**实战项目-Go 语言笔记服务**

**一、课程背景与目标**

**课程背景**

在大家已经学完前 5节 Go 原理与实践课程的基础上，通过项目实战帮助大家把前面学过的知识应用起来

**课程目标**

* 将前面所学的知识应用到项目中
* 熟悉项目的代码,可以将项目正常运行
* 熟悉 Hertz/Kitex/Gorm 的使用

**二、课前了解**

**安装 Docker/Postman/Git**

* 安装 Minikube 或 Docker Desktop [安装教程](https://www.runoob.com/docker/windows-docker-install.html)
* 可以使用 Minikube 或者使用 Docker Desktop 启动 Docker
* 安装 Postman
* 安装 Git [安装教程](https://www.liaoxuefeng.com/wiki/896043488029600/896067074338496)

**Hertz 初体验**

通过阅读https://www.cloudwego.io/zh/docs/hertz/getting-started/ 尝试运行 Hertz 的示例代码

* Hertz 框架地址: https://github.com/cloudwego/hertz

**Kitex 初体验**

通过阅读 https://www.cloudwego.io/zh/docs/kitex/getting-started/ 尝试运行 Kitex 的示例代码

* kitex 暂时没有针对 Windows 做支持，如果本地开发环境是 Windows 建议使用 [WSL2](https://docs.microsoft.com/zh-cn/windows/wsl/install)

**Gorm 初体验**

通过阅读 https://gorm.cn/docs/#Install 尝试运行 Gorm 的示例代码

**了解 Etcd 和 Opentracing**

了解 etcd 是什么以及 opentracing 是什么

**三、项目介绍**

**项目简介**

EasyNote 提供了一套比较完整的笔记后端API服务.

|  |
| --- |
| 项目地址 https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy\_note |

