

简介

- 因为无论配置交易文件.tx和初始区块文件block都是二进制格式，用户无法直接编辑。
- configtxlator工具将这些配置文件在二进制格式和方便阅读的json格式之间进行转换。
- 开发 configtxlator工具是为了支持独立于SDK来进行重新配置。
- configtxlator 工具被设计为提供一个API让任意一个SDK的用户都能够与之交互来更新配置。
- 工具的名称是 configtx和translator的拼接，意在传达该工具简单地在不同的等效数据之间进行转换。
- 它不产生配置。也不提交或撤回配置。它不修改配置本身，只是简单地提供一些配置格式的不同的展现。

编译

```
$ cd $GOPATH/src/github.com/hyperledger/fabric
$ make configtxlator
$ cp build/bin/configtxlator /usr/bin
```

参数说明

```
usage: configtxlator [<flags>] <command> [<args> ...]
```

Flags:

--help 显示上下文敏感的帮助信息 (also try **--help-long** and **--help-man**).

Commands:

help [<command>...]

显示帮助信息.

start [<flags>]

开启configtxlator REST服务端, 默认在本地7059端口启动监听

proto_encode **--type=TYPE** [<flags>]

转换JSON文件成指定的protobuf格式

proto_decode **--type=TYPE** [<flags>]

转换proto信息为JSON格式

compute_update **--channel_id=CHANNEL_ID** [<flags>]

带两个编码的common.Config消息, 并计算在两者之间转换的配置更新。

version

显示版本信息

restful api

- 解码: (接口地址为) /protolator/decode/{msgName}, 支持POST操作, 将二进制格式数据解码为Json格式, 其中{msgName}需要指定Fabric中定义的对应Protobuf消息结构的名称。
- 编码: (接口地址为) /protolator/encode/{msgName}, 支持POST操作, 将Json格式的数据编码为二进制格式, 其中{msgName}需要指定Protobuf消息类型名称
- 计算配置更新量: /configtxlator/compute/update-from-configs, 支持POST操作, 计算两个配置 (common.Config消息结构) 之间的二进制格式更新量。
-

编码

将本地使用configtxgen生成的Orderer系统通道初始区块，转换为Json格式。

```
configtxlator proto_encode --type common.Policy --output policy.pb  
curl -X POST --data-binary /dev/stdin  
"${CONFIGTXLATOR_URL}/protolator/encode/common.Policy" > policy.pb
```

解码

```
configtxlator proto_decode --input fabric_block.pb --type  
common.Block  
curl -X POST --data-binary @fabric_block.pb  
"${CONFIGTXLATOR_URL}/protolator/decode/common.Block"
```

更新

```
configtxlator compute_update --channel_id testchan --original  
original_config.pb --updated modified_config.pb | configtxlator  
proto_decode --type common.ConfigUpdate  
  
curl -X POST -F channel=testchan -F "original=@original_config.pb"  
-F "updated=@modified_config.pb"  
"${CONFIGTXLATOR_URL}/configtxlator/compute/update-from-configs" |  
curl -X POST --data-binary /dev/stdin  
"${CONFIGTXLATOR_URL}/protolator/encode/common.ConfigUpdate"
```