

App Example based on BaaS SDK for Java

简介

1. 本项目为基于Spring Boot的BaaS Java SDK应用示例工程。
2. 调用本项目提供的Restful APIs的示例，可参考app-web-example 目录下的基于AngularJS的web应用示例工程。

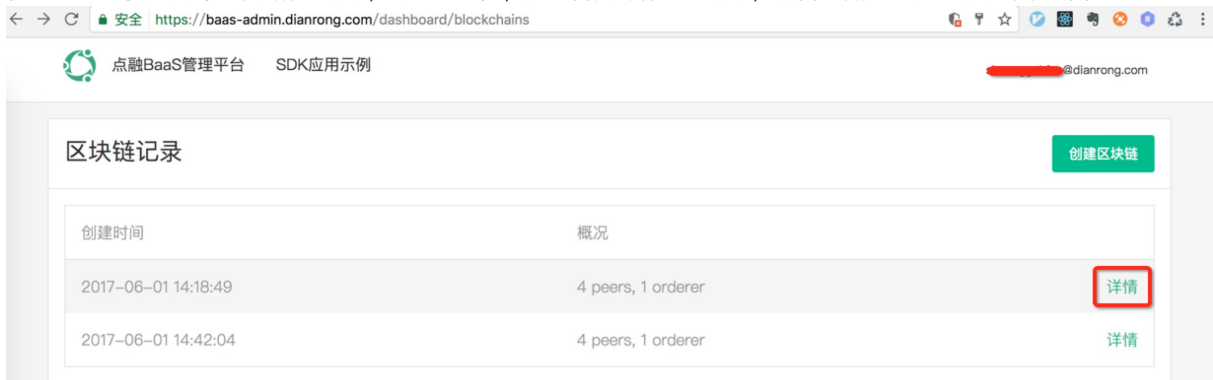
配置

证书

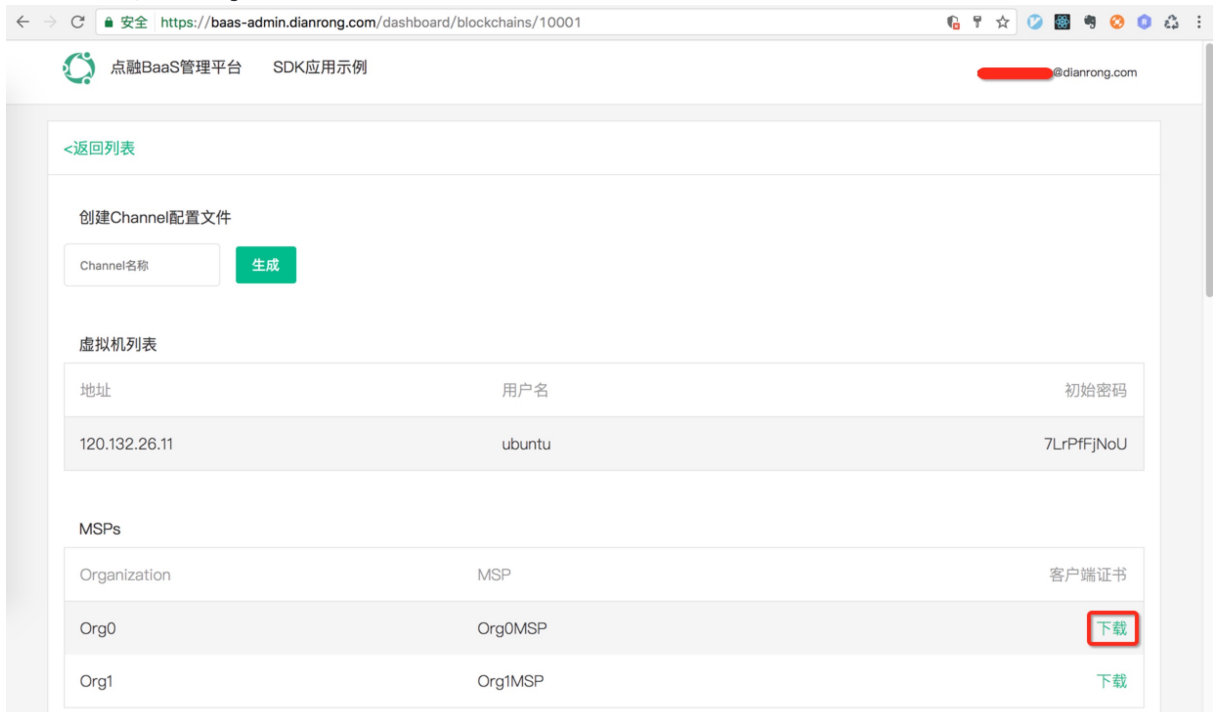
src/resources/msp目录下，包含了示例用户的证书(signcerts/cert.pem)和私钥(keystore/key.pem)，用于请求BaaS服务。开发应用时，需从[BaaS管理服务器](#)下载证书和私钥，替换示例证书和私钥。

