浪潮-分布式数字身份 SDK

文档修订记录

版本	时间	变更内容	作者
V0.1	2022-03-03	文档创建（生成公私钥、IID、DDoc、签名方法、验签方法接口）	冉悦
V0.2	2022-03-11	添加 IID 格式和对应 Document 敏感词的检测接口	冉悦
V0.3	2022-3-15	修改接口调用方式，添加远程 maven 仓库使用方式	冉悦
V0.4	2022-4-1	添加 diddocument 的签名和验签	冉悦
V0.5	2022-4-2	添加通用 bean 的签名和验签	冉悦

Inspur-IID-SDK　jar 包的使用

1. 远程 maven 仓库使用

1.1 正确配置 maven XML

具体 maven 配置请参照附件，修改附件中的 localRepository 为自己的 maven localRepository

1.2maven 中引入 jar 包依赖

```
<dependency>
      <groupId>com.inspur</groupId>
      <artifactId>qualitychain-baas-iidsdk</artifactId>
      <version>1.0-SNAPSHOT</version>
</dependency>
```

Inspur-IID-SDK 接口

1.获取 SDK 版本

接口名称：获取 SDK 版本接口

接口说明：获取 SDK 版本

示例使用代码：

```
IIDUtil.getSdkVersion();
```

接口返回数据：

名称	类型	是否必选项	示例值	描述
version	String	是	V1.0.1	SDK version

2.获取浪潮分布式数字身份-IID

生成 IID 是浪潮分布式系统的核心前提。

2.1 生成公私钥对

生成两对公私钥对作为主和备公私钥。公私钥生成算法暂时支持 secp256k1 和 RSA。

用 secp256k1 生成的生成公私钥对格式如下：

```
“privateKey”:”bdd776089322f1aa2cf762571c096505901d23e8d7c55d5e8e80c49d4e66a010”,
“publicKey”:”04cf18d5efde23cd366f3be1b0b49c98da0f669306e04e9e87d06742b39761eb77ae5fdd240fa
ac2a05f2a820957a3df97352a09aa2cd27c6763daf26717205256”,
type:” secp256k1”
```

2.2 生成 Base DID Document

生成的 base did document 如下:

```
{
"authentication":["#key-1"],
"context":"https://w3id.org/did/v1",
"publicKey":[
    {
    "id":"#key1",
    "publicKeyHex":"04720fc77955f9d6f832410635d41cbe50e78fdf655c2821bb660aec0d5e3f4d
6c7588dee06d18d1b68729d8cb3f7887fa575bf496ab71f8720dc522d737d3bb53",
    "type":"secp256k1"
    },
    {
    "id":"#key2",
    "publicKeyHex":"04595146ade4f5f9aebbbdd0069740fce82b3ec89c1c99e0ccbaad6db45df0da
d5924b32500390c74dee248c8547da0a90856fa8a55c186494e97385e04784e3c3",
    "type":"secp256k1"
    }
],
"recovery":["#key-2"]
}
```

2.3 对 Base DID Document 做 sha256

结果格式如下：

```
d842e1499150fd334959f5f0e74a83317a424709f57ca405c9317352b2051ebf
```

2.4 对哈希结果做 ripemd160

结果格式如下：

```
8a2fc470eb6ea91d591695528ade8003a808831f
```

2.5 对 RIPEMD 结果做 base58

结果格式如下：

```
3oUnm6HEcWkKAQ6Nd7sPax4BD1MH2nEK1vTFSwaFvccKrPhWob8AjsK
```

2.6 添加“did:iid:”前缀

结果格式如下：

```
did:iid:3oUnm6HEcWkKAQ6Nd7sPax4BD1MH2nEK1vTFSwaFvccKrPhWob8AjsK
```

3 接口使用

接口名称：获取 iid 接口

接口说明：获取浪潮分布式数字身份 iid

示例使用代码：

```
1.  ResponseResult responseResult = null;
2.          try {
3.              responseResult = IIDUtil.getIID("secp256k1");
4.          } catch (Exception e) {
5.              e.printStackTrace();
6.          }
7.  }
```