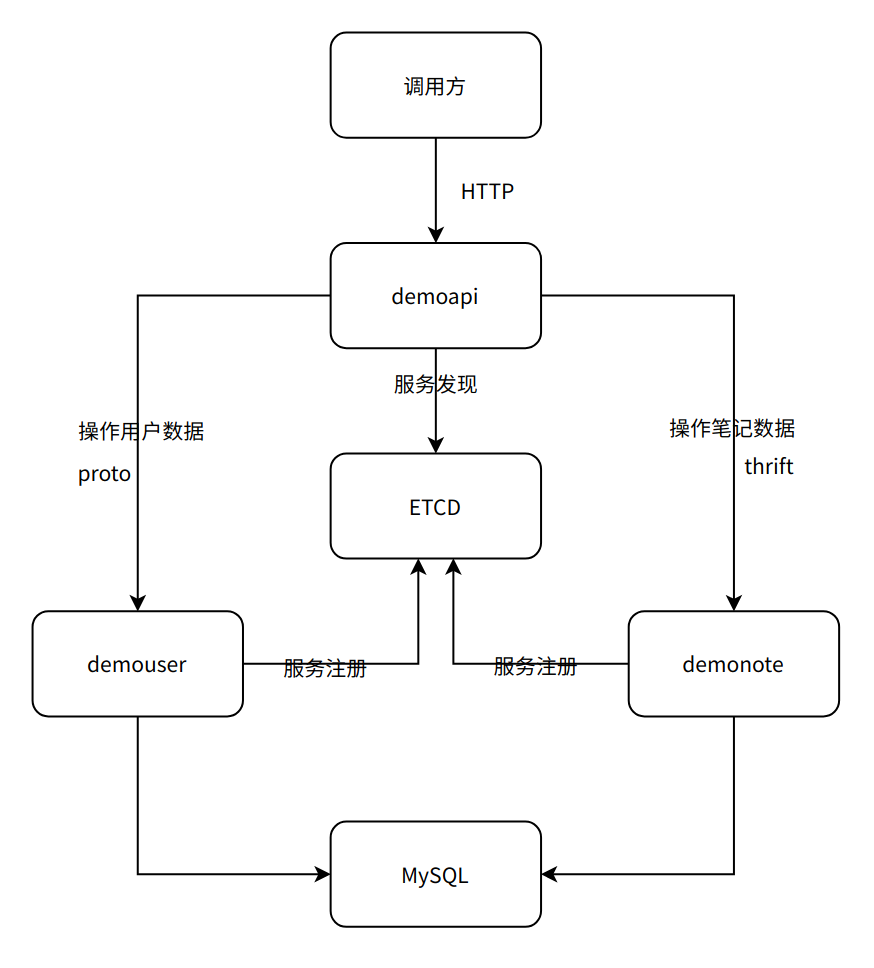
|  |
| --- |
| 推荐版本 Golang >= 1.15 |

**项目模块介绍**

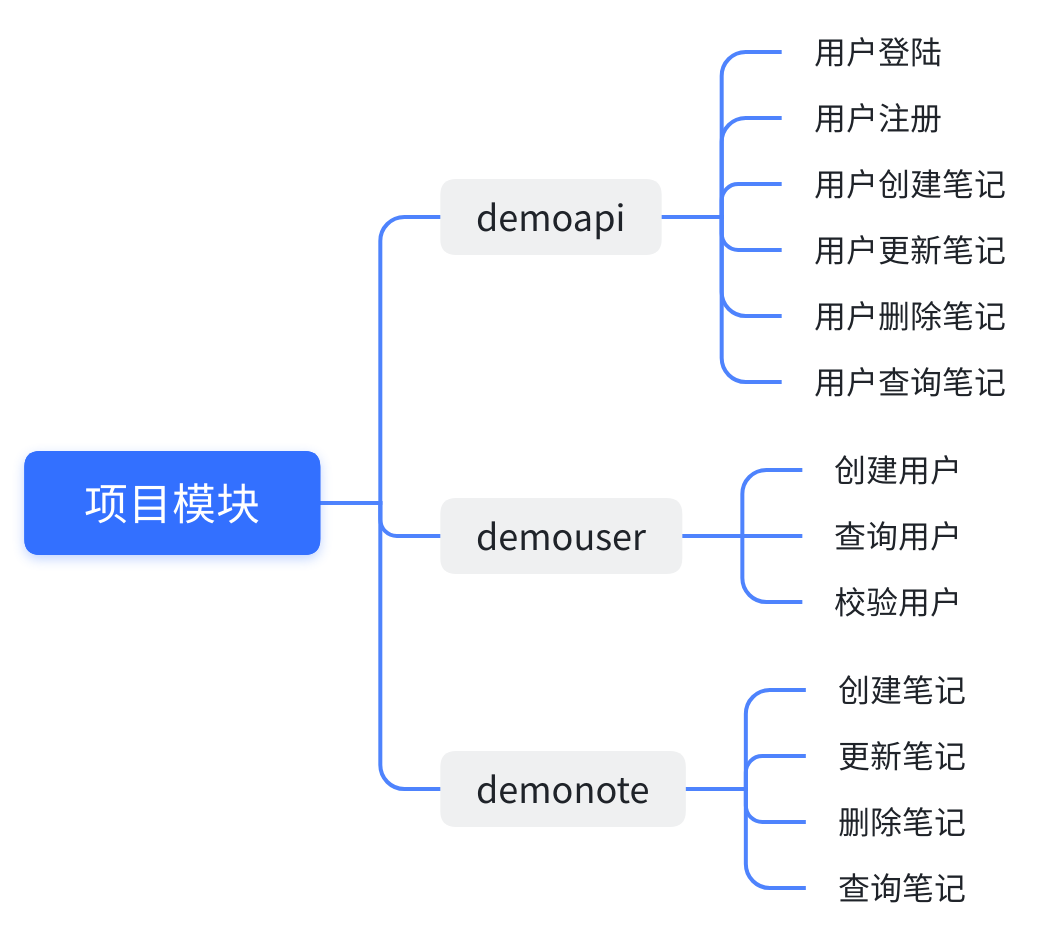


**Click the image to view the bitable.**

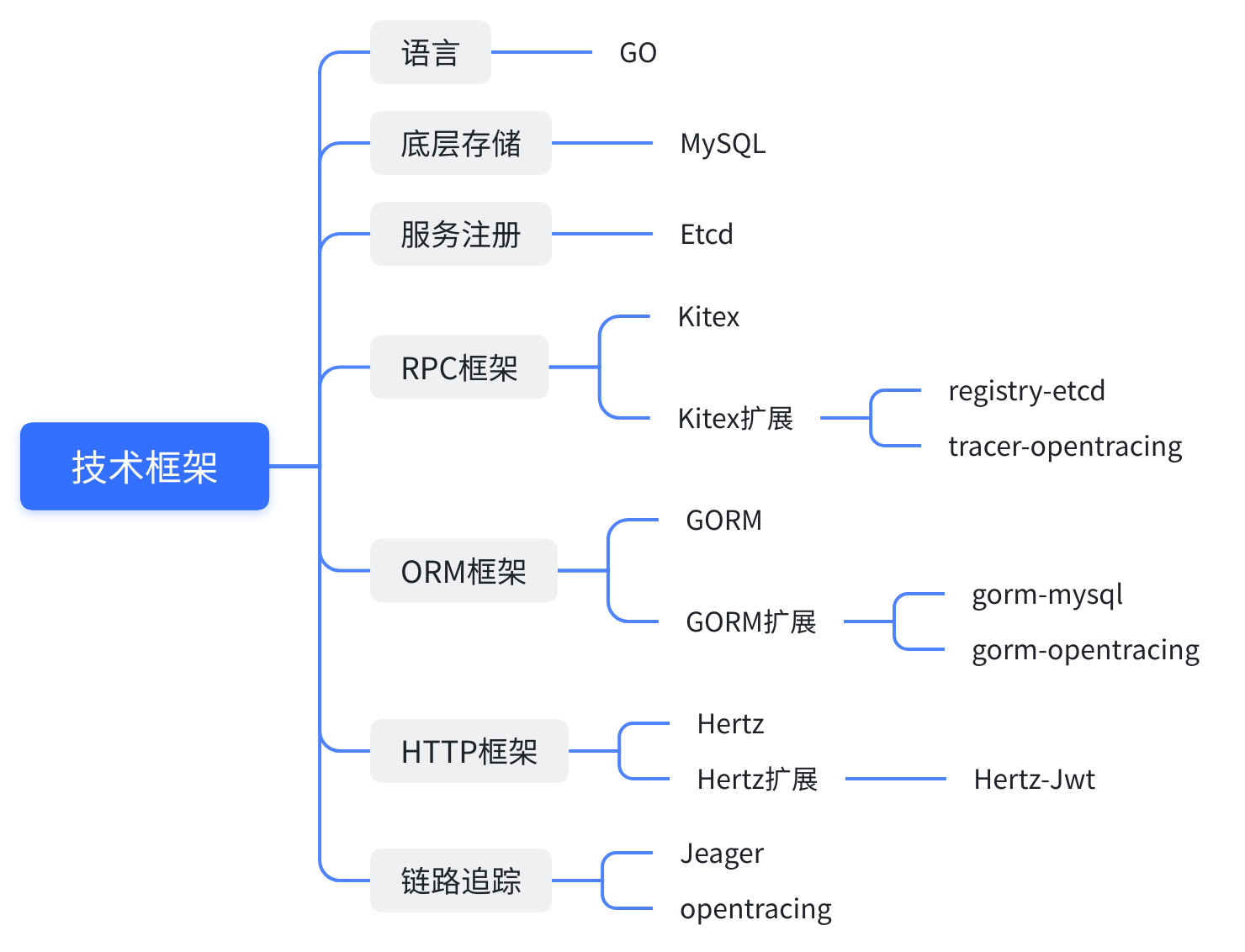
**项目服务调用关系**



**项目模块功能介绍**



**项目技术栈**



**项目相关的使用框架资料**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 框架文档地址 | github地址 | 拓展文档 |
| RPC框架Kitex | 框架文档 | https://www.cloudwego.io/zh/docs/kitex/overview/ | https://github.com/cloudwego/kitex |  |
| Kitex-etcd扩展 | https://github.com/kitex-contrib/registry-etcd | https://github.com/kitex-contrib/registry-etcd | * https://www.cloudwego.io/zh/docs/kitex/tutorials/framework-exten/registry/ * https://www.cloudwego.io/zh/docs/kitex/tutorials/framework-exten/service\_discovery/ |
| Kitex-OpenTracing扩展 | https://www.cloudwego.io/zh/docs/kitex/tutorials/service-governance/tracing/ | https://github.com/kitex-contrib/tracer-opentracing | * https://www.cloudwego.io/zh/docs/kitex/tutorials/framework-exten/middleware/ |
| ORM框架Gorm | 框架 | https://gorm.cn/zh\_CN/ | https://github.com/go-gorm/gorm |  |
| Gorm-Opentracing扩展 | https://github.com/go-gorm/opentracing | https://github.com/go-gorm/opentracing | * https://gorm.cn/zh\_CN/docs/write\_plugins.html |
| HTTP框架Hertz | 框架 | https://www.cloudwego.io/zh/docs/hertz/ | https://github.com/cloudwego/hertz |  |
