

余乐兑技术管理手册

V 1.0

编写：姜晓冬 更新日期：2024-11-22

概述

本手册旨在统一本公司技术管理思想，统一技术工作手段，促进工作协同，保证工作有序开展，特将相关规定、规范等汇编成册，悉知遵守。

指导原则

- 一. 本司奉行结果导向原则，因此各项工作务必遵循事先计划、事中留痕、准时完成、保证质量、事后总结之原则开展。以具体产品开发工作为例，收到分派任务后，相关责任人员应首先评估工作内容、工作要求、工作条件，制定具体工作计划，包括工作内容具体事项和预估完成时间等，准备好代码仓库和测试环境，协调好其他相关部门和人员，然后方可按照计划展开工作。期间所有工作内容应以书面形式及时提交到仓库或工作沟通群，包括但不限于每日工作日志记录、工作计划表、源代码、流程图、思维导图、数据字典、会议和沟通协调记录、信息截图等。当研发工作到达里程碑时点时，应及时对前期工作成果进行整理，包括整理和归档源代码、文档、过程记录信息等，并对下一阶段工作进行展望和预计。
- 二. 为方便代码管理和协同开发，工作人员需认真遵守相关工作规范，包括但不限于工作日志书写、工作计划表格、代码文件命名、代码工程目录结构、仓库命名等规范。虽然某些规范可以归属于个人工作习惯范畴，但为了统一行动，减少沟通成本，工作人员仍须配合遵守。
- 三. 本司以自主产品开发为目标，产品模块开发完成后，须遵守一定的流程进行严格测试后方可部署到生产环境。例如，对生产环境进行部署、变更、升级等操作，均需要有领导签字的书面同意书方可操作。

四. 各工作人员务须具备高度责任心，保质保量、按时做好自己职责范围内工作，同时应主动帮助同事，及时查漏补缺，提出合理化建议，对整个产品研发工作做出贡献。

代码仓库使用管理规范

一. 本规范旨在规范代码仓库（GIT）的使用和管理。

二. 仓库命名和信息：

1. 仓库名称应遵循以下规则：汉字名称+英文工程名称+版本说明，其中：

- (1) 汉字名称应简明扼要地概括描述仓库内容，如“微信收款小程序后端API服务”、“余乐兑运营后台管理Admin（UI+API服务）”；
- (2) 英文工程名称应与代码中工程名称保持一致。
- (3) 版本应表明代码用途，如“技术测试v1”、“正式版v1”或“测试v1（已废弃）”等。对于测试或实验性质的代码，必须加注“已废弃”、“测试通过”或“测试未通过”等字样。

三. 仓库说明信息（即About信息）：

1. 仓库说明信息应遵循以下原则：

- (1) 应简明扼要的概括说明该仓库内内容；
- (2) 应说明该仓库中是否包含测试或演示信息；
- (3) 应说明该仓库中是否包含测试和部署脚本或说明；
- (4) 应说明该仓库中是否包含已编译的Release版本；
- (5) 应说明该仓库主要编程语言；
- (6) 应说明该参数是否废弃或正在使用等生命周期状态；
- (7) 应说明该仓库中是否包含相关技术文档；

2. 示例：“本仓库下为余乐兑运营后台管理Admin项目代码，其中Doc目录下为所有相关设计文档，Install目录下为在Ubuntu系统的安装部署脚本和Conf文件，Backend目录下为后端服务代码，编程语言为Go，Frontend目录下为前端UI代码，编程语言为Uniapp，Release下为预编译可执行文件，Data目录下为数据库建库SQL，仓库中不包含测试和演示数据，详细说明见Readme.md。该仓库正在使用。”。
3. 仓库属性应设置为Public，协议为空。
4. 仓库信息及代码状态应及时提交和更新。

四. 仓库管理要求：

1. 没有特殊需要，禁止将工作代码托管到公网上的免费Git服务。
2. 没有特殊理由，不得删除公司服务器上的代码仓库。已废弃仓库应及时修改仓库名称和说明信息。
3. 没有特殊理由，不得擅自更新与己无关的仓库。
4. 禁止冒用他人账号登录公司Git服务器。

Go代码规范

一. 文件命名：

1. 文件名使用Pascal命名方式。
2. 类定义文件名称为XxxClass.go，例如UserClass.go，WebServerClass.go等。
3. 主程序文件名称为main.go（首字母小写）。
4. 常量定义文件名称为Consts.go。
5. 接口定义文件名称为Intf.go。
6. 公共函数文件名称为PublicFuncs.go（用于存放所有不便分类的函数）。

7. 独立设置文件名称为config.toml（框架默认文件名）。

二. 变量命名：

1. 变量名采用Pascal命名方式（首字母大小写遵循Go语言规定）。
2. 类名（Type）使用大写字母T开头，例如TUser，TWebServer等。
3. 参数使用大写字母A开头，例如func Test(AValue string)。
4. 局部变量使用小写字母m开头，例如var mPassword string。
5. 常量使用大写字母C开头，例如const CMainUrl = “http://”
6. 全局变量使用大写字母G开头，例如var GOrderTopic string = “order”
7. 变量名应尽可能使用可辨识意义的英文单词或缩写，特殊情况下允许使用汉语拼音的全拼，但禁止使用拼音首字母缩写。