下载方法：

1. 登陆[BaaS管理服务器](#)
2. 在区块链记录列表（如果没有区块链，可以创建），选择需要操作的区块链，点击“详情”打开区块链详情页面



3. 在MSPs列表，下载Org0MSP的客户端证书



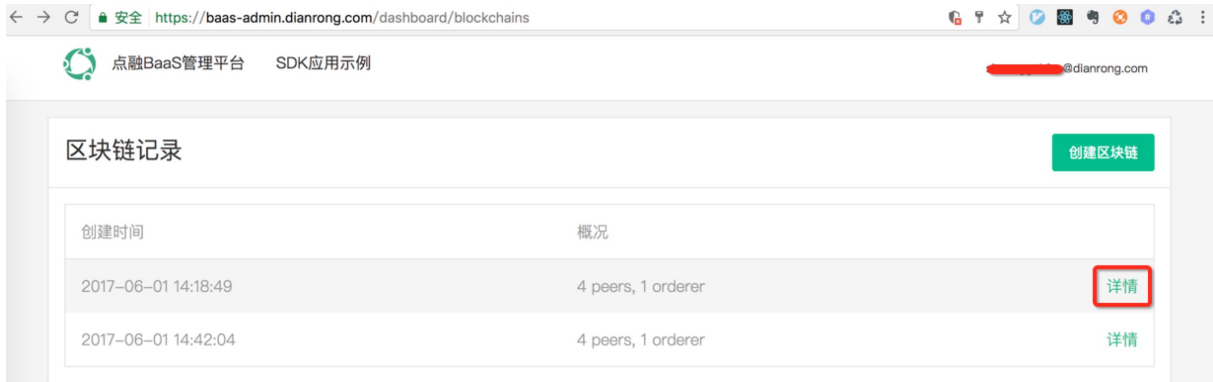
Channel tx文件

src/resources/channeltx目录下包含了创建Channel所需的示例tx文件。

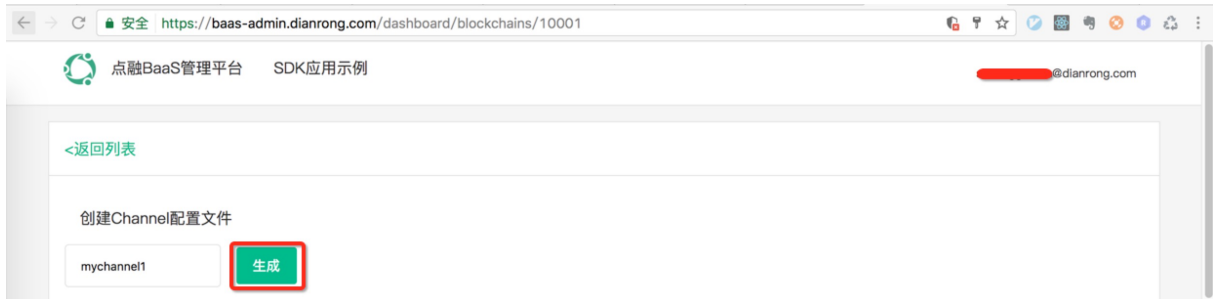
在创建新的Channel前，需预先在[BaaS管理服务器](#)上生成并下载以Channel名称命名的tx文件，复制到src/resources/channeltx目录下。

下载方法：

1. 登陆[BaaS管理服务器](#)
2. 在区块链记录列表（如果没有区块链，可以创建），选择需要操作的区块链，点击“详情”打开区块链详情页面



3. 填写Channel名称，然后点击“生成”



本项目的默认Channel名称为mychannel1，如果使用其它名称，需对下述3个文件进行修改

1. SDK客户端配置文件src/resources/appclient.yaml
修改“mychannel1”为自定义Channel名称
2. 应用配置文件src/main/resources/application.yml 修改app.channel.name为自定义Channel名称
3. app-web-example目录下的config.js 修改channel为自定义Channel名称 备注：本应用示例工程在启动后，会检查channel和chaincode的状态，如果未创建，将自动创建。
如果创建多个Channel，需修改应用配置文件src/main/resources/application.yml中的app.chaincode.name为不重复的名称。

Chaincode文件

src/resources/chaincode目录下包含了示例Chaincode文件。

设计实现Chaincode的细节，可参考[learn-chaincode](#)。

SDK客户端配置文件 (以下简称配置文件)