| Hertz-JWT扩展 | https://www.cloudwego.io/zh/docs/hertz/tutorials/basic-feature/middleware/jwt/ | https://github.com/hertz-contrib/jwt |  |

**四、项目代码介绍**

**项目代码目录结构介绍**

|  |  |  |  |
| --- | --- | --- | --- |
| [idl](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/idl) | [thrift](https://github.com/cloudwego/kitex-examples/blob/main/bizdemo/easy_note/idl/note.thrift)/[proto](https://github.com/cloudwego/kitex-examples/blob/main/bizdemo/easy_note/idl/user.proto) 接口定义文件 |  | 文档/子目录介绍 |
| [kitex\_gen](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/kitex_gen) | Kitex自动生成的代码 |  |  |
| [pkg](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/pkg) | [constants](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/pkg/constants) | 常量 |  |
| [errno](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/pkg/errno) | 错误码 | [关于错误码的讨论](https://github.com/cloudwego/kitex/discussions/248) |
| [middleware](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/pkg/middleware) | Kitex的中间件 | [Kitex Middleware 扩展](https://www.cloudwego.io/zh/docs/kitex/tutorials/framework-exten/middleware/) |
| [bound](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/pkg/bound) | Kitex Transport Pipeline-Bound 扩展 | [什么是Kitex的Transport Pipeline-Bound 扩展](https://www.cloudwego.io/zh/docs/kitex/tutorials/framework-exten/trans_pipeline/) |
