使用Golang设计数据和交易模型

在Hyperledger Fabric中，链码是由开发人员编写的智能合约的一种形式。链码实现了由区块链网络的利益相关者商定的业务逻辑。该功能暴露给客户端应用程序供其调用，只要它们具有正确的权限。

Chaincode在其自己的容器中作为独立进程运行，与Fabric网络的其他组件隔离。一个背书节点(endorsing peer)管理链码和事务调用的生命周期。通过响应客户调用，链码查询和更新账本并生成交易提议。

在本章中，我们将学习如何使用Go语言开发链码，并实施该场景下的智能合约业务逻辑。最后，我们将探讨开发全功能链码所需的关键概念和库。

在接下来的部分中，我们将探讨与概念相关的代码片段，您可以在以下地址完整实现链式代码：<https://github.com/HyperledgerHandsOn/trade-finance-logistics/tree/master/chaincode/src/github.com/trade_workflow_v1>

提示: 请注意，这也可以在我们上一章创建的本地git克隆中获得。我们有两个版本的链码，一个在trade\_workflow文件夹中，另一个在trade\_workflow\_v1文件夹中。 我们需要两个版本来演示第9章“区块链网络中的生活”中的升级。 在本章中，我们使用v1版本来演示如何在Go中编写链接代码。

在本章中，我们将涵盖以下内容：

* 创建chaincode
* 访问控制
* 实施chaincode功能
* 测试chaincode
* Chaincode设计主题
* 输出记录