

# CM-SDK

1.0.0

## SDK 使用说明

### 前置操作

引入 SDK 包

### 使用流程

配置服务信息 -> 调用所需方法 -> 使用SDK返回的数据

### 功能

- 1.身份认证
- 2.根据所需密钥数量获取国密SM2非对称密钥
- 3.根据已知的密钥编号获取国密SM2非对称密钥
- 4.根据所需密钥数量获取国密SM4对称密钥
- 5.根据已知的密钥编号获取国密SM4对称密钥
- 6.国密SM4加密
- 7.国密SM4解密
- 8.国密SM3摘要计算
- 9.国密SM2签名计算
- 10.国密SM2签名验证
- 11.国密SM2加密算法
- 12.国密SM2解密算法

### 示例

- 1.实例化CMClient对象(配置服务信息)

```
1
2 CMClient cmClient = new CMClient("clientId", "clientSecret", "pubkey",
  "prikey");
3 // 以下是全参数配置
4 // CMClient cmClient = new CMClient("ip", "port", "prefix", "clientId",
  "clientSecret", "pubkey", "prikey");
5
```

参数说明

```

1 String ip = 'IP地址或者域名' //默认 http://api.zjquantum.cn
2 String port = '端口号' //默认 80
3 String clientId = 'clientId' //由SDK提供方提供
4 String clientSecret = 'clientSecret' //由SDK提供方提供
5 String pubKey = '公钥' //由SDK提供方提供
6 String priKey = '私钥' //由SDK提供方提供
7 String prefix = '' // 请求前缀 //由SDK提供方提供，一般无需配置

```

2.调用所需的方法（以 根据所需密钥数量获取国密SM2非对称密钥 为例）

```

1 SM2KeyGenerateByNumberRequest requestParam = new
  SM2KeyGenerateByNumberRequest(1, "HEX");
2 SM2GenerateKeyResponse sm2GenerateKeyResponse =
  CMClient.sm2GenerateKeyPairByNumber(requestParam);

```

3.获取SDK返回的数据（返回结果为json格式）

```

1 // 成功
2 {
3     "requestId":"531ebfcb-b863-4af4-ab64-f594634f511f",
4     "code":0,
5     "msg":"success",
6     "resEncode":"hex",
7
8     "hashText":"9A3628E03508AAE81C0D78ABA35E7186C47E804EC8E05C41B7DA15635CEA097
9     3"
10 }
11 // 失败
12 {
13     "requestId":"fd55a717-6f83-486d-b18f-4c5645c6a53d",
14     "code":20001,
15     "msg":"签名输出结果编码类型必须为HEX或BASE64"
16 }

```

## 示例

PS: 调用方法前请务必实例化 `CMClient`

### 1.身份认证

```

1 CMClient CMClient = new
  CMClient("ip","port","prefix","clientId","clientSecret","pubKey","priKey");
2 CMClient.getToken();

```

### 2.根据所需密钥数量获取国密SM2非对称密钥

```

1 SM2KeyGenerateByNumberRequest request = new
  SM2KeyGenerateByNumberRequest(2, "resEncode");
2 SM2GenerateKeyResponse sm2GenerateKeyResponse =
  CMClient.sm2GenerateKeyPairByNumber(request);

```

参数说明

参数	参数说明	可传内容	默认值
resEncode	结果返回编码格式	HEX/BASE64	BASE64

### 3.根据密钥编号获取非对称密钥

```
1 List<String> keyIds = new ArrayList<>();
2 keyIds.add("keyId1"); // 设置已知的密钥编号
3 keyIds.add("keyId2");
4 SM2KeyGenerateByKeyIdsRequest request = new
SM2KeyGenerateByKeyIdsRequest(keyIds, "resEncode");
5 SM2GenerateKeyResponse sm2GenerateKeyResponse =
CMClient.sm2GenerateKeyPairById(request);
```

参数说明

参数	参数说明	可传内容	默认值
resEncode	结果返回编码格式	HEX/BASE64	BASE64

### 4.根据所需密钥数量获取国密SM4对称密钥

```
1 SM4KeyGetKeyByNumberRequest request = new SM4KeyGetKeyByNumberRequest(5,
"resEncode");
2 SM4KeyResponse sm4KeyResponse = CMClient.sm4GetKeyByNumber(request);
```

参数说明

参数	参数说明	可传内容	默认值
resEncode	结果返回编码格式	HEX/BASE64	BASE64

### 5.根据已知的密钥编号获取国密SM4对称密钥

```
1 List<String> keyIds = new ArrayList<>();
2 keyIds.add("keyId1"); // 设置已知的密钥编号
3 keyIds.add("keyId2");
4 SM4GetKeyByIdRequest request = new SM4GetKeyByIdRequest(keyIds,
"resEncode");
5 SM4KeyResponse sm4KeyResponse = CMClient.sm4GetKeyById(request);
```

参数说明

参数	参数说明	可传内容	默认值
resEncode	结果返回编码格式	HEX/BASE64	BASE64

## 6. 国密SM4加密

```
1 SM4EncryptRequest sm4EncryptRequest = new SM4EncryptRequest("key",
    "keyEncode", "iv", "ivEncode", "mode", "plaintext", "plaintextEncode",
    "resEncode");
2 SM4EncRes sm4EncRes = CMClient.sm4EncData(sm4EncryptRequest);
```

### 参数说明

参数	参数说明	可传内容	默认值
plaintextEncode	明文（待加密数据）编码格式	UTF8/HEX/BASE64	UTF8
keyEncode	密钥编码格式	HEX/BASE64	BASE64
mode	SM4加密模式	ECB/CBC	-
resEncode	结果返回编码格式	HEX/BASE64	BASE64
ivEncode	IV向量编码格式	HEX/BASE64	BASE64

