1、开发包概述

Byfn.win 是 Hyperledger Fabric 著名的 byfn.sh 脚本的 Windows 版本的移植，用于帮助开发人员在 Windows 环境中快速搭建 Hyperledger Fabric 链码及应用开发环境。官网下载：Byfn.win。

Byfn.win 的主要特点如下：

- 使用原生构建的 windows 版本的 Fabric 程序，不需要安装虚拟机/Linux 子系统/Docker
- 一键复位 BYFN 网络，一键启动 BYFN 网络，为开发人员节省大量时间和精力
- 支持 TLS 安全传输设置，支持 solo 共识和 etcdraft 共识
- 支持 Hyperledger Fabric 官方及第三方提供的各种语言的链码与应用开发包
- 解压即用，绿色软件

Byfn.win 采用 Golang 开发，目前版本是 1.0.0，主要文件清单如下：

文件	说明
byfn.exe	byfn.win 主程序
first-network/	BYFN 网络配置目录
first-network/crypto-config.yaml	BYFN 密码学资料配置
first-network/configtx.yaml	BYFN 创世区块配置
first-network/ccp-template.json	连接配置文件 json 模板

文件	说明
first-network/ccp-template.yaml	连接配置文件 yaml 模板
first-network/orderer.yaml	Fabric 排序节点配置
first-network/core.yaml	Fabric 对等节点核心配置
first-network/e2e.cmd	端到端测试脚本
chaincode_example02/go/	Golang 链码，演示
chaincode_example02/node/	Node.js 链码，演示
chaincode_example02/java/	Java 链码，演示
fabric/bin/	Fabric 原生程序目录
byfn.win/	byfn.win 源代码目录
byfn.win/go.mod	go module 配置文件
byfn.win/main.go	入口代码
byfn.win/pkg/cmd/bygn.go	命令行处理代码
byfn.win/pkg/shell/shell.go	操作系统接口代码
byfn.win/pkg/shell/generate.go	BYFN 网络资料生成代码
byfn.win/pkg/shell/network.go	BYFN 网络运行与管理代码

2、Byfn.win 使用说明

2.1 生成 BYFN 网络基础资料

使用 byfn.exe 的 reset 子命令来生成或复位 BYFN 网络运行所依赖的 基础资料：

```
D:\byfn.win>byfn reset
############################################################
##### Generate certificates using cryptogen tool #########
############################################################
org1.example.com
org2.example.com
############################################################
#########  Generating Orderer Genesis block ##############
############################################################
□[34m2020-01-13 00:58:39.841 CST [common.tools.configtxgen] main -> INFO 001□[0m Loading configuration
□[34m2020-01-13 00:58:39.919 CST [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002□[0m orderer t
ype: solo
□[34m2020-01-13 00:58:39.919 CST [common.tools.configtxgen.localconfig] Load -> INFO 003□[0m Loaded configuration: D:\by
fn.win\first-network\configtx.yaml
□[34m2020-01-13 00:58:39.999 CST [common.tools.configtxgen.localconfig] completeInitialization -> INFO 004□[0m orderer t
ype: solo
□[34m2020-01-13 00:58:40.000 CST [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 005□[0m Loaded configuratio
n: D:\byfn.win\first-network\configtx.yaml
□[34m2020-01-13 00:58:40.004 CST [common.tools.configtxgen] doOutputBlock -> INFO 006□[0m Generating genesis block
□[34m2020-01-13 00:58:40.006 CST [common.tools.configtxgen] doOutputBlock -> INFO 007□[0m Writing genesis block
□[34m2020-01-13 00:58:40.058 CST [common.tools.configtxgen] main -> INFO 001□[0m Loading configuration
□[34m2020-01-13 00:58:40.137 CST [common.tools.configtxgen.localconfig] Load -> INFO 002□[0m Loaded configuration: D:\by
fn.win\first-network\configtx.yaml
□[34m2020-01-13 00:58:40.215 CST [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003□[0m orderer t
ype: solo
□[34m2020-01-13 00:58:40.216 CST [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 004□[0m Loaded configuratio
n: D:\byfn.win\first-network\configtx.yaml
□[34m2020-01-13 00:58:40.216 CST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 005□[0m Generating new chann
el configtx
```

