数字身份 Server-SDK 接口设计

版本：V 1.0.3

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	添加 server-sdk 接口设计	2017/07/18	宋嘉		
1.0.1	M	修改授权信息获取返回方式，由回调返回改为直接返回	2017/07/20	宋嘉		
1.0.2	M	创建通信通道接口返回的通道信息实体 ChannelInfo 中添加待生成二维码的原数据字符串 qrData 属性	2017/07/27	宋嘉		
1.0.3	M	优化配置文件读取处理；将默认配置文件路径“META-INF”移除；将 sdk 依赖第三方 jar 分离出来。	2017/11/24	宋嘉		

*变化状态：A——增加，M——修改，D——删除，N——正式发布

目录

1 Server SDK 相关说明

1.1 sdk 引用

将附件“server-sdk-1.0.3-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-beanutils	1.9.2
commons-lang3	3.5
fastjson	1.2.28
apache-httpclient	4.5.2
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>commons-beanutils</groupId>
    <artifactId>commons-beanutils</artifactId>
    <version>1.9.2</version>
</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>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
```

```
    <version>4.5.2</version>
</dependency>
```

1.2 jdk 要求

server-sdk-1.0.3-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 授权信息类型定义

获取授权信息时调用 getAuthorizationInfo()方法需要传参 scope，此参数类型为字符串集合，存放需要获取的详细授权信息类型标识符，此类型定义与《第三方登录授权接口说明文档 V1.1》中的“4.3.1 二维码参数 scope 类型”章节定义一致：

授权作用域 scope	说明
snsapi_info	登录 （当第三方需要数字身份用户的基本信息时，如： 数字身份 id, 手机号等信息）
snsapi_idcard	授权 （当第三方需要数字身份用户授权身份证信息时）

1.4 配置文件说明

- server-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		
web.mapping.getTopicId	MappingServer-通信标识获取接口 URL	由我司提供
web.mapping.pullData	MappingServer-数据获取接口 URL	
基本信息		
imi.mappingOpenid	合作伙伴标识	由我司分配
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
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 server-sdk 的参数初始化配置接口“IMIConfiguration.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 server-sdk-doc-1.0.0.zip

2.1 实体定义

2.1.1 GetChannelInfoParams

名称	创建通信通道接口请求参数对象		GetChannelInfoParams
字段名	**类型（是否必填）**	**说明**	**备注**
version	String(Y)	版本号	JS-SDK 定义
scope	Set<String> (Y)	授权作用域字符集合	如：[snsapi_info,snsapi_idcard]

2.1.2 GetAuthorizationInfoParams

名称	授权信息获取接口请求参数对象		GetAuthorizationInfoParams
字段名	**类型（是否必填）**	**说明**	**备注**
topicId	String(Y)	Mapping server 通信标示	
scope	Set<String>	授权信息类型标识符集合(参考二维码参数 scope 类型)	snsapi_info：授权登录信息，snsapi_idcard：授权身份信息

2.1.3 ChannelInfo

名称	MappingServer 通道信息对象		ChannelInfo
字段名	**类型（是否必填）**	**说明**	**备注**
topicId	String(Y)	Mapping server 通信标示	作为生成二维码的数据
openId	String(Y)		
scope	String(Y)	授权作用域字符串	如：“snsapi_info,snsapi_idcard”
name	String(Y)	第三方数字身份名称	作为生成二维码的数据 此名称来源于配置文件

			imi-config.properties 中的“imi.name”参数
version	String(Y)	版本号	JS-SDK 定义
qrData	String(Y)	生成二维码的原数据串	页面上可以直接用此串来生成二维码

2.1.4 AuthorizationInfo

名称	授权详细信息对象		AuthorizationInfo
字段名	**类型（是否必填）**	**说明**	**备注**
authorizer	Authorizer(Y)	授权人信息	
identityCardInfo	IdentityCardInfo(N)	授权身份详细信息	
loginUserInfo	LoginUserInfo(N)	授权登录用户信息	

2.1.5 Authorizer

名称	授权人信息对象		Authorizer
字段名	**类型（是否必填）**	**说明**	**备注**
vportId	String(Y)	授权用户数字身份号	
name	String(Y)	授权用户名称	

2.1.6 LoginUserInfo

名称	授权用户信息对象		LoginUserInfo
字段名	**类型（是否必填）**	**说明**	**备注**
userName	String(Y)	用户名称	
mobile	String(Y)	用户手机号	
email	String(N)	用户电子邮箱	
image	Image(N)	用户头像信息	

2.1.7 IdentityCardInfo

名称	授权身份信息对象		IdentityCardInfo
字段名	**类型（是否必填）**	**说明**	**备注**
cin	String(Y)	居民身份号	
name	String(Y)	居民姓名	
sex	String(Y)	居民性别	
authority	String(Y)	签发机关	
dateBirth	String(Y)	居民出生日期	

dateIssue	String(Y)	有效期开始日期	
dateExpiry	String(Y)	有效期结束日期	
image	Image(N)	居民头像信息	

2.1.8 Image

名称	头像数据对象		Image
字段名	**类型 （是否必填）**	**说明**	**备注**
data	String(Y)	头像数据	
type	String(Y)	头像数据格式类型	

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	此接口根据用户业务需求决定是否调用，当需要通过绝对路径读取系统目录下的配置文件时使用，且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:

