

区块链 SDK 接入文档

Table of Contents

引入Fabric开发包

Maven

Gradle

配置文件

代码示例

引入Fabric开发包

Maven

```
<dependency>
  <groupId>org.hyperledger.fabric</groupId>
  <artifactId>fabric-gateway-java</artifactId>
  <version>1.4.2</version>
</dependency>
```

Gradle

```
compile group: 'org.hyperledger.fabric', name: 'fabric-gateway-java', version: '1.4.2'
```

配置文件



配置文件需复制到项目中，代码中需要使用，文件格式：.yaml



配置文件中，组织名称、成员ID、节点列表、访问地址、证书等由区块链平台提供

参考内容如下 connection-orgsi.yaml

```
# 名称，自定义
name: ncp-contract-orgsi
# 版本，自定义
version: 1.0.0
# 客户端
client:
  # 组织（区块链提供）
  organization: orgsi
organizations:
  # 与client.organization 匹配
  orgsi:
    # 成员ID（区块链提供）
    mspid: orgsiMSP
    # 节点列表（区块链提供）
    peers:
      - peer0.orgsi-com
      - peer1.orgsi-com
# 节点配置，配置的所有节点与organizations.orgsi.peers列表匹配
peers:
```

```
peer0.orgsi-com:
# 节点访问地址（区块链提供）
url: grpcs://peer0.orgsi-com:7051
# 证书文件（区块链提供）
tlsCACerts:
  path: F:/项目文档/南宁/电子合同/FabricSDK/正式/tlsca.orgsi-com-cert.pem
# Grpc选项
grpcOptions:
  ssl-target-name-override: peer0.orgsi-com
  hostnameOverride: peer0.orgsi-com
peer1.orgsi-com:
url: grpcs://peer1.orgsi-com:7051
tlsCACerts:
  path: F:/项目文档/南宁/电子合同/FabricSDK/正式/tlsca.orgsi-com-cert.pem
grpcOptions:
  ssl-target-name-override: peer1.orgsi-com
  hostnameOverride: peer1.orgsi-com
```

代码示例



配置文件参考上方示例，配置在代码工程中；其他参数（证书文件、私钥文件、通道名称、合约名称、调用方法等）均由区块链平台提供；节点部署成功后联系相关人员获取Host配置，配置Host后方可与区块链网络建立连接

```
Path walletPath = Paths.get("wallet");
Wallet wallet = Wallet.createFileSystemWallet(walletPath);
// 私钥文件为: xxx-Admin.tar\admin\msp\keystore\*_sk; 证书文件为: xxx-Admin.tar\admin\msp\signcerts\Admin@xxx.xxx-com-
cert.pem; (xxx-Admin.tar 由区块链提供)
Wallet.Identity user = Wallet.Identity.createIdentity(mspId, new InputStreamReader(new FileInputStream(证书路径)), new
InputStreamReader(new FileInputStream(私钥文件路径)));
wallet.put("admin", user);
ClassPathResource classPathResource = new ClassPathResource(配置文件路径);
Gateway.Builder builder = Gateway.createBuilder();
builder.identity(wallet, "admin").networkConfig(classPathResource.getInputStream()).discovery(true);
Gateway gateway = builder.connect();
network = gateway.getNetwork(通道名称);
Contract contract = network.getContract(合约名称);
// 保存数据使用submitTransaction
contract.submitTransaction(调用方法名称, Json对象/Json数组);
// 查询数据使用evaluateTransaction, 返回数据格式为Json
byte[] result = contract.evaluateTransaction(调用方法名称, 调用参数/查询条件);
```