialVersionUID = 1486339284747460
095L;
3     private Long id;
4     private String name;
5     private List<Course> courses;
6     private List<Live> lives;
7 }
8 public class Live implements Serializable {
9
10    private static final long serialVersionUID = -467991267609617
7203L;
11    private Long id;
12    private Course course;
13    private Teacher teacher;
14
  
```

```

15 }
16 Set<Module> saveModule(Set<Module> modules);

```

第一步：由于Module是自定义POJO对象，所以需要先到类型管理里面添加Module对象的定义：

ID	类型名	类型描述	状态	类型结构定义	最后修改人	创建时间	创建人	最后修改时间	最后修改人
1	20 POJO-TestModel-2222	测试POJO类型	草稿	预览		2020/12/04 22:09:27	admin		
2	19 POJO-TestModel-1111	测试POJO类型	草稿	预览		2020/12/04 22:05:43	admin		
3	26 POJO-TestModel-Module-set	测试POJO类型-模块对象Set集合	已发布	预览	admin	2020/12/05 16:36:15	admin	2020/12/05 16:36:34	admin
4	25 POJO-TestModel-Answer-list	测试POJO类型-答案对象list	已发布	预览	admin	2020/12/05 16:08:40	admin	2020/12/05 16:08:56	admin
5	24 POJO-TestModel-Question	测试POJO类型-问答对象（复杂对象）	已发布	预览	admin	2020/12/05 15:47:08	admin	2020/12/05 15:47:20	admin
6	23 POJO-TestModel-Course-list	测试POJO类型-课程对象list	已发布	预览	admin	2020/12/05 15:18:13	admin	2020/12/05 15:18:29	admin
7	22 POJO-TestModel-Course	测试POJO类型-课程对象	已发布	预览	admin	2020/12/05 15:03:41	admin	2020/12/05 15:04:41	admin
8	21 POJO-TestModel-Teacher[]	测试POJO类型-老师对象数组	已发布	预览	admin	2020/12/05 14:33:39	admin	2020/12/05 14:33:54	admin
9	18 POJO-AddInviteCodeRequest	创建租户邀请码	已发布	预览	admin	2020/12/04 18:51:47	test	2020/12/04 20:50:58	admin
10	17 POJO-TestModel	测试POJO	已发布	预览	admin	2020/12/03 23:51:46	admin	2020/12/04 20:50:48	admin

```

{
  collectionType: "java.util.Set",
  children: [
    {
      name: "courses",
      type: "com.zhiyong.gateway.facade.model.Course",
      desc: "课程列表"
    },
    {
      children: [
        {
          name: "course",
          type: "com.zhiyong.gateway.facade.model.Course",
          desc: "课程对象"
        },
        {
          children: [
            {
              name: "teacher",
              type: "com.zhiyong.gateway.facade.model.Teacher",
              desc: "老师对象"
            }
          ],
          name: "lives",
          type: "com.zhiyong.gateway.facade.model.Live",
          desc: "直播列表"
        }
      ],
      name: "modules",
      type: "com.zhiyong.gateway.facade.model.Module",
      desc: "模块集合"
    }
  ]
}

```

第二步：在API里面选择该类型，进行参数配置

**API基本信息**

*API名称:	zhiyong.gateway.api.demo.saveModule (格式: 产品线.业务线.业务类型.操作, API名称+版本号全局唯一)	*RPC接口名:	com.zhiyong.gateway.facade.api.GatewayTestFacad (例如: com.zhiyong.saas.facade.api.SegmentIdQueryFacade)
*API描述:	测试demo-保存模块 (POJO复杂对象)	*RPC方法名:	saveModule
*API分类:	配置Demo	*RPC版本:	1.0.0
*API版本:	1.0	*登录检查:	不需要
*超时时间:	3000 (单位: 毫秒)	*权限检查:	不需要
*请求方式:	POST	*限流次数:	99999 (TPS或QPS)

**API请求参数**

序号	参数名	参数描述	参数类型	参数来源	是否必填	参数示例
1	modules	模块列表	测试POJO类型-模块对象Set集合[POJO-TestModel-Modu]	请求Body	必填	

\*提示: 当参数为POJO复杂参数时 (例如: UserModel|List<UserModel>|Map<<String,UserModel>>|UserModel[]等POJO及嵌套类型), 需要先到"参数类型"管理菜单配置类型, 然后再选择类型!

POST http://localhost:19002/router/rest?appKey=ABC123&session=1234&timestamp=2020-11-25 22:35:34&version: 1.0.0

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

Body (1) Headers (5) Test Results Status: 200 OK Time: 799 ms Size: 1.22 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "name": "模块A",
4     "courses": [{"name": "课程A", "type": 1, "sku": 1001}],
5     "lives": [
6       {"course": {"name": "课程A", "type": 1, "sku": 1001},
7        "teacher": {"name": "张三", "age": 33}}
8     ]
9   },
10  {
11    "name": "模块B",
12    "courses": [{"name": "课程B", "type": 1, "sku": 1001}],
13    "lives": [
14      {"course": {"name": "课程B", "type": 1, "sku": 1001},
15      }
16    ]
}

