CKEY HTTP API For Web Application

接口说明

接口文档

文档历史

序号	版本	修改说明	编写人	审核人	审批人	时间
01	v1.0	创建文档	Chappy			2019-11-15
02	V1.1	增加 Postman 测试实例章节	Chappy			2019-12-5

目录

1. 接口

接口地址 http[s]://IP:[PORT]/aaa-api/api/auth
接口请求方式：post

说明：IP 表示 CKEY 服务器 IP 地址，端口是 CKEY 服务器端口。CKEY 服务器默认采用 https 连接 TCP 端口为 443。本接口文档适用于 iAMS3.x.x 之上的版本。

2. 参数

接口参数列表

参数名称	参数类型	参数值	说明	必传
clientName	string	设备名称	此值必须与 ckey 服务器上所设置的名称相同，	是
sharedSecret	string	共享密码	加密秘钥 加密秘钥说明： 1. 加密算法 AES/CBC/PKCS7Padding 2. 密码等于 16 位时，直接用共享密码加密密码 3. 密码大于 16 位，从 0 位开始取一共取 16 个字符，将取出的字符串用共享密码进加密； 4. 密码小于 16 位时，把不足 16 位的部分用“0”填充在密码左边，然后在用共享密码进行加密。	是
loginCode	string	登陆账号	登陆用户名	是
loginWord	string	登陆密码	登陆密码	是
authType	integer(int32)	认证类型	认证类型： “0”：握手 “1”：单步静态认证 “2”：单步动态认证 “3”：单步混合认证	是

			“4”: 多步认证	
sessionId	string	认证会话	在多不认证时为必传参数	否
deviceType	integer(int32)	设备类型	“6”：httpAPI 类型	是
seedMessage	string	短信参数	多步认证时为必传参数 “0”：表示该用户不发短信口令， “1”：表示该用户发送短信口令	否
serviceId	string	服务器 ID(服务信息中序列号)		是

3. 接口返回值

参数名称	说明
Accept	成功
Challenge	challenge 请求
Error_code	具体信息建接口状态码章节

4. 接口状态码

状态码	参数名称	说明
200	AUTH_PASS	认证成功
201	AUTH_CHALLENGE	challenge 请求
202	USER_NOT_EXISTS	用户不存在
203	AUTH_EQUIP_ERROR	认证设备信息错误
210	AUTH_EQUIP_ADDRESS_ERROR	认证设备地址错误
204	AUTH_CONFIG_ERROR	认证服务器配置错误
205	PASSWORD_ERROR	密码/密钥错误

206	NULL_PASSWORD	密码为空
207	PAR_ERROR	参数错误
208	SERVICE_ID_ERROR	服务 ID 错误

5. 实例代码

1　码加密实例

共享密码为：123456
加密算法：AES/CBC/PKCS7Padding
共享密码密文: O4SkNWTfpKVOSrvpdcwbXg==
说明：共享密码加密种子为共享密码本身

1.1 密码长度等于 16 位

密码等于 16 位时，直接用共享密码加密密码
实例代码
密码码种子（共享密码）：123456
密码明文：!QAZ2wsx87127711
密码密文：Ay08qVMovipwdZ5q0KDGC0p2oaEmfJE4uoXpKLhYK80=

1.2 密码大于 16 位

密码大于 16 位，从 0 位开始取一共取 16 个字符，将取出的字符串用共享密码进加密。
实例代码
密码码种子（共享密码）：123456
密码明文：!QAZ2wsx8712771111
密码密文：Ay08qVMovipwdZ5q0KDGC9S2gH358HG9iYD61w0sGiw==