示例配置文件src/resources/appclient.yaml，包含对Chain和Channel的配置。

```
app:
  chains:
    mychain1:
      caserver: http://caserver:7054
      peers: peer0@grpc://peer0:7051, peer1@grpc://peer1:17051
      orderers: orderer0@grpc://orderer0:7050
      eventhubs: peer0@grpc://peer0:7053
      user:
        name: admin
        mspid: Org0MSP
        affiliation: Org0
        certpath: src/resources/msp/signcerts/cert.pem
        keystorepath: src/resources/msp/keystore/key.pem
  channels:
    mychannel1:
      peers: peer0, peer1
      orderers: orderer0
```

```
eventhubs: peer0
channeltx:
  path: src/resources/channeltx/
```

app.chains下包含了BaaS管理服务服务器上已创建的区块链，示例配置文件中仅包含一个区块链mychain1。目前区块链名称（mychain1）可由用户自定义，后续引用使用相同名称即可。
其它配置项具体使用细节可参考后续BaaS Java SDK使用方法部分的描述。

应用配置文件

示例应用配置文件src/main/resources/application.yml中，定义了应用所使用的channel，chaincode信息。

```
app:
  config:
    path: src/resources/appclient.yaml
  chain:
    name: mychain1
  channel:
    name: mychannel1
  chaincode:
    name: example_go
    version: 1
    path: github.com/example
    instantiation:
      fnc: Init
      args: testuser,1000,1
```

Channel的名称默认为mychannel1，如果生成Channel的tx文件时，使用了其它名称，需要修改为相同名称。应用在启动后，会检查channel和chaincode的状态，如果未创建，将自动创建。

使用

编译

在项目根目录下，执行

```
./gradlew build -x test
```

启动服务

```
./gradlew run
```

BaaS Java SDK

概述 BaaS Java SDK 部分常用类的使用方法

- AppClient
 - getChain
- Chain
 - createChannel
 - getChannel
 - installChaincode
 - instantiateChaincode
 - invokeChaincode
 - queryChaincode

getChain

```
Chain getChain(String chainName)
```

方法名：getChain

类型：实例方法

参数：String 类型的 Chain Name

返回: Chain

准备步骤:

1. 从BaaS管理服务器下载示例Org0的证书和私钥, 替换cert.pem及key.pem;
2. 根据BaaS管理服务器上显示的caserver, peers, orderers及eventhubs的urls, 修改/etc/hosts文件, 添加例如
120.132.25.168 peer0
120.132.25.168 peer1
120.132.25.168 orderer0
120.132.25.168 caserver
3. 确认配置文件 (例如src/resources/appclient.yaml) 存在如下Chain (例如mychain1) 的示例配置。

配置示例:

```
app:
  chains:
    mychain1:
      caserver: http://caserver:7054
      peers: peer0@grpc://peer0:7051, peer1@grpc://peer1:17051
      orderers: orderer0@grpc://orderer0:7050
      eventhubs: peer0@grpc://peer0:7053
      user:
        name: cli
        mspid: Org0MSP
        affiliation: Org0
        certpath: example/src/resources/msp/signcerts/cert.pem
        keystorepath: example/src/resources/msp/keystore/key.pem
```

使用示例:

```
AppClient client = new AppClient("src/resources/appclient.yaml");
String chainName = "mychain1";
Chain chain = client.getChain(chainName);
```

createChannel

```
Channel createChannel(String channelName)
```

方法名: createChannel

类型: 实例方法

参数: String 类型的 Channel Name

返回: Channel 准备步骤:

0. 完成getChain的准备步骤;
1. 在BaaS管理服务器上生成并下载以Channel名称命名的tx文件 (例如, mychannel.tx) , 复制到src/resources/channeltx目录下;
2. 确认配置文件中存在Channel (例如mychannel1) 如下述示例配置项。

配置示例:

```
...
channels:
  mychannel1:
    peers: peer0, peer1
    orderers: orderer0
    eventhubs: peer0
    channeltx:
      path: example/src/resources/channeltx/ //保存tx文件的目录
```

使用示例:

```
AppClient client = new AppClient("src/resources/appclient.yaml");
String chainName = "mychain1";
Chain chain = client.getChain(chainName);
String channelName = "mychannel1";
Channel channel = chain.createChannel(channelName);
```

getChannel

```
Channel getChannel(String channelName)
```

方法名: `getChannel`

类型：实例方法

参数: String 类型的 Channel Name

返回: Channel

准备步骤:

1. 确认配置文件中存在Channel（例如mychannel1）如下述示例配置项，配置需与Channel创建时保持一致。

配置示例：

```
...
channels:
  mychannel1:
    peers: peer0, peer1
    orderers: orderer0
    eventhubs: peer0
```

使用示例：

```
AppClient client = new AppClient("src/resources/appclient.yaml");
String chainName = "mychain1";
Chain chain = client.getChain(chainName);
String channelName = "mychannel1";
Channel channel = chain.getChannel(channelName);
```

installChaincode

```
ChaincodeTransactionResponse installChaincode(ChaincodeTransactionRequest chaincodeTransactionRequest)
```