| [tracer](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/pkg/tracer) | Jarger 初始化 |  |
| [cmd](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/cmd) | [api](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/cmd/api) | demoapi服务的业务代码 | * handlers : 封装了 api 的业务逻辑 * rpc : 封装了调用其它 rpc 服务的逻辑 |
| [note](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/cmd/note) | demonote服务的业务代码 | * dal : 封装了数据库的访问逻辑 * service: 封装了业务逻辑 * rpc : 封装了调用其它 rpc 服务的逻辑 * pack : 数据打包/处理 |
| [user](https://github.com/cloudwego/kitex-examples/tree/main/bizdemo/easy_note/cmd/user) | demouser服务的业务代码 |

**项目运行**

**运行基础依赖**

|  |
| --- |
| Bash docker-compose up |

执行上述命令启动 MySQL、Etcd、Jaeger 的 docker 镜像

**运行 demonote 服务**

|  |
| --- |
| Bash cd cmd/note  sh build.sh  sh output/bootstrap.sh |

**运行 demouser 服务**

|  |
| --- |
| Plaintext cd cmd/user  sh build.sh  sh output/bootstrap.sh |

**运行 demoapi 服务**

|  |
| --- |
| Bash cd cmd/api  chmod +x run.sh  ./run.sh |

**参考文档**

* https://www.cloudwego.io/zh/docs/kitex/
* https://github.com/kitex-contrib/registry-etcd
* https://github.com/kitex-contrib/tracer-opentracing
* https://github.com/cloudwego/kitex
* https://gorm.io/docs/index.html
* https://github.com/go-gorm/gorm
* https://github.com/go-gorm/opentracing
* https://github.com/cloudwego/hertz
* https://github.com/hertz-contrib/jwt