三. 文件头信息：

1. 代码文件头部应加入文件头信息（可在IDE中设置默认信息），其中至少需要标明：
 - (1) 文件说明信息；
 - (2) 代码创建人信息；

四. 工程文件夹：

1. 工程根目录下放置：
 - (1) 入口主程序main.go；
 - (2) Go.mod
 - (3) config.toml
 - (4) readme.md

- (5) 其他必须放置在根目录下的文件。
- 2. App子目录下放置：
 - (1) 所有类定义文件；
 - (2) 所有函数文件；
- 3. Data子目录下放置：
 - (1) 本地数据库文件（如需要使用本地数据库的话）；
 - (2) 数据库建库SQL脚本（如需要的话）；
 - (3) 数据库维护SQL脚本（如需要的话）；
- 4. Static目录下放置：
 - (1) 所有用到的Js、Css和资源文件（如需要的话）；
 - (2) 需要提供Web访问的静态文件（如需要的话）；
- 5. Template目录下放置：
 - (1) 所有用到的html模版文件，扩展名为html；
 - (2) 其他与gView渲染有关的模版文件；
- 6. Build目录下放置：
 - (1) 预编译的可执行文件
 - (2) 可执行文件运行所需的其他文件
 - (3) Makefile（如果需要的话）
- 7. Doc目录下放置：
 - (1) 所有说明文件，Markdown格式，扩展名为.md；

8. l18n目录下放置：

(1) 所有需要的语言文件，toml格式，扩展名为.toml；

9. Plugins目录下放置：

(1) 所有需要用到的Plugin可执行文件及运行所需文件（如需要的话）；

10. 其余子目录根据需求自行创建，但需要在Readme.md里注记说明。

五. 编程建议：

1. 多使用逻辑流程图、数据流程图或对象关系图，理清逻辑后再编程；
2. 尽量使用面向对象式设计，将可重用的数据结构和逻辑过程封装成对象；
3. 如果对象需要读写数据库，把方法封装到对象中；
4. 从输入输出开始设计对象的内部逻辑；
5. 尽量把对象的导入导出写成显式过程，避免使用指针Scan；
6. 为对象编写单独的测试程序；
7. 尽量将常量抽离出来单独定义或写在配置文件中，避免在代码里写死，如数据库名、表名、字段名、缓存Key名等，如果是对象，使用属性进行配置；
8. 避免直接使用SQL语言进行数据库操作，尽量不要使用SQL方言，尽可能使用ORM方法进行数据库操作；
9. 在合理且可选择情况下，尽量使用进程实现并行计算；
10. 使用独立的网络消息协调和调度进程是最简单的方法；
11. 使用if..else或其他语言技巧避免逻辑逃逸；
12. 避免直接使用Interface{}类型；
13. 关键业务处理环节前后都要记录日志；

14. 使用JSON字符串在进程间传递数据；
15. 在可选择情况下尽量使用简单、易于理解的技术；
16. 尽量在Linux操作系统环境中进行测试；
17. 谨慎使用指针；
18. 谨慎使用异步计算；
19. 谨慎使用数据库的Join操作；
20. 谨慎使用数据库的视图、存储过程；
21. 使用新的开源库之前必须进行测试和实验，并做好记录；
22. 在每个函数、方法、定义前面书写注释；
23. 在每个逻辑分支的入口书写注释；
24. 先实现MVP（最小可用原型），然后再完善和丰富。

工作日志规范

- 一. 工作日志以Markdown格式书写，旨在记录每天工作进展，总结遇到的问题及解决方案，规划后面的具体工作内容，是帮助个人改善工作状况、加强沟通和协调、加强计划性、养成良好工作习惯、提高自身能力的有益工具。
- 二. 每人建立一个Git仓库，名称为“XXX工作日志”，以便保存个人工作日志。
- 三. 工作日志一般每月创建一个MD文件，名称为“XX年XX月工作日志”。
- 四. 工作日志书写格式参照Markdown书写格式，每段标题为日期（#），正文则为当日工作情况。
- 五. 通常情况下正文记录以下信息：
 1. 自己在当天对哪个项目做了哪些工作，大致描述即可；
 2. 自己在当天工作中遇到了哪些问题并如何解决；

3. 预估后面的工作内容或工作重点。

六. 每天下班前应及时提交工作日志到仓库。

七. 示例

2024年10月22日

* 今天完成了TWebServer、TUser和TDocument类，测试通过，代码提交到仓库《MyResume V1 MVP版》；

* 遇到主要问题：

* * 为减少保存数据库麻烦，改为直接保存JSON对象到数据库（SaveToDb方法）；

* * 保存数据时使用SHA证书加密（加密方法在SaveToDb方法里）；

* 下步规划：

* * 准备迭代TDocument类，增加导入导出方法和消息触发机制。