目录

**[简介](#_Toc27002_WPSOffice_Level1)** **[2](#_Toc27002_WPSOffice_Level1)**

**[解析规则](#_Toc2977_WPSOffice_Level1)** **[2](#_Toc2977_WPSOffice_Level1)**

**[标识符&对应作用](#_Toc24010_WPSOffice_Level1)** **[3](#_Toc24010_WPSOffice_Level1)**

**[支持类型](#_Toc17826_WPSOffice_Level1)** **[3](#_Toc17826_WPSOffice_Level1)**

**[拆分工具支持拆分类型](#_Toc21842_WPSOffice_Level1)** **[4](#_Toc21842_WPSOffice_Level1)**

**[优势](#_Toc13725_WPSOffice_Level1)** **[4](#_Toc13725_WPSOffice_Level1)**

**[注意事项](#_Toc9106_WPSOffice_Level1)** **[5](#_Toc9106_WPSOffice_Level1)**

**[优化](#_Toc9121_WPSOffice_Level1)** **[5](#_Toc9121_WPSOffice_Level1)**

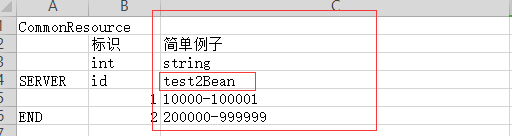
## 

## 简介

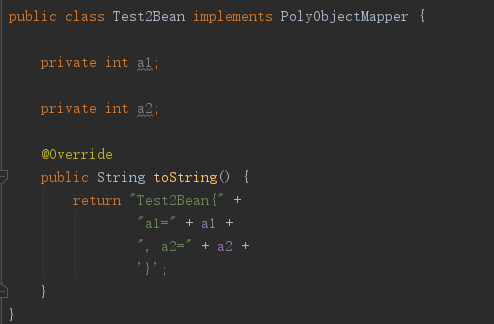
自定义通用解析规则，适用于游戏中的各种数值配置，主要为策划多提供一种新的配置方式，进而解决json配置的臃肿和不方便。

## 解析规则

使用了java的反射机制，先根据配置头的字段对应到配置中的对象，再将对象内的Field将配置反射为对应对象的对象成员，然后再根据不同标识符解析为对应的类型。举个小例子，比如：







在配置中可以直接配置为 10000-100001，10000代表字段a1的值,100001代表字段a2的值

## 标识符&对应作用

- 对象成员分隔符，格式 1-2，待定 {1\_2\_3}、1=2=3，交给策划选择

, array和list分隔符，格式1,2,3....

{} 对象分隔符，在最外围的可以不加，组合对象需加，格式{1-2-3}

: ; map类型使用，格式 key:value1;key:value2;，待定

@ 多态分隔符，格式 XXX@{1-2-3}

## 支持类型

- 单对象解析

- 对象数组解析

- 成员为对象

- 多个成员为对象

- 数组成员为数组List

- 成员为对象数组

- 成员为对象数组，并且list里边的对象再包含一个list

- 数组成员为map

- 数组成员为map map.value为Object

- 数组成员为map map.value为数组

- 数组成员为map map.value为对象数组 并且其中的对象还会再嵌套一层

- 数组成员为map map.value为单对象

- 数组成员为map map.value为数组对象

- 成员为array数组[] 基础类型

- 成员为array数组[] 数组对象

- 多态

- 枚举

## 拆分工具支持拆分类型

单纯字段的拆分、

bean(在内层)的拆分、

map(在最外层)中key和value的拆分、

map(在内层)中key和value的拆分、

list/数组(在最外层)的拆分、

list/数组(在内层)的拆分

## 优势

通用解析支持多态，在配置中便可以指定要转换的bean是哪种，因此不需要在项目中自己根据枚举类型进行转换。比如Consume会根据配置的枚举生成不同的IConsume子类，而使用通用解析则可以直接在配置的时候指定类型，无需在启动的时候二次解析。

json配置啰嗦，策划嫌麻烦。

## 注意事项

基于通用解析的配置：List的泛型参数不支持List（标识符重复无法拆分），Map的泛型参数不支持Map（标识符重复无法拆分）；

配表格式需要由开发人员先定义好，而不是交给策划。

拆表工具：每个Field只支持单层拆分（多层拆分配置无法控制）；

## 优化

2019-04-03

1.1、支持json格式自动解析，现有的t5已支持

1.2、bean自身的私有解析私有定制，新增了SelfParser 接口、@ParseIgnore注解，看业务需要，实现接口，重写doParse方法，该方法会在构造bean的最后被调用，例子：test5Bean

1.3、复杂类型直接转成所需要的对象，查看 通用解析.特殊类 型标黄色部分

1.4、枚举的反射，直接根据id反射成枚举。例子：test6Bean

1.5、拆表工具优化，目前新增支持，查看 通用解析.转表拆分-总览表

## 待优化

多态类名太麻烦，是否加个枚举处理掉；

写出一份模板，作为范本使用的模板参照，简单明了描述该如何配置，并且给出配置例子；

是否还要支持Date类型的配置，不用支持；

TypeEnum 优化代码，查看spring内部的converter处理。