方法名: `installChaincode`

类型：实例方法

参数：对象类型的 ChaincodeTransactionRequest

返回: ChaincodeTransactionResponse

准备步骤:

0. 完成getChain的准备步骤，如果没有创建过Channel，需完成createChannel的执行；
1. 设计实现Chaincode；
2. 确认配置文件中存在如下示例配置项；

配置示例：

```
...
chaincode:
  path: src/resources/chaincode/
  endorsementpolicy: src/resources/policy/endorsementpolicy.yaml
```

备注：Chaincode和endorsementpolicy的细节可参考[learn-chaincode](#)。使用示例：

[illegible]

```
chaincodePath,  
null,  
null);  
ChaincodeTransactionResponse installResponse = chain.installChaincode(installRequest);
```

instantiateChaincode

```
ChaincodeTransactionResponse instantiateChaincode(ChaincodeTransactionRequest chaincodeTransactionRequest)
```

方法名: instantiateChaincode

类型: 实例方法

参数: 对象类型的 ChaincodeTransactionRequest

返回: ChaincodeTransactionResponse

示例:

```
AppClient client = new AppClient("src/resources/appclient.yaml");  
String chainName = "mychain1";  
Chain chain = client.getChain(chainName);  
String channelName = "mychannel1";  
Channel channel = chain.getChannel(channelName);  
String chaincodeName = "example_cc_go";  
String chaincodeVersion = "1";  
String chaincodePath = "github.com/example";  
String fnc = "init";  
String[] args = new String[]{"testuser", "1000", "1"};  
ChaincodeTransactionRequest instantiateRequest = new ChaincodeTransactionRequest(channelName,  
chaincodeName,  
chaincodeVersion,  
chaincodePath,  
fnc,  
args);  
ChaincodeTransactionResponse instantiateResponse = chain.instantiateChaincode(instantiateRequest);
```

invokeChaincode

```
ChaincodeTransactionResponse invokeChaincode(ChaincodeTransactionRequest chaincodeTransactionRequest)
```

方法名: invokeChaincode

类型: 实例方法

参数: 对象类型的 ChaincodeTransactionRequest

返回: ChaincodeTransactionResponse

示例:

```
AppClient client = new AppClient("src/resources/appclient.yaml");  
String chainName = "mychain1";  
Chain chain = client.getChain(chainName);  
String channelName = "mychannel1";  
Channel channel = chain.getChannel(channelName);  
String chaincodeName = "example_cc_go";  
String chaincodeVersion = "1";  
String chaincodePath = "github.com/example";  
String fnc = "invoke";  
String[] args = new String[]{"start", "testuser", "1495786988"};  
ChaincodeTransactionRequest invokeRequest = new ChaincodeTransactionRequest(channelName,  
chaincodeName,  
chaincodeVersion,  
chaincodePath,  
fnc,  
args);  
ChaincodeTransactionResponse invokeResponse = chain.instantiateChaincode(invokeRequest);
```

queryChaincode

```
ChaincodeTransactionResponse queryChaincode(ChaincodeTransactionRequest chaincodeTransactionRequest)
```

方法名: queryChaincode

类型: 实例方法

参数: 对象类型的 ChaincodeTransactionRequest

返回: ChaincodeTransactionResponse

示例:

```
AppClient client = new AppClient("src/resources/appclient.yaml");
String chainName = "mychain1";
Chain chain = client.getChain(chainName);
String channelName = "mychannel1";
Channel channel = chain.getChannel(channelName);
String chaincodeName = "example_cc_go";
String chaincodeVersion = "1";
String chaincodePath = "github.com/example";
String fnc = "invoke";
String[] args = new String[]{"query", "testuser"};
ChaincodeTransactionRequest queryRequest = new ChaincodeTransactionRequest(channelName,
                                                                              chaincodeName,
                                                                              chaincodeVersion,
                                                                              chaincodePath,
                                                                              fnc,
                                                                              args);

ChaincodeTransactionResponse queryResponse = chain.queryChaincode(queryRequest);
```

Restful APIs

Chaincode的使用分为四个步骤。其中, Install/Instantiate只需执行一次, Invoke/Query可以多次执行。

- Install
- Instantiate
- Invoke
- Query

app-example提供了Restful APIs, 分别为

- installChaincode(@RequestBody ChaincodeTransactionRequest tx)
- instantiateChaincode(@RequestBody ChaincodeTransactionRequest tx)
- invokeChaincode(@RequestBody ChaincodeTransactionRequest tx)
- queryChaincode(@RequestBody ChaincodeTransactionRequest tx)

示例chaincode "rentalexample" 在src/resources/chaincode/rentalexample/src/github.com/example目录下。
具体调用invokeChaincode和queryChaincode的APIs可参考app-web-example下的实现。