

# 应用程序开发配置和构建

可以使用 Go 开发应用程序，并利用 [Hyperledger Fabric SDK Go](#) 中的可用 API 来调用链代码，以在区块链网络中完成事务处理。现阶段我们先运行测试项目。

步骤：

- 1、示例应用服务端代码运行
- 2、下载证书
- 3、构建证书目录
- 4、Fabric SDK配置
- 5、应用程序开发
- 6、应用程序配置
- 7、应用程序构建

## 步骤一：示例应用服务端代码运行

0、root用户配置

1、Go环境安装

- 资源下载

官网地址：`https://golang.org/dl/`

下载路径：`https://dl.google.com/go/go1.10.2.linux-amd64.tar.gz`

解压到/usr/local/目录下：`tar -xvf /root/Downloads/go1.10.2.linux-amd64.tar.gz -C /usr/local/`

- 配置环境变量

打开配置文件：`gedit ~/.bashrc`

填写如下内容：

```
export GOPATH=$HOME/go
export GOROOT=/usr/local/go
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

永久生效：`source ~/.bashrc`

- 检查安装结果

```
go
```

2、docker环境安装

```
# step 1: 安装必要的一些系统工具
apt update
apt -y install apt-transport-https ca-certificates curl software-properties-common
# step 2: 安装GPG证书
curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
# Step 3: 写入软件源信息
add-apt-repository "deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu
$(lsb_release -cs) stable"
# Step 4: 更新并安装 Docker-CE
apt -y update
apt -y install docker-ce
# Step 5: 检查安装结果
docker version
# Step 6: 安装composer
curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -
s)-$(uname -m) -o /usr/local/bin/docker-compose
# Step 7: 修改权限
chmod +x /usr/local/bin/docker-compose
# Step 8: 检查安装结果
docker-compose --version
```

### 3、示例代码下载

服务端：[api-server.rar](#)

前端镜像：[portal.rar](#)

#### 注意

服务器文件解压完成后的目录结构。

### 4、Go开发工具安装

[https://www.jetbrains.com/zh/go/specials/go/go.html?utm\\_source=baidu&utm\\_medium=cpc&utm\\_campaign=cn-bai-pro-goland-ex-pc&utm\\_content=goland-ide&utm\\_term=go%20ide&gclid=CIGQn4jqn9sCFYFLvAod1kYFAw&gclsrc=ds&dclid=COHQr4jqn9sCFZswKgodokHMQ](https://www.jetbrains.com/zh/go/specials/go/go.html?utm_source=baidu&utm_medium=cpc&utm_campaign=cn-bai-pro-goland-ex-pc&utm_content=goland-ide&utm_term=go%20ide&gclid=CIGQn4jqn9sCFYFLvAod1kYFAw&gclsrc=ds&dclid=COHQr4jqn9sCFZswKgodokHMQ)

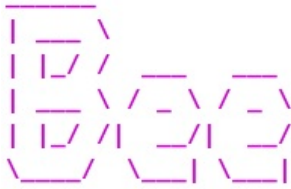
### 5、第三方包安装

```
go get -u github.com/kardianos/govendor
go get -u github.com/astaxie/beego
go get -u github.com/beego/bee
```

### 6、运行api-server

运行 `bee run api-server`

```
root@ubuntu:~/go/src/api-server# bee run api-server
```



```
2018/05/21 19:55:21 INFO    ▶ 0001 Using 'api-server' as 'appname'
2018/05/21 19:55:21 INFO    ▶ 0002 Initializing watcher...
2018/05/21 19:55:24 SUCCESS ▶ 0003 Built Successfully!
2018/05/21 19:55:24 INFO    ▶ 0004 Restarting 'api-server'...
2018/05/21 19:55:24 SUCCESS ▶ 0005 './api-server' is running...
2018/05/21 19:55:24 [I] [asm_amd64.s:2361] http server Running on http://0.0.0.0:8080
```

注意：

项目文件与build文件路径问题。我们使用Golang IDE进行项目的运行。

build文件夹放到\$GOPATH下，项目文件放到\$GOPATH/src目录下

## 步骤二：下载证书

应用程序开发人员需要到区块链服务管理页面下载对应服务的证书，如下图：

服务管理 ②

购买区块链服务

删除

所有状态 (1)

输入服务名称

Q

C

服务名称	服务状态	容器集群	共识策略	创建时间	操作
itcastqkl	运行	itcast-test	Kafka(CFT)	2018-05-15T11:49:47+08:00	更新版本 链代码管理 更多

组织名称	组织状态	组织类型	实例数量	操作
itcastqkl-orderer	运行	共识	2	下载管理员证书
DoHM	运行	节点	1	伸缩 下载管理员证书 下载用户证书
XXPD	运行	节点	1	伸缩 下载管理员证书 下载用户证书
PBoC	运行	节点	1	伸缩 下载管理员证书 下载用户证书

说明

目前支持两种证书：管理员证书和用户证书。创建通道、加入通道、更新通道、安装链代码、实例化链代码、升级链代码和删除链代码需要使用管理员证书，交易和查询推荐使用用户证书。

## 步骤三：构建证书目录

将证书文件压缩包下载到\$GOPATH/src/api-server/conf/crypto目录，并解压，完成后目录应类似如下结构：