接口返回数据：

名称	类型	是否必返项	示例值	描述
message	String	是	获取 IID 成功	返回结果是否成功描述
data	Object	是		返回结果对象
data.iid	String	是	did:iid:3Z71YPzReEDnt9AFErHreCRFbrMJ7tVjiCBrbJAzp8j6ma3P8fSpWHQ	iid
data.keyPairsMap	Map	是	{"master":{"privateKey":"c25296b0c3c0d1aec11e52a5c4667f19b886e2c2b78101c5c5b714026bda0cf1","publicKey":"0407aa60bb24ed9504b6ae62568c86e955ac58a68a02da5999c520e0fdaa43ea585cbb17114470bef08cbffab705f8766e002290a63a5acb3c39db126cd2b87e6d","type":"secp256k1"},"slave":{"privateKey":"c12d0965cdd8dd454e184e790fef22900a3fae68cafe0752a403e525f5872c89","publicKey":"0442c6c9400349acc4f469fc82f44ee360ca7e5f3246fb7788c14d2c79b2ae0df87f8d9fcbd6a656a6755e25d454f5d8a56c3db095f44d78521c004347514a4170","type":"secp256k1"}}	主备公私钥对
data.didDocument	Object	是		初始 DID 文档
data.didDocument.context	String	是	https://w3id.org/did/v1	DID 文档 context
data.didDocument.id	String	是	did:iid:3Z71YPzReEDnt9AFErHreCRFbrMJ7tVjiCBrbJAzp8j6ma3P8fSpWHQ	DID 文档 id
data.didDocument.version	String	是	"1"	DID 文档版本
data.didDocument.created	String	是	"2022-03-04T11:18:27.335"	DID 文档创建时间
data.didDocument.updated	String	是	"2022-03-04T11:18:27.335"	DID 文档更新时间

data.didDocument.publicKey	list	是	[{"id":"did:iid:3Z71YPzReEDnt9AFErHreCRFbrMJ7tVjiCBrbJAzp8j6ma3P8fSpWHQ#key1","publicKeyHex":"0407aa60bb24ed9504b6ae62568c86e955ac58a68a02da5999c520e0fdaa43ea585cbb17114470bef08cbffab705f8766e002290a63a5acb3c39db126cd2b87e6d","type":"secp256k1"},{"id":"did:iid:3Z71YPzReEDnt9AFErHreCRFbrMJ7tVjiCBrbJAzp8j6ma3P8fSpWHQ#key2","publicKeyHex":"0442c6c9400349acc4f469fc82f44ee360ca7e5f3246fb7788c14d2c79b2ae0df87f8d9fcbd6a656a6755e25d454f5d8a56c3db095f44d78521c004347514a4170","type":"secp256k1"}]	DID 文档公钥列表
data.didDocument.authentication	list	是	["did:iid:3Z71YPzReEDnt9AFErHreCRFbrMJ7tVjiCBrbJAzp8j6ma3P8fSpWHQ#key-1"]	公钥对应的私钥为该文档拥有者
data.didDocument.recovery	list	是	["did:iid:3Z71YPzReEDnt9AFErHreCRFbrMJ7tVjiCBrbJAzp8j6ma3P8fSpWHQ#key-2"]	可用于恢复的公钥列表
data.didDocument.service	list	否		一些能够使用此 DID 的 终端服务
data.didDocument.proof	Object	否		可验证声明
data.didDocument.operation	String	是	"create"	DID 文档执行操作
data.didDocument.timestamp	String	是	1646363907	DID 文档创建时间戳

3.获取签名

接口名称：获取签名

接口说明：获取签名接口

示例使用代码：

```
1.  String priKey = "c25296b0c3c0d1aec11e52a5c4667f19b886e2c2b78101c5c5b714026bda0cf1";
2.  String msg = "example sign message";
3.  String type = "secp256k1";
4.  ResponseResult sign = IIDUtil.sign(priKey,msg,type);
```

请求参数：

名称	类型	是否必填	描述
privateKey	String	是	签名私钥
message	String	是	签名的对象信息
type	String	是	签名算法

