

代码生成工具v1.2开发-中间件构架支持

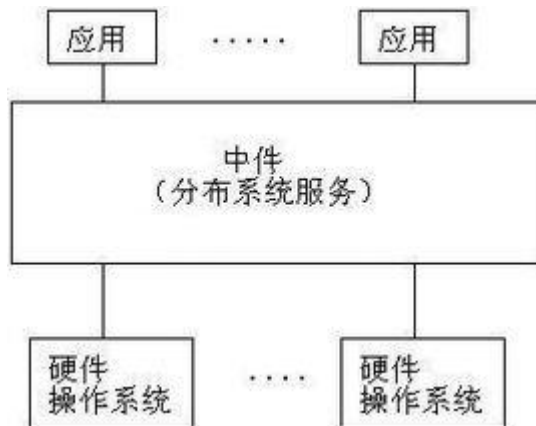
联系QQ: [2816010068](#), 加入会员群

目录

- 中间件是什么？
- 中间件设计
- 中间件代码开发

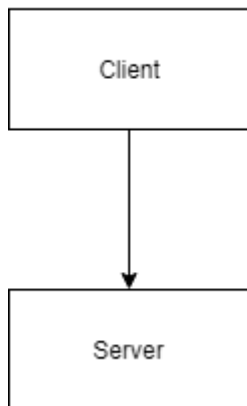
中间件是什么？

- 中间件：是一类连接软件组件和应用的计算机软件，它包括一组服务。以便于运行在一台或多台机器上的多个软件通过网络进行交互。该技术所提供的互操作性，推动了一致分布式体系架构的演进，该架构通常用于支持并简化那些复杂的分布式应用程序，它包括web服务器、事务监控器和消息队列软件。

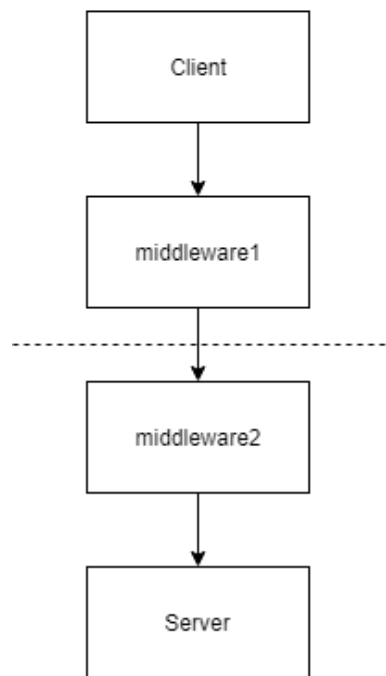


Kaola框架的中间件

- client-server执行流程



- 中间件

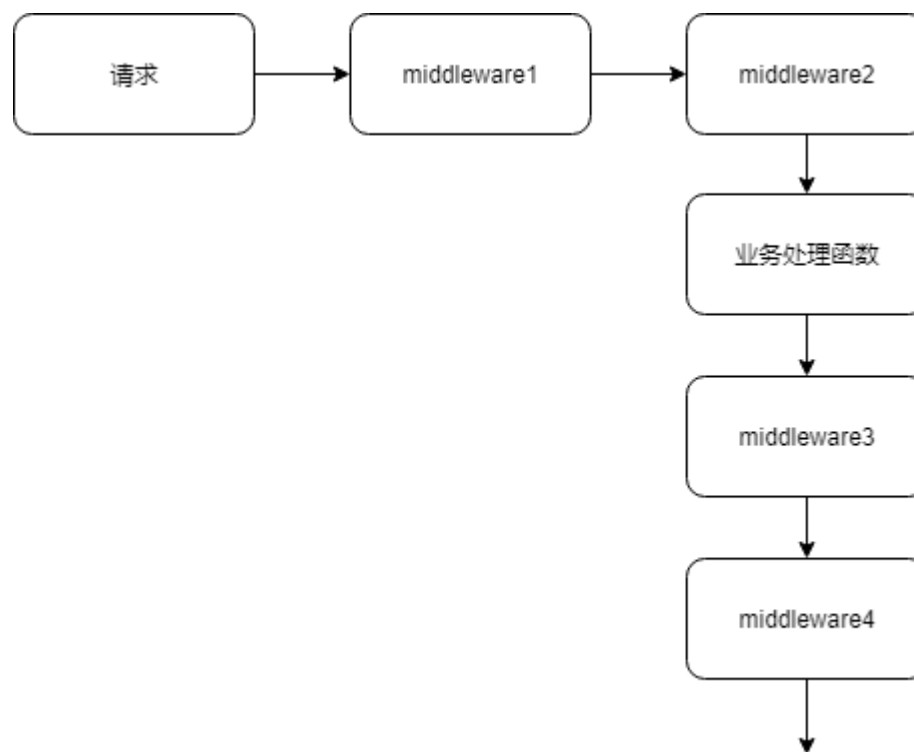


Koala的中间件

- 在客户端调用rpc函数到请求路由到服务端的处理函数中间执行的通用的组件
 - 客户端中间件
 - Rpc访问日志
 - 负载均衡
 - 熔断限流
 - ...
 - 服务端中间件
 - 服务访问日志
 - Metrics打点
 - 限流
 - ...

Koala中间件设计

- 中间件流程图



koala中间件设计

- grpc处理函数

```
//SayHello函数的实现
func (s *SayHelloController) Run(ctx context.Context, r*hello.HelloRequest) (
    resp*hello.HelloResponse, err error) {
    return
}
```

koala中间件设计

- 中间件处理函数设计

```
type MiddlewareFunc func(ctx context.Context, req interface{}) (resp interface{}, err error)
```

- 中间件设计要点
 - 需要把所有中间件串联起来，依次执行
 - 每个中间件可以选择执行的时机，比如在其他中间件之前执行，或者在其他中间件之后执行

```
type Middleware func(MiddlewareFunc) MiddlewareFunc
```


koa中间件设计

- 把所有的中间件串成一条链

```
// is created as the outermost middleware.  
func Chain(outer Middleware, others ...Middleware) Middleware {  
    return func(next MiddlewareFunc) MiddlewareFunc {  
        for i := len(others) - 1; i >= 0; i-- { // reverse  
            next = others[i](next)  
        }  
        return outer(next)  
    }  
}
```

koala中间件设计

- 支持用户自定义中间件

```
func Use(m ...Middleware) {  
    userMiddleware = append(userMiddleware, m...)  
}
```

中间件开发

- 和代码生成器进行整合
 - 把处理函数抽象成MiddlewareFunc类型
 - 从框架中获取当前所有的中间件，进行调用

中间件开发

- 测试
 - 写一个自定义中间件，进行测试