1.3 密码小于 16 位

密码小于 16 位时，把不足 16 位的部分用"0"填充在密码左边，然后在用共享密码进行加密。

实例代码

密码码种子（共享密码）：123456

密码明文：!QAZ2wsx871277

密码密文：ywyUTECTaLx3FLP+E/9mFw==

2 认证类型实例

以下是 java 实现认证实例，其他语言请使用请使用类似 java 的方法实现.。

1.4 单步静态认证

authType 1

实例代码

```
/**
 * 单步静态密码认证
 */
@Test
public void staticPasswordTest() throws Exception {
    Map<String, String> parameterMap = new HashMap<String, String>();
    parameterMap.put("clientName", "http-001");
                        parameterMap.put("sharedSecret",  AESUtils.encrypt("123456",
"123456")); //"123456"为共享密钥
    parameterMap.put("loginCode", "test001");
    parameterMap.put("loginWord", AESUtils.encrypt("!QAZ2wsx", "123456"));
    parameterMap.put("authType", "1");
    parameterMap.put("deviceType", "6");
    parameterMap.put("serviceId", "HS9EA6AD");
    ResponseInfo responseInfo = executeRequest(parameterMap);
    Assert.assertEquals(200,responseInfo.getCode());
}
```

1.5 单步动态认证

authType 2
实例代码

```
/**
 * 单步动态密码认证
 */
@Test
public void dynaPasswordTest() throws Exception {
    Map<String, String> parameterMap = new HashMap<String, String>();
    parameterMap.put("clientName", "http-001");
                        parameterMap.put("sharedSecret",    AESUtils.encrypt("123456",
"123456")); //"123456"为共享密钥
    parameterMap.put("loginCode", "chappy");
    parameterMap.put("loginWord", AESUtils.encrypt("872110", "123456")); //动态密码
    parameterMap.put("authType", "2");
    parameterMap.put("deviceType", "6");
    parameterMap.put("serviceId", "HS9EA6AD");
    ResponseInfo responseInfo = executeRequest(parameterMap);
    Assert.assertEquals(200,responseInfo.getCode());
   }
```

1.6 单步混合认证

authType 3
实例代码

```
/**
 * 混合密码认证
 */
@Test
public void hybridPasswordTest() throws Exception {
    Map<String, String> parameterMap = new HashMap<String, String>();
    parameterMap.put("clientName", "http-001");
                        parameterMap.put("sharedSecret",    AESUtils.encrypt("123456",
"123456")); //"123456"为共享密钥
    parameterMap.put("loginCode", "chappy");
```

```
1                parameterMap.put("loginWord", AESUtils.encrypt("!QAZ2wsx8712771111",
"123456")); //Exigen@2012 为静态密码，473171 为动态口令
        parameterMap.put("authType", "3");
        parameterMap.put("deviceType", "6");
        parameterMap.put("serviceId", "HS9EA6AD");
        ResponseInfo responseInfo = executeRequest(parameterMap);
        Assert.assertEquals(200,responseInfo.getCode());
    }
```

1.7 多步认证

多步认证用短信认证时，需要将 seedMessage 参数带上（0/1）
0：表示不发短信
1：表示发送短信
参数 sessionId 必须参数

authType 4
实例代码

```
/**
     * 多步认证
     */
    @Test
    public void multiStepTest() throws Exception {
        Map<String, String> parameterMap = new HashMap<String, String>();
        parameterMap.put("clientName", "http-001");
                                parameterMap.put("sharedSecret",    AESUtils.encrypt("123456",
"123456"));  //"123456"为共享密钥
        parameterMap.put("loginCode", "alice01");
             parameterMap.put("loginWord", AESUtils.encrypt("Exigen@2012", "123456"));
//Exigen@2012 为静态密码
        parameterMap.put("authType", "4");
        parameterMap.put("deviceType", "6");
        parameterMap.put("serviceId", "HSAC851E");
        parameterMap.put("seedMessage","0"); //0.不会发短信口令，1.发送短信口令
        ResponseInfo responseInfo = executeRequest(parameterMap);
        Assert.assertEquals(201,responseInfo.getCode());
        Assert.assertNotNull(responseInfo.getContent());
        parameterMap.put("sessionId",responseInfo.getContent());
          parameterMap.put("loginWord", AESUtils.encrypt("238768", "123456")); //238768
为动态密码
```