返回数据：

名称	类型	是否必选项	示例值	描述
message	String	是	"签名成功"	返回消息
data	Object	是		返回对象
data.msg	String	是	"example sign message"	签名的对象信息
data.sig	String	是	"30440220b49837ca136be2d06af1d7b6381db9464b30faf8eb9c4023b4c1d2afaff66c270220671b5c2ec5cedaaa9ea613d5dddff6be309de035eae8c7699597b2666f91b11f"	生成的签名
data.type	String	是	"secp256k1"	生成签名的算法

4.验证签名

接口名称：验证签名

接口说明：验证签名接口

示例使用代码：

```
1.  boolean verify = IIDUtil.verify(msg, sig, publicKey, type);
```

请求参数：

名称	类型	是否必填	描述
publicKey	String	是	验签公钥
message	String	是	签名的信息
signature	String	是	签名
type	String	是	签名算法

返回数据：

名称	类型	是否必选项	示例值	描述
result	boolean	是	"验证成功"	返回消息

5.验证 IID

接口名称：验证 IID

接口说明：验证 IID 的格式和对应 document 中的敏感词

示例使用代码：

```
String iid = "did:iid:3Z71YPzReEDnt9AFErHreCRFbrMJ7tVjiCBrbJAzp8j6ma3P8fSpWHQ";
boolean res = false;
boolean res1 = false;
boolean res2 = false;
try {
      res = IIDUtil.validateIID(iid);
      res1 = IIDUtil.validateIID(iid, "yyds");
      res2 = IIDUtil.validateIID(iid, "shit");
 } catch (Exception e) {
      e.printStackTrace();
}
```

请求参数：

名称	类型	是否必填	描述
iid	String	是	需验证的 IID
didDocument	String	否	需验证的 didDocument

返回数据：

名称	类型	是否必选项	示例值	描述
result	boolean	是	True	验证结果

6.公钥加密

接口名称：公钥加密

接口说明：使用公钥对信息加密

示例使用代码：

```
String pub = "043a010a0147bba4b56574a8d9bafffd1532a8105f2dc461cdee511a4885075f3193e0bf9470eca15d50eded6c4e978b3c21b476b66101f914429247ca301daed1";
String pri = "04f5a623add5f6ce8efca5da6786f90b342e0fb3f6e2712e8c9490c2f602da7e";
String msg = "{\n" +
                " \"data\": {\"name\":\"zoe\",\"age\":\"20\"},\n" +
                " \"did\": \"did:iid:3bVRS6dRxikwbvTcEfHLhKpB4kSA3Fd1fLp8n6Wid5wQEmMKvPqhqn6\",\n" +
                "\"status\": \"1\"\n" +
                "}";
String encrypt = IIDUtil.encrypt(pub, msg);
String decrypt = IIDUtil.decrypt(pri, encrypt);
System.out.println(msg.equals(decrypt));
```

请求参数：

名称	类型	是否必填	描述
pub	String	是	公钥
msg	String	是	需加密的原始消息

返回数据：

名称	类型	是否必选项	示例值	描述
result	String	是	True	加密后的结果的十六进制字符串

7.私钥解密

接口名称：私钥解密

接口说明：用私钥对公钥加密过的信息解密

示例使用代码：

```
String pub = "043a010a0147bba4b56574a8d9bafffd1532a8105f2dc461cdee511a4885075f3193e0bf9470eca15d50eded6c4e978b3c21b476b66101f914429247ca301daed1";
String pri = "04f5a623add5f6ce8efca5da6786f90b342e0fb3f6e2712e8c9490c2f602da7e";
String msg = "{\n" +
                " \"data\": {\"name\":\"zoe\",\"age\":\"20\"},\n" +
                " \"did\": \"did:iid:3bVRS6dRxikwbvTcEfHLhKpB4kSA3Fd1fLp8n6Wid5wQEmMKvPqhqn6\",\n" +
                "\"status\": \"1\"\n" +
                "}";
String encrypt = IIDUtil.encrypt(pub, msg);
String decrypt = IIDUtil.decrypt(pri, encrypt);
System.out.println(msg.equals(decrypt));
```

