

区块链API升级使用说明

因趣链平台Java SDK升级，近期要对区块链门户网站上已有的需要与区联平台对接的功能进行升级。本次已完成数据合约的上链与查询功能的改造，并在测试环境通过了对接的联调测试。

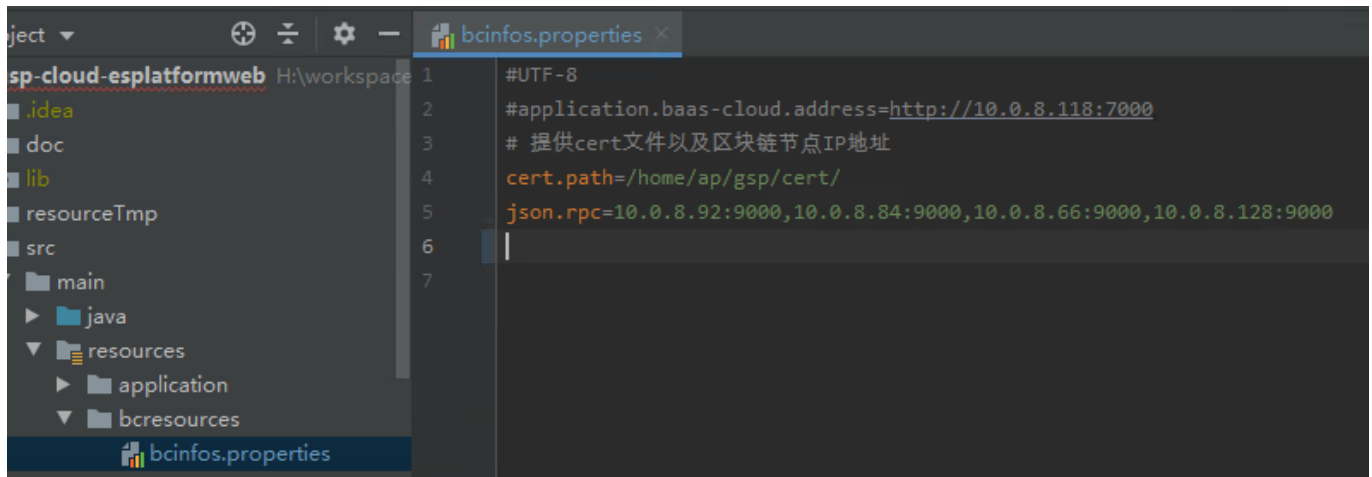
开发使用步骤

1. 环境准备

1.1 Cert文件与区块链节点地址

区块链数据接入层接口调用趣链Javasdk访问合约，趣链jvasdk直接和节点通信，因此需要由应用程序提供cert文件以及区块链节点IP地址。由趣链平台提供相应环境的Cert文件与区块链节点信息。

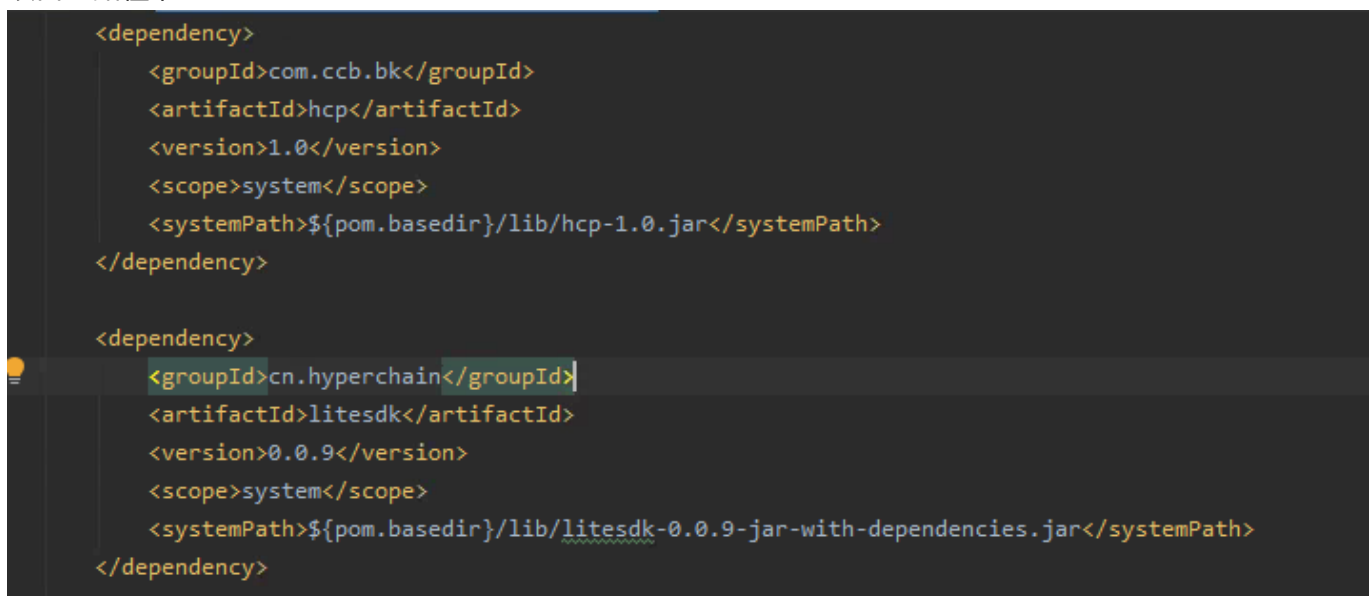
添加Cert文件与节点信息的配置：



```
1 #UTF-8
2 #application.baas-cloud.address=http://10.0.8.118:7000
3 # 提供cert文件以及区块链节点IP地址
4 cert.path=/home/ap/gsp/cert/
5 json.rpc=10.0.8.92:9000,10.0.8.84:9000,10.0.8.66:9000,10.0.8.128:9000
6
7
```

