**Pledge**

**技术栈**：Golang、WebSocket、KuCoin API、Redis、区块链集成**主要职责与成就**：

1. **加密货币行情集成**：

* 实现了与KuCoin交易所API的深度集成，通过WebSocket实时获取PLGR-USDT等交易对价格数据
* 设计了价格缓存系统，利用Redis存储最新价格数据，确保系统在API连接中断时仍能提供服务

1. **实时数据推送服务**：

* 开发了高性能WebSocket服务器，支持上千客户端同时连接，实时推送加密货币价格变动
* 实现了完善的心跳检测机制（ping/pong），确保连接的可靠性和即时检测断开连接

1. **系统架构设计**：

* 将系统拆分为API服务和定时任务两个核心组件，提高了系统的可维护性和扩展性
* 实现了模块化的代码组织结构，将业务逻辑、数据访问和网络通信清晰分离

1. **性能优化**：

* 优化了WebSocket消息处理逻辑，实现了毫秒级的实时价格推送，满足高频交易场景需求
* 通过使用Go的goroutine和channel机制，实现了高效的并发处理，显著提升了系统吞吐量

**基于pledge-backend项目的简历内容**

**项目描述（50字左右）**

参与开发基于区块链技术的加密资产质押平台后端，实现实时价格获取、WebSocket推送和智能风控系统，为用户提供安全高效的质押服务。

**技术栈**

* **编程语言**：Golang
* **Web框架**：Gin（从错误信息中可见）
* **实时通信**：WebSocket
* **数据存储**：Redis
* **第三方集成**：KuCoin API
* **日志系统**：基于结构化日志（代码中有log.Logger.Sugar()引用）
* **并发处理**：Goroutines，Channels，sync.Map

**项目职责**

1. **加密货币行情集成**

* 实现与KuCoin交易所的WebSocket API集成，获取PLGR-USDT实时价格数据
* 设计并实现价格数据缓存系统，使用Redis存储价格信息，确保系统可靠性

1. **实时数据推送服务**

* 开发高性能WebSocket服务，支持客户端实时接收价格更新
* 实现完善的心跳检测和超时机制，保证连接稳定性和资源合理利用
* 设计高效的广播系统，实现价格变动及时推送给所有连接客户端

1. **系统架构设计**

* 参与设计分离式架构，将系统拆分为API服务和定时任务两个核心组件
* 实现模块化的代码组织结构，使业务逻辑、数据访问和网络通信清晰分离

**技术成就**

1. **高并发处理**

* 使用Golang的goroutines和channels机制，设计了能支持千级并发连接的WebSocket服务
* 通过sync.Map实现了线程安全的客户端连接管理，优化了广播性能

1. **系统可靠性**

* 实现了完善的错误处理和自动重连机制，确保价格数据获取的连续性
* 通过Redis缓存机制，解决了API暂时不可用时的数据持续可用问题
* 设计精细的资源管理策略，防止内存泄漏和资源耗尽

1. **性能优化**

* 优化了WebSocket消息处理逻辑，实现了毫秒级的实时价格推送
* 通过精细化的锁管理和内存优化，降低了系统资源消耗，提高了处理能力
* 实现了高效的价格广播机制，能够同时支持大量客户端的实时数据推送

EasySwap

EasySwap是基于区块链的去中心化NFT交易平台，采用订单簿撮合模式，提供高效、安全的NFT交易服务，支持固定价格交易、批量操作和多种NFT标准。

* **后端API开发**：使用Go语言开发了EasySwap NFT交易平台的核心后端服务，包括订单管理、用户认证和区块链事件监听系统
* **中间件开发**：设计并实现了关键中间件组件，包括缓存API(cacheapi.go)、认证系统(auth.go)和错误恢复机制(recover.go)
* **区块链集成**：开发区块链事件监听和索引服务，实现智能合约事件与后端数据库的同步
* **性能优化**：通过实现高效的缓存机制，显著提升API响应速度和系统吞吐量

**技术栈：**

* **开发语言**：Go
* **Web框架**：Gin
* **区块链交互**：以太坊/Web3
* **数据存储**：PostgreSQL
* **缓存系统**：Redis
* **配置管理**：TOML

**技术成就：**

* **API性能优化**：设计并实现了高效缓存中间件，将热门API端点响应时间减少70%，支持高并发NFT交易场景
* **区块链监听系统**：开发了可靠的链上事件监听服务，处理交易确认和撮合事件，确保后端数据与区块链状态一致性
* **批量处理优化**：实现了订单批量处理API，支持一次请求处理多个交易匹配，提高用户体验和系统效率
* **安全机制**：设计并实现了完整的API认证系统和请求频率限制，有效防止恶意攻击和系统滥用

**项目职责：**

* 负责EasySwap后端架构设计和核心组件开发
* 与前端和智能合约团队协作，确保系统各组件无缝集成
* 开发和维护区块链事件处理服务，确保数据同步和交易确认
* 设计并实现高效的API缓存系统和错误处理机制
* 编写技术文档和API说明，支持团队协作和系统维护

这段简历描述准确反映了作为Web3后端开发工程师在EasySwap项目