请求参数：

名称	类型	是否必填	描述
Pri	String	是	私钥
encodedMsg	String	是	公钥加密处理之后的十六进制字符串

返回数据：

名称	类型	是否必选项	示例值	描述
result	String	是	True	原始信息字符串

8.对 DID 文档签名

接口名称：对 did 文档签名

接口说明：获取对 did 文档签名 s

示例使用代码：

```
ResponseResult sign = IIDUtil.signDoc(priKey,doc);
```

请求参数：

名称	类型	是否必填	描述
privateKey	String	是	签名私钥
doc	DIDdocument	是	did 文档

返回数据：

名称	类型	是否必选项	示例值	描述
message	String	是	"签名成功"	返回消息
data	Object	是		返回对象
data.sig	String	是	"30440220b49837ca136be2d06af1d7b6381db9464b30faf8eb9c4023b4c1d2afaff66c270220671b5c2ec5cedaaa9ea613d5dddff6be309de035eae8c7699597b2666f91b11f"	生成的签名
data.type	String	是	"secp256k1"	生成签名的算法

9.验证对 DID 文档的签名

接口名称：验证对 did 文档的签名

接口说明：验证对 did 文档的签名

示例使用代码：

```
2.  boolean verify = IIDUtil.verifyDocSign(publicKey, doc);
```

请求参数：

名称	类型	是否必填	描述
publicKey	String	是	公钥
doc	DIDdocument	是	did 文档

返回数据：

名称	类型	是否必选项	示例值	描述
result	boolean	是	true	返回消息

10．通用对象的签名

接口名称：对通用对象签名

接口说明：获取对通用对象签名

示例使用代码：

```
@SignExclude("sig")
public class SignExcludeTestClass {
    String iid;
    String uuid;
    @JsonProperty("sig")
    String sig;
}
```

```
@Test
    public void SignJsonTest() {
        SignExcludeTestClass signExcludeTestClass = new SignExcludeTestClass();
        signExcludeTestClass.setIid("111111");
        signExcludeTestClass.setUuid("2222");
        String s = IIDUtil.signJson("04f5a623add5f6ce8efca5da6786f90b342e0fb3f6e2712e8c9490c2f602da7e", signExcludeTestClass);

        signExcludeTestClass.setSig(s);

        boolean b = IIDUtil.verifySignedJson("043a010a0147bba4b56574a8d9bafffd1532a8105f2dc461cdee511a4885075f3193e0bf9470eca15d50eded6c4e978b3c21b476b66101f914429247ca301daed1", s, signExcludeTestClass);
        System.out.println(b);
    }
```

请求参数：

名称	类型	是否必填	描述
privateKey	String	是	签名私钥
object	Object	是	签名对象

返回数据：

名称	类型	是否必选项	示例值	描述
sig	String	是	″30440220b49837ca136be2d06af1d7b6381db9464b30faf8eb9c4023b4c1d2afaff66c270220671b5c2ec5cedaaa9ea613d5dddff6be309de035eae8c7699597b2666f91b11f″	生成的签名

11. 通用对象的验签

接口名称：通用对象签名的验签

接口说明：验证通用对象签名的验签

示例使用代码：

```java
@SignExclude("sig")
public class SignExcludeTestClass {
    String iid;
    String uuid;
    @JsonProperty("sig")
    String sig;
}
```

```java
@Test
    public void SignJsonTest() {
        SignExcludeTestClass signExcludeTestClass = new SignExcludeTestClass();
        signExcludeTestClass.setIid("111111");
        signExcludeTestClass.setUuid("2222");
        String s = IIDUtil.signJson("04f5a623add5f6ce8efca5da6786f90b342e0fb3f6e2712e8c9490c2f602da7e", signExcludeTestClass);

        signExcludeTestClass.setSig(s);

        boolean b = IIDUtil.verifySignedJson("043a010a0147bba4b56574a8d9bafffd1532a8105f2dc461cdee511a4885075f3193e0bf9470eca15d50eded6c4e978b3c21b476b66101f914429247ca301daed1", s, signExcludeTestClass);
        System.out.println(b);
    }
```

请求参数：

名称	类型	是否必填	描述
publicKey	String	是	验签公钥
object	Object	是	签名对象
sig	String	是	签名

返回数据：

名称	类型	是否必选项	示例值	描述
result	boolean	是	True	返回消息