

[前言](#)[Fabric 事件](#)[事件类型](#)[事件实现过程](#)[总结](#)[参考资料](#)

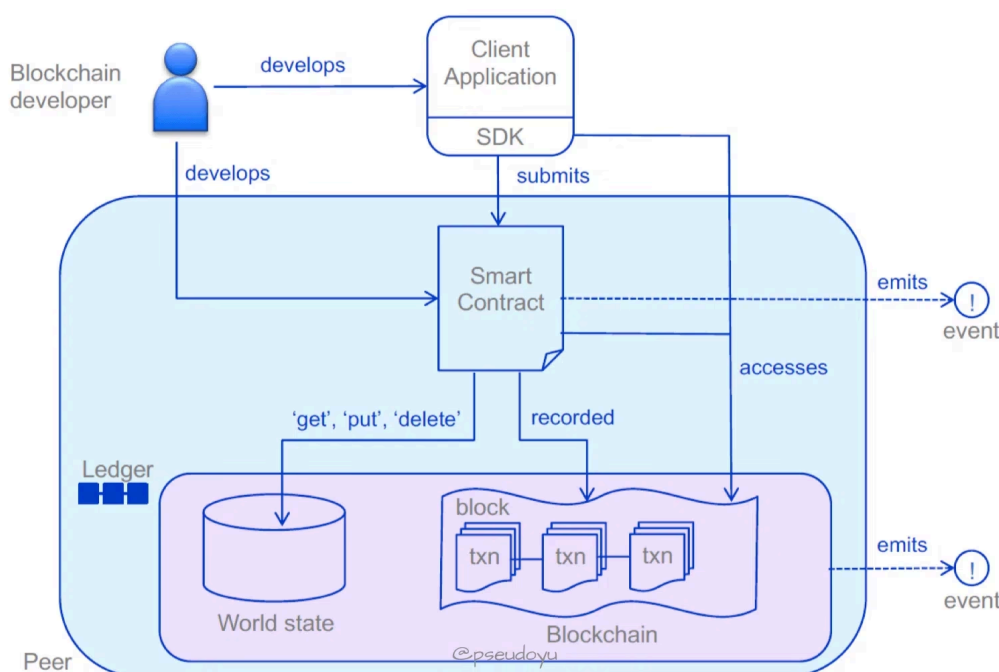
Hyperledger Fabric Go SDK 事件分析

前言

最近在做跨链适配器，需要在一条本地链上利用 Go SDK 来连接 fabric 网络，并监听事件，所以对 fabric 所支持的事件与 SDK 所提供的监听方法做一下汇总。

Fabric 事件

事件类型



事件是客户端与 Fabric 网络进行交互的一种方式，如上图所示，事件主要由 Ledger 和存有链码合约的容器触发。Fabric 共支持四种事件形式：

1. BlockEvent 监控新增到 fabric 上的块时使用
2. ChaincodeEvent 监控链码中发布的事件时使用，也就是用户自定义事件

3. TxStatusEvent 监控节点上的交易完成时使用
4. FilteredBlockEvent 监控简要的区块信息

在 Fabric Go SDK 中则通过以下几种事件监听器进行操作

1. `func (c *Client) RegisterBlockEvent(filter ...fab.BlockFilter) (fab.Registration, <-chan *fab.BlockEvent, error)`
2. `func (c *Client) RegisterChaincodeEvent(ccID, eventFilter string) (fab.Registration, <-chan *fab.CCEvent, error)`
3. `func (c *Client) RegisterFilteredBlockEvent() (fab.Registration, <-chan *fab.FilteredBlockEvent, error)`
4. `func (c *Client) RegisterTxStatusEvent(txID string) (fab.Registration, <-chan *fab.TxStatusEvent, error)`

而当监听完成后需要通过 `func (c *Client) Unregister(reg fab.Registration)` 来取消注册并移除事件通道

事件实现过程

实现时间过程需要两个步骤

1. 在链码中调用 `SetEvent` 方法
2. 在客户端中通过 Go SDK 实现事件监听器

SetEvent 方法

方法定义

```
func (s *ChaincodeStub) SetEvent(name string, payload []byte) error
```

调用实例

```
func (s *SmartContract) Invoke(stub shim.ChaincodeStubInterface) sc.Response {
    err = stub.PutState(key, value)

    if err != nil {
        return shim.Error(fmt.Sprintf("unable put state (%s), error: %v", key, err))
    }

    // Payload 需要转换为字节格式
    eventPayload := "Event Information"
    payloadAsBytes := []byte(eventPayload)
```

```
// SetEvent 方法通常位于 PutState、DelState 等与账本交互的操作之后
err = stub.SetEvent("<事件名称>", payloadAsBytes)

if (eventErr != nil) {
    return shim.Error(fmt.Sprintf("事件触发失败"))
}

return shim.Success(nil)
}
```

客户端事件监听器

```
// 实现一个链码事件监听
// 传入相应参数，这里的 eventId 必须与链码里的 <事件名称> 匹配以实现监听
reg, eventChannel, err := eventClient.RegisterChaincodeEvent(chaincodeID, eve

if err != nil {
    log.Fatalf("Failed to regitser block event: %v\n", err)
    return
}

// 取消注册并移除事件通道
defer eventClient.Unregister(reg)
```

总结

以上就是通过 Go SDK 对 fabric 网络上的事件进行监听操作的基本介绍，正在看 fabric Go SDK 源码，后续将补充一些解读。

参考资料

1. [hyperledger/fabric-sdk-go](#)
2. [Hyperledger Fabric Packages for Go Chaincode](#)
3. [fabric 支持的事件](#)
4. [Fabric 1.4 源码解读 3：事件\(Event\)原理解读](#)
5. [如何监听 Fabric 链码的事件](#)