```
total 64
drwxr-xr-x 4 paas users 4096 Apr 16 15:11 4fa097b85bc0d65271f9fe9dce98e6370e344c13.peer-4fa097b85bc0d65271f9fe9dce98e6370e344c13.default.svc.cluster.local
drwxr-xr-x 4 paas users 4096 Apr 16 15:12 62ffe38ae8e41ac33953b7c861309b7f4391c5d6.peer-62ffe38ae8e41ac33953b7c861309b7f4391c5d6.default.svc.cluster.local
drwxr-xr-x 4 paas users 4096 Apr 16 15:08 91201e2dc724320ab0d19e27f4ed5660e002f81d.orderer-91201e2dc724320ab0d19e27f4ed5660e002f81d.default.svc.cluster.local
drwxr-xr-x 4 paas users 4096 Apr 16 15:11 d4b95a85435a089cb70c1ec3c0de9a3f64c28aca.peer-d4b95a85435a089cb70c1ec3c0de9a3f64c28aca.default.svc.cluster.local
```

## 步骤四：Fabric SDK配置

1、单击“更多 > 下载SDK配置”，如下图：

删除

所有状态 (1)

输入服务名称

Q

↺

服务名称	服务状态	容器集群	共识策略	创建时间	操作
test	运行	test	快速拜占庭容错共识算法	2018-04-16T11:47:12+08:00	更新版本 链代码管理 更多

组织名称	组织状态	组织类型	实例数量	操作
test-orderer	运行	共识	4	下载管理员证书
xxx1	运行	节点	2	伸缩 下载管理员证书
xxx2	运行	节点	2	伸缩 下载管理员证书 下载用户证书
xxx3	运行	节点	2	伸缩 下载管理员证书 下载用户证书

下载SDK配置

重置链代码管理密码

服务信息

删除

## 2、配置SDK文件参数，如下图：

服务管理 ②

购买区块链服务

删除

所有状态 (1)

输入服务名称

Q

↺

服务名称	服务状态	容器集群	共识策略	创建时间	操作
test	运行	test	快速拜占庭容错共识算法	2018-04-16T11:47:12+08:00	更新版本 链代码管理 更多

