

数据共享平台 SDK 接口设计

(JAVA 版)

版本：V 1.0.0

2017 年 11 月

文档修订记录

| 版本 编号 | *变化 状态 | 变更内容和范围 | 变更日期 | 变更人 | 批准 日期 | 批准 人 |
|----------|-----------|-------------------|------------|-----|----------|---------|
| 1.0.0 | A | 创建数据共享平台 SDK 接口设计 | 2017/11/28 | 宋嘉 | | |
| | | | | | | |

*变化状态：A——增加，M——修改，D——删除，N——正式发布

目录

| | |
|------------------------------|----|
| 数据共享平台 SDK 接口设计 | 1 |
| 1 数据共享平台 SDK 相关说明 | 4 |
| 1.1 SDK 引用 | 4 |
| 1.2 JDK 要求 | 4 |
| 1.3 配置文件说明 | 5 |
| 2 接口设计 | 5 |
| 2.1 实体定义 | 5 |
| 2.1.1 StoreDataParams | 5 |
| 2.1.2 GetDataParams | 6 |
| 2.1.3 VerifyDataParams | 6 |
| 2.1.4 GetDataVO | 6 |
| 2.1.5 VerifyDataVO | 7 |
| 2.2 接口定义 | 7 |
| 2.2.1 配置文件路径初始化接口 | 7 |
| 2.2.2 数据上链存储接口 | 8 |
| 2.2.3 上链数据查询接口 | 9 |
| 2.2.4 上链数据验证接口 | 10 |

1 数据共享平台 SDK 相关说明

1.1 sdk 引用

将附件“open-datashare-sdk-1.0.0-RELEASE-with-dependencies.jar”包导入应用程序工程内，并确保工程中已导入 sdk 相关依赖 jar，依赖列表如下：

| 依赖包名称 | 版本 |
|---------------|--------|
| commons-lang3 | 3.5 |
| fastjson | 1.2.28 |
| okhttp | 3.4.1 |
| slf4j-api | 1.7.25 |