注意：

1. 每次执行 reset 命令都会清空已有的区块链数据和密码学资料
2. 节点的输出日志在 first-network/logs 目录下

2.2 启动 BYFN 网络

使用 byfn.exe 的 up 子命令来启动 BYFN 网络：

```
D:\byfn.win>byfn up
Orderer Type:  solo
Bring Up All Peers:  false
Enable TLS:  false
Starting orderer: orderer.example.com ...
Starting peer: peer0.org1.example.com ...
Waiting for requests. Press Ctrl+C to quit.
```

up 子命令的选项如下：

- --tls：启用 tls，默认：false
- --full / -f：是否启动所有节点，默认：false，仅启动一个节点
- --orderer / -o：选择排序器实现，默认：solo，可选：solo 或 etcdraft。

默认情况下，byfn.win 禁用 TLS 并仅启动一个排序节点和一个对等节点，即：

- orderer.example.com
- peer0.org1.example.com

可以使用上述选项切换启动设置，例如启用 tls、etcdraft 排序并 启动所有 peer 节点：

```
D:\byfn.win>byfn up --tls -f -o etcdraft
Orderer Type:  etcdraft
Bring Up All Peers:  true
Enable TLS:  true
Starting orderer: orderer.example.com ...
Starting orderer: orderer2.example.com ...
Starting orderer: orderer3.example.com ...
Starting orderer: orderer4.example.com ...
Starting orderer: orderer5.example.com ...
Starting peer: peer0.org1.example.com ...
Starting peer: peer0.org2.example.com ...
Starting peer: peer1.org1.example.com ...
Starting peer: peer1.org2.example.com ...
Waiting for requests. Press Ctrl+C to quit.
```

2.3 进入 Peer 节点管理控制台

使用 `byfn.exe` 的 `admin` 子命令进入 peer 节点的管理控制台：

```
D:\byfn.win>byfn admin

BYFN-ADMIN@[peer0.org1.example.com]>
```

`admin` 子命令的选项如下：

- --peer / -p：设置节点编号，默认：0
- --org / -o：设置机构编号，默认：1

默认情况下进入 peer0.org1.example.com 的管理控制台，可以 使用上述选项进入不同的 peer 节点的控制台，例如进入 peer1.org2.example.com 的管理控制台：

```
byfn admin -p 1 -o 2
```

注意：

1. 当网络启用了 TLS 时，在进入管理终端时也需要启用 tls，例如：

```
byfn admin --tls
```

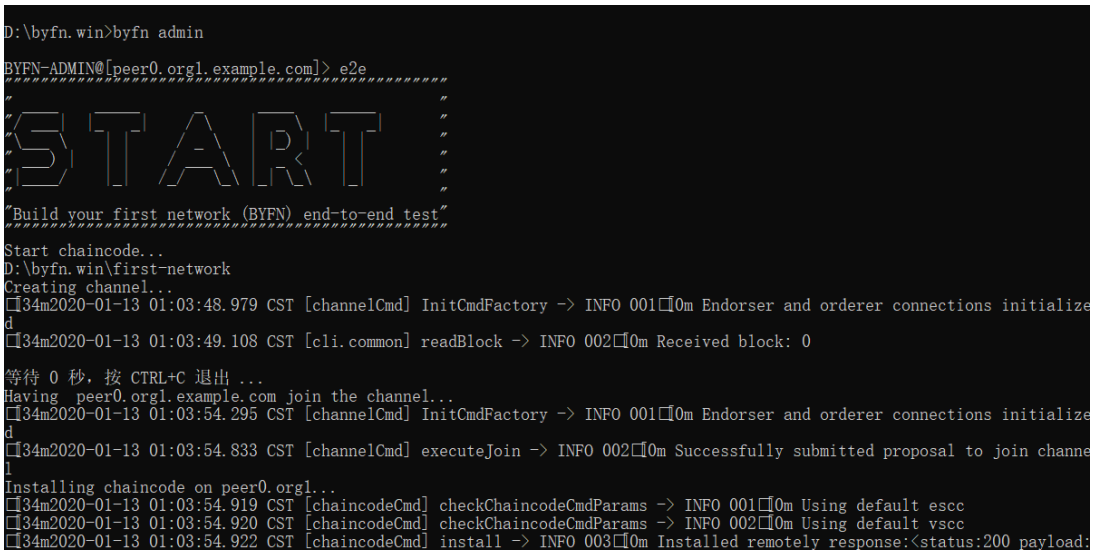
1. peer 命令需要额外的 tls 相关的参数，例如：

```
> peer channel list --tls --cafile=%ORDERER_CA%
```

