

Tahiti 构件选择文档

说明

我们组在比较了以下小组的构件后，最终选择了我们认为最好的构件：

小组	项目地址
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	(Self)
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	https://github.com/bookish-component
10	https://github.com/ShiehShieh/software-reuse-group
11	https://github.com/WeiyiChen/SoftwareReuse-Project
12	N/A

选择

构件	选择
配置管理 (CM)	Team1
性能监控 (PM)	Team1
许可证 (License)	Team1

构件选择：配置管理 (CM)

选择原因

Team1:

1. 支持自定义配置加载路径
2. 不限定配置内容，具有很高的可复用性
3. 支持监听配置文件的修改
4. 文档全面，含有示例，方便使用

其他组存在的问题

Team2: 文档中 maven 安装方式本质是源码人工安装；只支持字符串 JSON，不如直接用 fastjson

Team3: 接口使用方式复杂，不如 Team1 简洁，也不如直接用 fastjson

Team4: 未找到构件

Team6: 接口命名不规范

Team7: 限定了配置格式和字段，没有可定制性，无法复用

Team8: 文档不全，没有接口文档；限定了配置格式和字段，无法复用

Team9: 没有问题，因为 Team1 排在前面所以选了 Team1

Team10: 没有文档；没有 jar；限定了格式和字段，无法复用

Team11: 没有找到使用说明文档，不会用

构件选择：性能监控 (PM)

选择原因

Team1:

- 接口明确，功能全面，可以指定输出地址和输出文件名，指定输出时间，并且可以自定义指标，例如灵活地添加或者删除登录成功和登录失败的指标，根据实际的变化改变指标值。
- 代码简洁，架构清晰。但是代码里面缺少注释，易读性不是很强，希望后期更新增加注释。

- 可扩展性强，利于二次开发，裁剪或者增加新功能。
- 该可复用模块的文档简明易懂，易于理解使用。

其他组存在的问题

Team2:

- Maven 安装方式本质是源代码安装；
- 性能监控不完全符合需求

Team3:

- 文档不全

Team4:

- 无法指定监控指标
- 文档代码示例不清楚

Team6:

- 支持监控的指标类型不足

Team7:

- 功能完善，文档详细
- 有附加测试情况

构件选择：许可证 (License)

选择原因

Team1:

选择的该构件满足本项目的系统与技术需求并且容易实施，具有质量较高的可复用对象 License，并且文档条理清晰。

该构件提供capacity与throughput两种license功能。其中throughput还可改变默认值的时间单位。许可服务use()方法返回的结果涵盖了所有可能-AVAILABLE（license可用）、CAPACITYEXCEEDED（容量超过限制）、THROUGHPUTEXCEEDED（流量超过限制）、BOTHEXCEEDED（容量流量均超过限制），方便我们做出对应的处理。