Maven 方式引用配置：

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.25</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.7.25</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.5</version>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.28</version>
</dependency>
<dependency>
  <groupId>com.squareup.okhttp3</groupId>
  <artifactId>okhttp</artifactId>
  <version>3.4.1</version>
</dependency>
```

1.2 jdk 要求

open-datashare-sdk-1.0.0-RELEASE-with-dependencies.jar
是基于 jdk1.7 开发与编译，故引用该包的工程环境必须为 jdk1.7+。

1.3 配置文件说明

➤ sdk 需要使用到两个配置文件：

1、通用参数配置文件：imi-config.properties

| 参数名称 | 描述 | 示例 |
|---------------------------|----------------|-----------------------|
| 通信地址 | | |
| url.share.addData.prepare | 数据上链参数与处理接口地址 | 由我司提供 |
| url.share.addData.commit | 数据上链交易提交接口地址 | |
| url.share.getData | 上链数据查询接口地址 | |
| url.share.verifyData | 上链数据验证接口地址 | |
| 基本信息 | | |
| imi.vportId | 数字身份号 | 现暂由我司分配，后期通过网络申请 |
| imi.name | 数字身份名称 | 由用户自定义（如：第三方系统名称） |
| imi.ks.pass | 私钥 KeyStore 密码 | 暂由我司提供，后期通过网络申请，用户自定义 |

2、区块链私钥存储 KeyStore 文件：imi-ks

KeyStore 文件暂由我司提供，后期通过网络申请；

➤ 配置文件存储位置：

1、应用程序内部存储：

存储于应用程序 classpath 根目录下，文件路径为

/imi/imi-config.properties

/imi/imi-ks

2、应用程序外部存储

存储于系统文件目录下，如 linux 下的存储位置如下：

/home/imi/imi-config.properties

/home/imi/imi-ks

在应用程序启动是通过调用 sdk 的参数初始化配置接口
“IMConfiguration.initConfigPath(String imiConfigPath, String imiKsPath)”传入两个配置文件路径。

2 接口设计

接口相关内容可参考 API 文档包 open-datashare-sdk-doc-1.0.0

2.1 实体定义

2.1.1 StoreDataParams

| 名称 | 上链数据存储接口参数对象 | StoreDataParams |
|----|--------------|-----------------|
|----|--------------|-----------------|

| 字段名 | 类型 (是否必填) | 说明 | 备注 |
|---------------|--------------|----------------|---------------------|
| data | String(Y) | 需要上链的用户原数据 | SDK 中会对原数据进行 jwt 封装 |
| typeCode | String(Y) | 上链数据类型编号 | JSON 模版编号 |
| subjectCid | String(Y) | 上链数据所属用户的身份证号 | |
| subjectName | String(Y) | 上链数据所属用户的身份证姓名 | |
| issuerVportId | String(Y) | 上链数据发布者数字身份编号 | 发布者即第三方业务系统 |
| isAppend | int(N) | 上链数据方式 | 添加：0，更新：1，默认是 0 |

2.1.2 GetDataParams

| 名称 | 上链数据查询接口参数对象 | | GetDataParams |
|-----------------|--------------|------------------|-------------------------------|
| 字段名 | 类型 (是否必填) | 说明 | 备注 |
| typeCode | String(Y) | 上链数据类型编号 | JSON 模版编号 |
| subjectCid | String(Y) | 上链数据所属用户的身份证号 | |
| subjectName | String(Y) | 上链数据所属用户的身份证姓名 | |
| issuerVportId | String(Y) | 上链数据发布者数字身份编号 | 发布者即第三方业务系统 |
| endTime | String(N) | 按上链时间查询：截止时间 | 默认当前时间，格式：yyyy-MM-dd hh:mm:ss |
| num | Integer(N) | 查询数据条数 | 默认值：10 |
| transactionHash | String(N) | 交易编号可以查询到唯一的上链数据 | 该条件与上述条件互斥，该条件有效时，其他查询条件失效 |

2.1.3 VerifyDataParams

| 名称 | 上链数据验证接口参数对象 | | VerifyDataParams |
|-----------------|--------------|------------------|------------------|
| 字段名 | 类型 (是否必填) | 说明 | 备注 |
| transactionHash | String(N) | 交易编号可以查询到唯一的上链数据 | |

2.1.4 GetDataVO

| 名称 | 上链数据查询接口响应结果对象 | | GetDataVO |
|----|----------------|--|-----------|
|----|----------------|--|-----------|

| 字段名 | 类型 (是否必填) | 说明 | 备注 |
|-----------------|--------------|--------------|----------------------------|
| data | String(Y) | 已上链的用户数据 | SDK 中会对接口返回的 jwt 数据进行解封 |
| typeCode | String(Y) | 上链数据类型编号 | JSON 模版编号 |
| transactionHash | String(Y) | 上链数据的区块链交易编号 | |
| createTime | String(Y) | 数据上链时间 | 格式: yyyy-MM-dd hh:mm:ss |

2.1.5 VerifyDataVO

| 名称 | 上链数据验证接口响应结果对象 | | VerifyDataVO |
|-----------------|----------------|--------------|-------------------------|
| 字段名 | 类型 (是否必填) | 说明 | 备注 |
| data | String(Y) | 已上链的用户数据 | SDK 中会对接口返回的 jwt 数据进行解封 |
| typeCode | String(Y) | 上链数据类型编号 | JSON 模版编号 |
| transactionHash | String(Y) | 上链数据的区块链交易编号 | |

2.2 接口定义

2.2.1 配置文件路径初始化接口

| | | | | | |
|------------------|--|------------------|-------------|--|--|
| 接口名称 | 配置文件路径初始化接口 | | | | |
| 类名 | 方法名 | | | | |
| IMIConfiguration | initConfigPath(String imiConfigPath, String imiKsPath) | | | | |
| 功能 | 此接口调用需要放在应用程序启动时调用 | | | | |
| 输入参数 | | | 输出参数 result | | |
| imiConfigPath | String(Y) | Sdk 通用参数配置文件路径 | | | |
| imiKsPath | String(Y) | 私钥 KeyStore 文件路径 | | | |

示例:

| |
|---|
| <pre>// 配置文件系统存储绝对路径 String imiConfigPath = "/home/imi/imi-config.properties"; String imiKsPath = "/home/imi/imi-ks"; try { IMIConfiguration.initConfigPath(imiConfigPath, imiKsPath); } catch (FileNotFoundException e) { e.printStackTrace(); }</pre> |
|---|

```
}
```

2.2.2 数据上链存储接口

| | | | | | |
|-------------------|--|--------------|---------------------|--------|------------------|
| 接口名称 | 数据上链存储接口 | | | | |
| 类名 | 方法名 | | | | |
| DataSharingRouter | storeData(StoreDataParams params) | | | | |
| 功能 | 此接口内封装了数据共享平台数据上链的两个 Http 接口调用，以及交易数据本地签名操作 | | | | |
| 输入参数 | | | 输出参数 result | | |
| params | StoreDataParam s (Y) | 上链数据存 储参数 | transaction Hash | String | 数据上链交易 Hash 值 |

示例：

```
// 上链原始数据
String rawData = "{\\"idCard\\":\\"43072119801024008x\\",\\"gender\\":\\"男\\",\\"idCardStartTime\\":\\"2015-02-15\\",\\"idCardEndTime\\":\\"2035-02-15\\",\\"idCardName\\":\\"向南\\",\\"idCardAddr\\":\\"北京朝阳酒仙桥\\",\\"idCardOrganization\\":\\"北京市公安局\\",\\"idCardNation\\":\\"汉族\\",\\"idCardHeadImg\\":\\"...\\"}";