## 7. 国密SM4解密

```
1 SM4DecryptRequest sm4DecryptRequest = new SM4DecryptRequest("key",
    "keyEncode", "iv", "ivEncode", "mode", "ciphertext", "ciphertextEncode",
    "resEncode");
2 SM4DecRes sm4DecRes = CMClient.sm4DecData(sm4DecryptRequest);
```

### 参数说明

参数	参数说明	可传内容	默认值
ciphertextEncode	密文（待解密数据）编码格式	HEX/BASE64	BASE64
keyEncode	密钥编码格式	HEX/BASE64	BASE64
mode	SM4加密模式	ECB/CBC	-
resEncode	结果返回编码格式	UTF8/HEX/BASE64	UTF8
ivEncode	IV向量编码格式	HEX/BASE64	BASE64

## 8. 国密SM3摘要计算

```
1 SM3HashRequest sm3HashRequest = new SM3HashRequest("message",
    "messageEncode", "resEncode");
2 SM3HashRes sm3HashRes = CMClient.hash(sm3HashRequest);
```

### 参数说明

参数	参数说明	可传内容	默认值
messageEncode	待计算数据编码格式	UTF8/HEX/BASE64	UTF8
resEncode	结果返回编码格式	UTF8/HEX/BASE64	UTF8

## 9.国密SM2签名计算

```
1 SM2SignRequest sm2SignRequest = new SM2SignRequest("publicKey",
  "privateKey", "keyEncode", "userId", "userIdEncode", "plaintext",
  "plaintextEncode", "resEncode");
2 SM2SignRes sm2SignRes = CMClient.sm2Sign(sm2SignRequest);
```

### 参数说明

参数	参数说明	可传内容	默认值
userIdEncode	签名者ID编码格式	UTF8/HEX/BASE64	UTF8
keyEncode	密钥编码格式	HEX/BASE64	BASE64
plaintextEncode	待签名数据编码格式	UTF8/HEX/BASE64	UTF8
resEncode	结果返回编码格式	HEX/BASE64	HEX/BASE64

## 10.国密SM2签名验证

```
1 SM2VerifyRequest sm2VerifyRequest = new SM2VerifyRequest("publicKey",
  "keyEncode", "userId", "userIdEncode", "signData", "signDataEncode",
  "plaintext", "plaintextEncode");
2 SM2VerifySignRes sm2VerifySignRes =
  CMClient.sm2VerifySign(sm2VerifyRequest);
```

### 参数说明

参数	参数说明	可传内容	默认值
userIdEncode	签名者ID编码格式	UTF8/HEX/BASE64	UTF8
keyEncode	密钥编码格式	HEX/BASE64	BASE64
plaintextEncode	待验证数据（原文）编码格式	UTF8/HEX/BASE64	UTF8
signDataEncode	签名数据编码格式	HEX/BASE64	BASE64

## 11.国密SM2加密算法

```
1 SM2EncryptRequest sm2EncryptRequest = new SM2EncryptRequest("publicKey",
  "keyEncode", "plaintext", "plaintextEncode", "resEncode");
2 SM2EncRes sm2EncRes = CMClient.sm2EncData(sm2EncryptRequest);
```

### 参数说明

参数	参数说明	可传内容	默认值
keyEncode	密钥编码格式	HEX/BASE64	BASE64
plaintextEncode	待加密数据（原文）编码格式	UTF8/HEX/BASE64	UTF8
resEncode	加密结果数据编码格式	HEX/BASE64	BASE64

## 12.国密SM2解密算法

```
1 SM2DecryptRequest sm2DecryptRequest = new SM2DecryptRequest("privateKey",
2   "keyEncode", "ciphertext", "ciphertextEncode", "resEncode");
   SM2DecRes sm2DecRes = CMClient.sm2DecData(sm2DecryptRequest);
```

### 参数说明

参数	参数说明	可传内容	默认值
keyEncode	密钥编码格式	HEX/BASE64	BASE64
ciphertextEncode	密文（带解密）数据编码格式	HEX/BASE64	BASE64
resEncode	解密结果数据编码格式	UTF8/HEX/BASE64	UTF8

## 错误代码

错误代码	错误内容	建议解决方法
10001	获取token失败	检查ip、端口、请求前缀（格式为"/api/v1"、"/pro"）、用户公私钥信息是否正确
20001	签名输出结果编码类型必须为HEX或BASE64	检查输出数据编码类型是否为HEX或BASE64
20002	签名密钥编码类型必须为HEX或BASE64	检查密钥编码类型是否为HEX或BASE64
30001	验签密钥编码类型必须为HEX或BASE64	检查密钥编码类型是否为HEX或BASE64
40001	SM2加密输出结果编码类型必须为HEX或BASE64	检查输出数据编码类型是否为HEX或BASE64
40002	SM2加密密钥编码类型必须为HEX或BASE64	检查密钥编码类型是否为HEX或BASE64
50001	SM2解密密钥编码类型必须为HEX或BASE64	检查密钥编码类型是否为HEX或BASE64
50002	SM4加密输出结果编码类型必须为HEX或BASE64	检查输出数据编码类型是否为HEX或BASE64
60000	sm2获取密钥对输出结果编码必须为HEX或BASE64	检查输出数据编码类型是否为HEX或BASE64
60001	获取密钥对失败	检查参数是否正确及数据和数据对应编码格式是否正确
70001	获取密钥失败	检查参数是否正确及数据和数据对应编码格式是否正确
70002	sm4获取密钥输出结果编码必须为HEX或BASE64	检查输出数据编码类型是否为HEX或BASE64
80000	sm3杂凑运算输出结果编码必须为HEX或BASE64或UTF8	检查输出数据编码类型是否为HEX或BASE64