```
// 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();
}
```

2.2.2 通信通道创建接口

接口名称	通信通道创建接口
类名	**方法名**
IMIAuthorizationRouter	createChannel(GetChannelInfoParams params)

功能		获取 topicId 与第三方数字身份名称等相关信息，作为生成二维码的数据			
输入参数			输出参数 result		
version	String(Y)	版本号	result	ChannelInfo	通信通道信息
scope	Set<String> (Y)	授权作用域字符集合			

示例：

```
try {
    // 版本数据与授权域数据根据 JS-SDK 的定义
    String version = "2.0";
    Set<String> scope = new HashSet<String>();
    scope.add("snsapi_info");
    scope.add("snsapi_idcard");

    GetChannelInfoParams params = new GetChannelInfoParams();
    params.setScope(scope);
    params.setVersion(version);

    ChannelInfo info = IMIAuthorizationRouter.createChannel(params);
    System.out.printf("createChannel 响应：ChannelInfo=[%s]\n",
JSONObject.toJSONString(info));
} catch (IMIRpcException e) {
    e.printStackTrace();
}
```

2.2.3 授权信息获取接口

接口名称		授权信息获取接口			
类名		方法名			
IMIAuthorizationRouter		getAuthorizationInfo(GetAuthorizationInfoParams params)			
功能		此接口从 MappingServer 获取 app 推送的授权详细信息			
输入参数			输出参数 result		
topicId	String(Y)	Mapping server 通信标示	result	AuthorizationInfo	授权信息
scope	Set<String>	授权信息类型标识符集合（参考二维码参数 scope 类型）			

示例：

```
try {
    // 通信标识，通过 createChannel()接口获得
    String topicId = "85a2c1c03501421ba86cfae6e263acfa";
```

```
    // 需要获取的授权信息类型集合
    Set<String> scopeSet = new HashSet<String>();
    scopeSet.add("snsapi_info");        // 获取用户信息
    scopeSet.add("snsapi_idcard");      // 获取身份信息

    GetAuthorizationInfoParams params = new GetAuthorizationInfoParams();
    params.setTopicId(topicId);
    params.setScope(scopeSet);

    AuthorizationInfo authorizeInfo =
IMIAuthorizationRouter.getAuthorizationInfo(params);
    System.out.printf("getAuthorizationInfo 响应：AuthorizationInfo=[%s]\n",
JSONObject.toJSONString(authorizeInfo));
} catch (IMIRpcException e) {
    e.printStackTrace();
}
```