组织名称	组织状态	组织类型	实例数量	操作
test-orderer	运行	共识	4	下载管理员证书
xxx1	运行	节点	2	伸缩 下载管理员证书
xxx2	运行	节点	2	伸缩 下载管理员证书 下载用户证书
xxx3	运行	节点	2	伸缩 下载管理员证书 下载用户证书

配置SDK文件 ②

×

链代码名称：

fabbank

链代码版本：

1.0

证书存放根路径：

/home/paas/code/baas-dcmc

通道名称：

testchannel

选择组织：

xxx1

选择节点：

peer-d4b95a85435a089cb70c1ec3c0d...

下载

取消

## 参数对照表

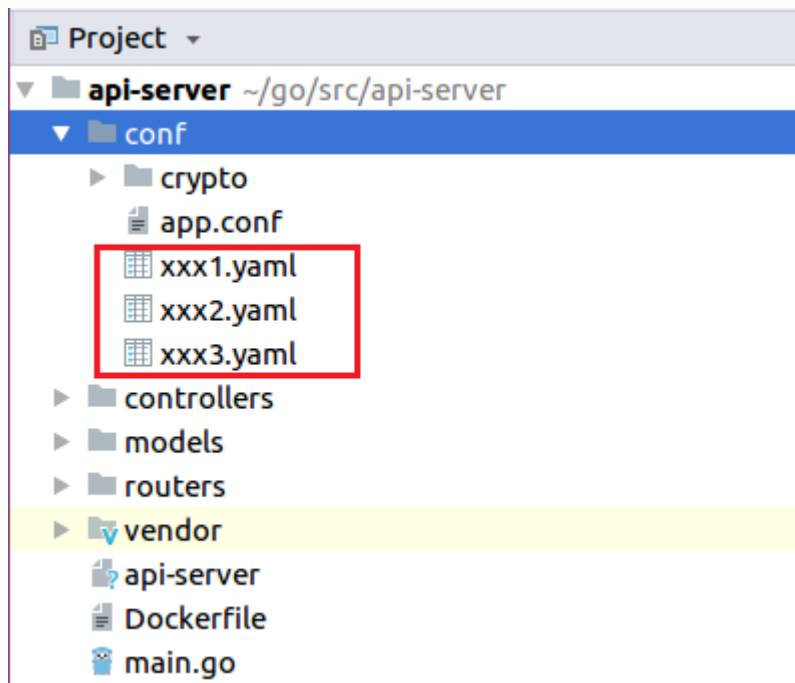
参数名称	参数值
链代码名称	fabbank
链代码版本	1.0
证书存放根路径	/opt/gopath/src/github.com/hyperledger/api-server/conf/crypto
通道名称	testchannel
选择组织	xxx1
选择节点	节点使用关联出来的默认配置

3、单击下载。下载后的文件命名为test-sdk-config.zip。

4、下载后解压出test-sdk-config.yaml文件，重命名为“组织名.yaml”，例如xxx1.yaml

5、组织xxx1的Fabric SDK下载完成后，需再用同样的方法，下载其他组织的SDK。下载后解压并重命名为“对应组织名.yaml”。

SDK最终目录结构如下：



## 步骤五：应用程序开发

用户开发人员可在自己的应用程序中通过 Fabric 的API来调用链代码，执行事务处理。可参考 `$GOPATH/src/api-server/controllers/transaction/transaction.go`。

示例代码：[transaction.go](#)

## 步骤六：应用程序配置

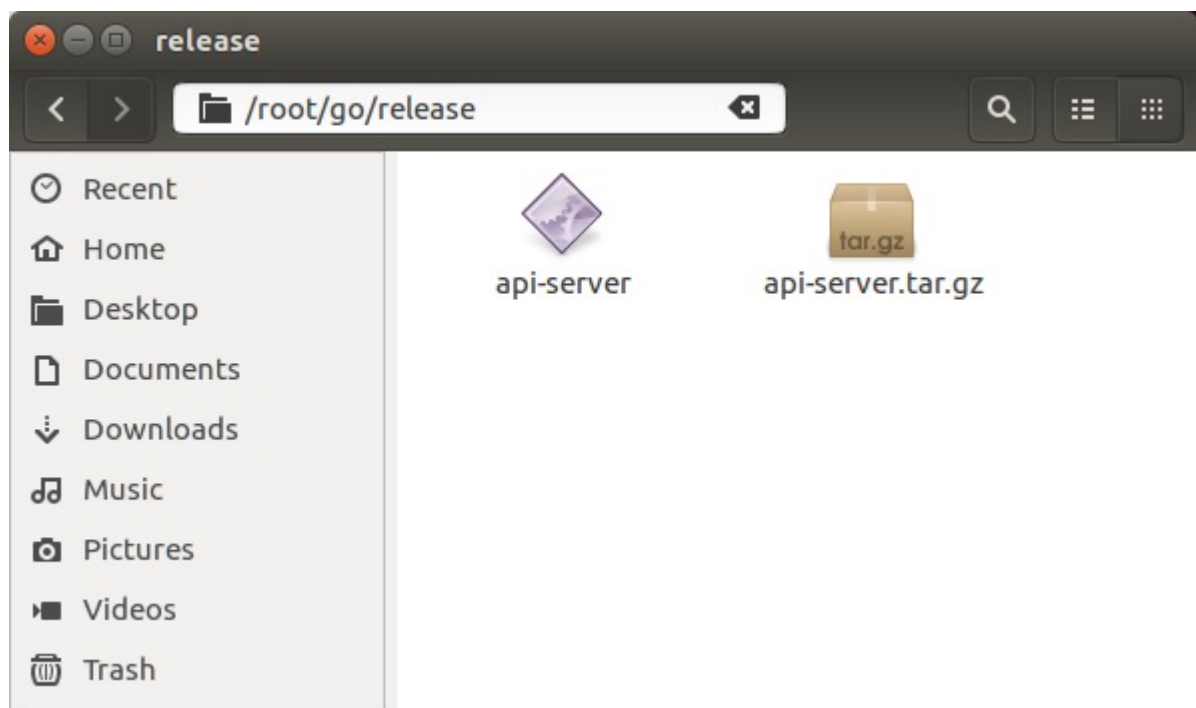
配置文件 `$GOPATH/src/api-server/conf/app.conf` 中链代码（参数为`chaincode_id`）及通道名称（参数为`channel_id`）的值，需与安装链代码时一致。

配置示例：[app.conf](#)

## 步骤七：应用服务端构建

服务端构建

进入 `$GOPATH/build` 目录执行命令：`bash -x build.sh`，生成应用程序服务端镜像 `api-server.tar.gz`。如下



注意：

需要把\$GOPATH/bin目录备份一下。