

选择可复用构件

- [各组仓库地址](#)
- [CM构件](#)
- [PM构件](#)
- [License构件](#)

各组仓库地址

小组	仓库地址
1	https://github.com/Wheellllll/Software-Reuse-Practice-One
2	https://github.com/anzhehong/Software-Reuse
3	https://github.com/Gavin96/SoftwareReuse
4	https://github.com/BrunoQin/Open-Reuse
5	https://github.com/SummerWish/Tahiti
6	https://github.com/BelleZhao/Ericsson-SoftwareReuse
7	https://github.com/2016-Software-Reuse-Group-7/course-teamwork-1
8	https://github.com/Yiinsh/MessageTransferModel
9	本组
10	https://github.com/ShiehShieh/software-reuse-group
11	https://github.com/WeiyiChen/SoftwareReuse-Project
12	-

CM构件

选择构件 -- Team5

[Team5](#)

选择理由

小组	说明	选择与 否	理由
Team1	支持ini格式以及动态加载，可以读取并写入配置	×	项目原本采用的是json配置，想要在不修改配置文件类型的情况下进行使用
Team2	支持Json格式	×	每次使用都要指定文件路径，且不支持嵌套配置的读取
Team3	支持Json格式	×	参数已经确定，扩展性差
Team4		×	文档不知道怎么打不开呀= =
Team5	支持Json和Yaml两种格式	✓	配置文件读取是直接读成Java对象，这与我们之前没有构建CM构件时的方法相同，也就是说我们之前已经创建了Config这个Java bean了，使用起来就很方便
Team6	文件读取方式	×	不支持我们的配置文件格式
Team7		×	参数已经确定，扩展性差
Team8		×	文档太简单了
Team10	支持json	×	不支持嵌套配置的读取
Team11	支持json	×	不支持嵌套配置的读取

改进建议

- 无法读取嵌套配置
- 可以添加更多的接口，不仅可以直接读成Java bean，也可以根据键来读值

PM构件

选择构件 Team5

选择理由

1. 方便使用，与系统耦合低，可以很方便使用到我们的构建中，因为我们的pm部分是由一个单独的类处理，整体改动就非常少
2. 尝试过第4组的pm，但是不能自定义字段，只能记录成功与失败，无法很好的适应该项目

改进建议

希望能够自自定义输出路径或者格式

License构件

选择构件——Team2

选择理由

与本项目原构件使用方法基本一致，无需改变原代码结构，使用方便。

改进建议

- 对参数提供set和get方法，使用起来就可以更加的灵活。
- 改变一下类名，MaxNumOfMessage这个类听起来不像是一个限制器，更像是一个数值，容易让人混淆。