1.2 添加jar包

项目lib路径下



```
<dependency>
  <groupId>com.ccb.bk</groupId>
  <artifactId>hcp</artifactId>
  <version>1.0</version>
  <scope>system</scope>
  <systemPath>${pom.basedir}/lib/hcp-1.0.jar</systemPath>
</dependency>

<dependency>
  <groupId>cn.hyperchain</groupId>
  <artifactId>litesdk</artifactId>
  <version>0.0.9</version>
  <scope>system</scope>
  <systemPath>${pom.basedir}/lib/litesdk-0.0.9-jar-with-dependencies.jar</systemPath>
</dependency>
```

2. 代码开发

2.1 ManageServiceImpl初始化

初始化ManageServiceImpl实例，必须传入正确的certpath和jsonrpc，如果路径不对或者证书认证不通过以及节点连接不上会返回null。

```
CatalogDaoImpl.java x
@Autowired
ManageServiceImpl manageService;

@Bean
public ManageServiceImpl manageServiceBean() {
    String certPath = CommonUtils.getProperties( relativePath: "bcresources/bcinfos.properties", propName: "cert.path");
    String jsonRpc = CommonUtils.getProperties( relativePath: "bcresources/bcinfos.properties", propName: "json.rpc");
    LOG.info( s: "get the configuration, certPath: {}, jsonRpc: {}", certPath, jsonRpc);
    ManageServiceImpl instance = ManageServiceImpl.getInstance(certPath, jsonRpc);
    if (instance == null) {
        LOG.info( s: "===== call getInstance of ManageServiceImpl, but return null instance =====");
    }
    return instance;
}
```

调用接口之前，做实例为空的判断

```
CatalogDaoImpl.java x
310
311 @ private JSONObject handleWhenEmptyInstance() {
312     if (manageService == null) {
313         LOG.info( s: "error! the instance of manageService is null");
314         Map<String, String> map = new HashMap<>();
315         map.put("result", "get the instance of manage service is null");
316         return ContrUtils.ToJson( success: "false", (new Gson()).toJson(map));
317     }
318     return null;
319 }
```

2.2 业务接口使用

2.2.1 数据上链

保存数据到链上，传入数据合约地址、资源归属方签名文件内容、资源标识、数据集Map。

```
CatalogDaoImpl.java x
98 @Override
99 public JSONObject addData(String resourceID, Map<String, String> dataMap, String contractAddr, String txOriJson) {
100     JSONObject jsonObject = handleWhenEmptyInstance();
101     if (jsonObject != null) return jsonObject;
102     LOG.info( s: " ----- call manageService.addData, contractAddr: {}, txOriJson: {} -----" + txOriJson.toJson());
103     String callMethod = CommonUtils.getProperties( relativePath: "bcresources/bcinfos.properties", propName: "call.method");
104     return manageService.updtData(contractAddr, txOriJson, resourceID, dataMap);
}
```

说明：目录合约要先上链，数据合约才可以使用对应资源的resourceID进行上链；参与方必须要有该资源的读写权限；调用方需要注意区分权限合约地址和数据合约地址；集合Map的key为数据记录的业务主键值，value为整条记录；单次上链的数据量不要超过500条；

2.2.2 数据查询

查询链上数据, 传入数据合约地址、资源归属方签名文件内容、资源标识、数据id列表。

```
CatalogDaoImpl.java
280 @Override
281 public JSONObject getData(ExTaskDo extask, List<String> resIdList, ContractEntity contract) throws IOException {
282     /*...*/
286
287     String resItemId = extask.getResourceID();
288     LOG.info( s: "Blockchain call getData Contract start, itemId: {}", data pkValues: {}, extask.getId(), resItemId, resIdList);
289     /*...*/
291     JSONObject jsonObject = handleWhenEmptyInstance();
292     if (jsonObject != null) return jsonObject;
293     JSONObject result = manageService.getData(contract.getContractAddr(), contract.getTxOriJson(), resItemId, resIdList);
294     LOG.info( s: "Blockchain call getData Contract end, itemId: {}, result: {}", extask.getId(), resItemId, result);
295     return result;
296 }
```