其中环境变量 `ORDERER_CA` 中已经设置了相应的路径，可以直接使用。

2.4 执行端到端测试

进入管理控制台后，可以调用 `e2e.cmd` 来进行基本的测试：



e2e.cmd 主要执行如下任务：

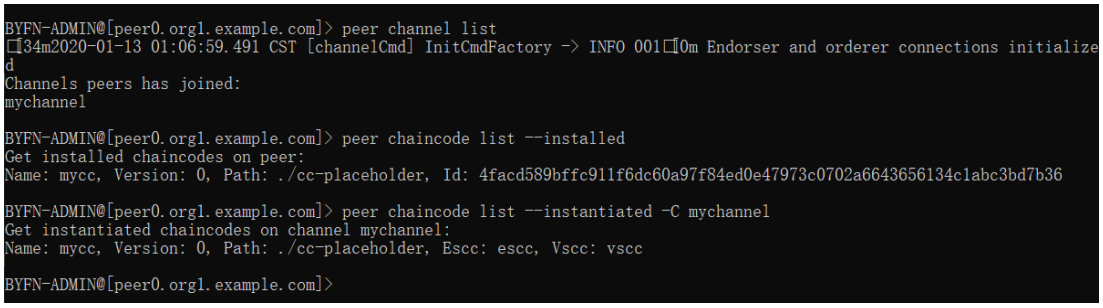
- 启动预置的链码 chaincode_example02
- 创建通道 mychannel
- 将 peer0.org1.example.com 加入 mychannel
- 在 peer0.org1.example.com 安装链码 mycc:0
- 在通道 mychannle 激活链码 mycc:0
- 查询链码 mycc:0 的状态

- 提交交易修改链码 mycc:0 的状态
- 再次查询链码 mycc:0 的状态
- 关闭链码 chaincode_example02

2.5 在管理控制台使用 fabric 预置命令

e2e.cmd 是一个标准的 windows 批处理文件，每一个命令都可以 在管理控制台单独执行。

例如，下面的三个命令分别用于查询当前所管理节点加入的通道、 当前节点安装的链码和指定通道激活的链码：



3、使用 byfn.win 测试自己开发的链码

首先使用 up 子命令启动网络：

```
byfn up
```

然后启动链码应用，例如启动预置的 nodejs 链码：

```
cd chaincode_example02/node

npm install

node index.js --peer.address=peer0.org1.examplecom:7052 --peer.id.name=myccjs:0
```

现在进入管理终端，就可以进行链码的安装、激活、查询或交易操作了。

安装链码：

```
> peer chaincode install -n myccjs -v 0 -l node -p ..\chaincode_example02\node
```

激活链码：

```
> peer chaincode instantiate -n myccjs -c "{\"Args\":[\"init\",\"tom\",\"1000\",\"mary\":\"2000\"]}" -C mychannel -o orderer.example.com
```

查询链码状态：

```
> peer chaincode query -n myccjs -c "{\"Args\":[\"invoke\",\"tom\"]}" -C mychannel
```

提交链码交易：

```
> peer chaincode invoke -n myccjs -c "{\"Args\":[\"invoke\",\"tom\",\"mary\",\"100\"]}" -C mychannel -o orderer.example.com
```

注意：

1. 在激活链码之前，需要先启动链码
2. 可以随时修改链码或重新运行链码，不需要重新激活

4、使用 byfn.win 开发应用

在执行 `reset` 子命令时，会自动生成 org1 的连接配置文件：

- connection-org1.json
- connection-org1.yaml

Hyperledger Fabric 官方提供的 SDK 可以直接使用上述连接配置文件， 可以根据自己的需要选择 json 或 yaml 格式。