// 数据模版类型编号
String typeCode = "10001";

// 上链数据所属用户身份证号
String subjectCid = "43072119801024008x";

// 上链数据所属用户身份证姓名
String subjectName = "向南";

// 第三方业务系统自行实现分布式锁
Lease lock = null;
try {
    StoreDataParams params = new StoreDataParams();
    params.setData(rawData);
    params.setTypeCode(handler.typeCode);
    params.setSubjectCid(subjectCid);
    params.setSubjectName(subjectName);
    params.setIssuerVportId(CommonConstants.imiVportId); // 第三方业务系统数字身份
    params.setIsAppend(0); // 0:添加数据, 1:更新数据
```



```
// 加锁
lock = Lock.getLock(handler.LOCK_DATASHARE_STOREDATA);

String transactionHash = DataSharingRouter.storeData(params);

System.out.printf("storeData 响应: transactionHash=%s\n", transactionHash);
} catch (IMIRpcException e) {
    e.printStackTrace();
} finally {
    // 解锁
    if (lock != null) {
        Lock.unlock(lock);
    }
}
```

2.2.3 上链数据查询接口

| | | | | | |
|-------------------|--------------------------------------|---|-------------|-----------------------------------|----------|
| 接口名称 | | 上链数据查询接口 | | | |
| 类名 | | 方法名 | | | |
| DataSharingRouter | | getData(GetDataParams params) | | | |
| 功能 | | 此接口可以通过多查询条件查询多条上链数据集合，也可以通过上链数据交易 Hash 值查询单条数据 | | | |
| 输入参数 | | | 输出参数 result | | |
| params | GetDataParams (Y) | 上链数据查询参数 | result | List< GetDataVO > | 上链数据查询结果 |

示例：

```
/****** 通过多存储条件查询多数据集合 *****/
// 数据模版类型编号
String typeCode = "10001";

// 上链数据所属用户身份证号
String subjectCid = "43072119801024008x";

// 上链数据所属用户身份证姓名
String subjectName = "向南";

// 查询上链数据条件：截止时间
String endTime = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").format(new Date());

// 查询上链数据条件：数据条数
int num = 10;
```

```

try {
    GetDataParams params = new GetDataParams();
    params.setTypeCode(typeCode);
    params.setSubjectCid(subjectCid);
    params.setSubjectName(subjectName);
    params.setIssuerVportId(CommonConstants.imiVportId); // 第三方业务系统数字身份
号(来自配置文件)
    params.setEndTime(endTime);
    params.setNum(num);

    List<GetDataVO> dataList = DataSharingRouter.getData(params);
    System.out.printf("getData      响      应      :      dataList=[%s]\n",
JSONObject.toJSONString(dataList));
} catch (IMIRpcException e) {
    e.printStackTrace();
}

/***** 通过交易 Hash 值查询单条数据 *****/
// 数据上链的交易 Hash 值
String                                transactionHash                                =
"0xc60ec052a6fe245ff9a361755814080c8196aecbdc0e218da064b8739ece71fe";

try {
    GetDataParams params = new GetDataParams();
    params.setTransactionHash(transactionHash);

    List<GetDataVO> dataList = DataSharingRouter.getData(params);
    GetDataVO vo = null;
    if (null != dataList && !dataList.isEmpty()) {
        vo = dataList.get(0);
    }
    System.out.printf("getData 响应: data=[%s]\n", JSONObject.toJSONString(vo));
} catch (IMIRpcException e) {
    e.printStackTrace();
}

```

2.2.4 上链数据验证接口

| | |
|-------------------|--|
| 接口名称 | 上链数据验证接口 |
| 类名 | 方法名 |
| DataSharingRouter | verifyData(VerifyDataParams params) |
| 功能 | 此接口可以通过上链数据查询接口获得的上链数据中的交易 Hash 值查询改数据的验证数据 |
| 输入参数 | 输出参数 result |

| | | | | | |
|--------|--------------------------------------|----------|--------|------------------------------|--------------|
| params | VerifyDataParams (Y) | 上链数据验证参数 | result | VerifyDataVO | 上链数据验证数据查询结果 |
|--------|--------------------------------------|----------|--------|------------------------------|--------------|

示例:

```
// 上链数据的交易 Hash 值
String transactionHash =
"0xc60ec052a6fe245ff9a361755814080c8196aecbdc0e218da064b8739ece71fe";

try {
    VerifyDataParams params = new VerifyDataParams();
    params.setTransactionHash(transactionHash);

    VerifyDataVO vo = DataSharingRouter.verifyData(params);

    System.out.printf("verifyData 响应: data=[%s]\n", JSONObject.toJSONString(vo));
} catch (IMIRpcException e) {
    e.printStackTrace();
}
```