```

### 3.8 方法参数为多个POJO对象 (不支持)

```

1 int saveModule(String name, List<Course> courses, List<Live> lives
);

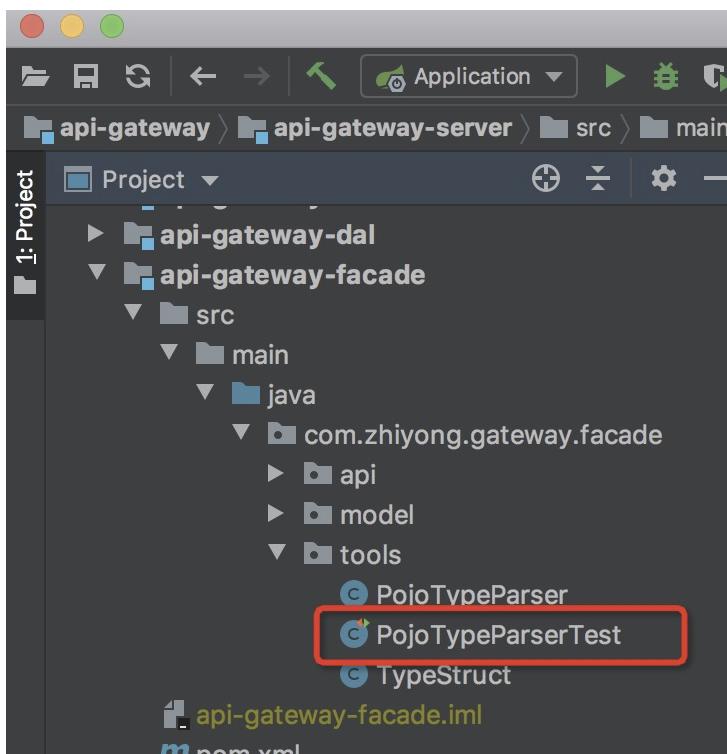
```

网关不支持多个复杂POJO对象的参数，原因：复杂参数只能通过Body请求，但Body只能表示一个参数对象。

## 4 POJO参数类型配置生成工具

除了手工配置，还提供了生成类型配置的工具。当需要添加某个复杂对象的类型配置时，本地main方法运行下工具类即可生成类型配置的JSON，然后将该json粘贴到网关系统中的类型配置中，即可使用了。

- 工具类存放位置



- 工具类调用方法

```
1 public class PojoTypeParserTest {  
2  
3     @Test  
4     public void test() {  
5         String typeName = "questions";  
6         String typeJson = new PojoTypeParser<List<Course>>() {  
7             .run(typeName).toJson();  
8  
9             System.out.println(typeJson);  
10        }  
11    }
```

PojoTypeParser<T> 泛型T就是你需要生成类型配置的类，可以是单对象，也可以是：集合类；例如：  
PojoTypeParser<Course>、PojoTypeParser<List<Course>>、PojoTypeParser<Map<String, Course>>、PojoTypeParser<Course[]>。

不支持的转换类型：PojoTypeParser<Pojo<PojoA, PojoB>>。