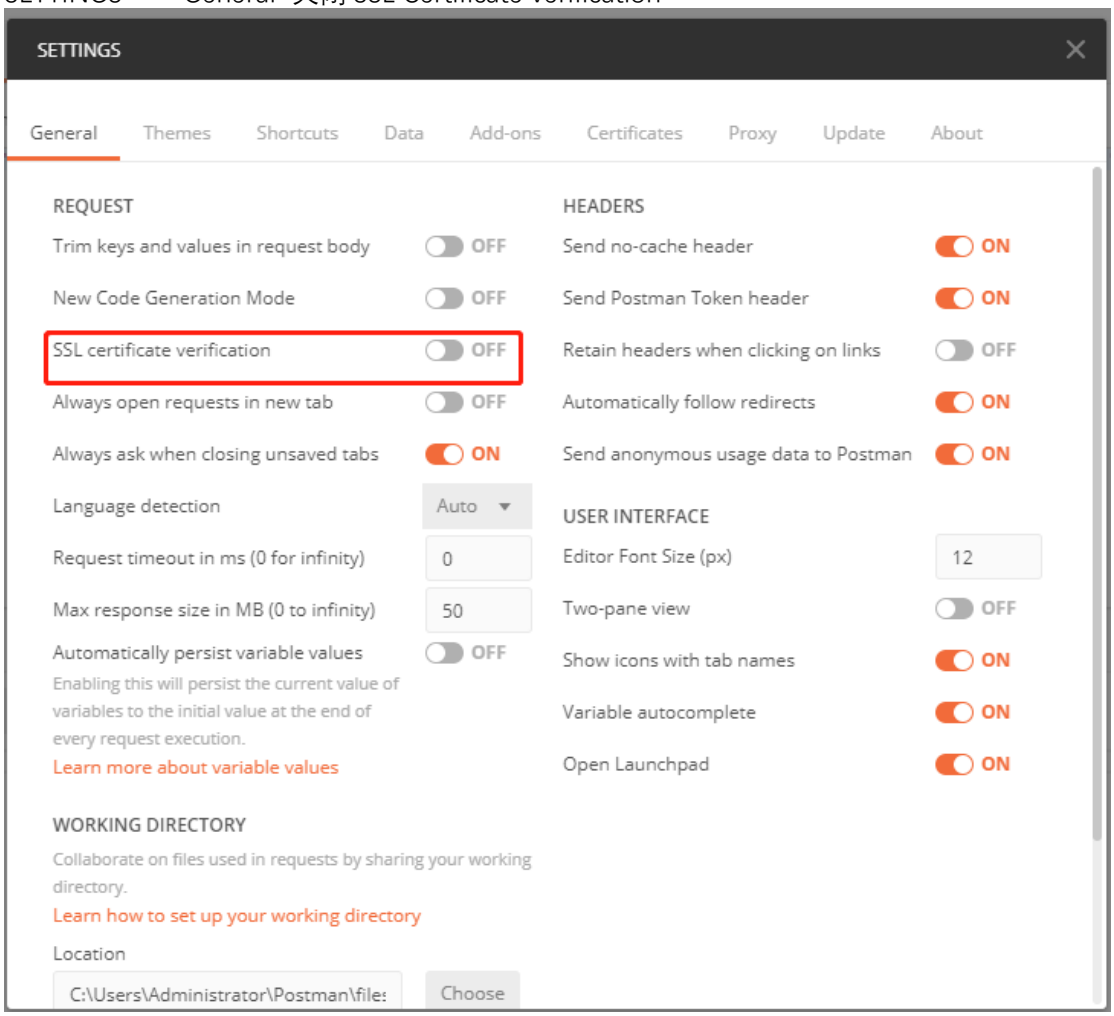
```
        responseInfo = executeRequest(parameterMap);
        Assert.assertEquals(200,responseInfo.getCode());

    }
```

6. Postman 测试实例

设置 postman

SETTINGS--->General 关闭 SSL Certificate verification



1 设置 body 体数据格式

测试数据中的 loginWord 明文是:!QAZ2wsx87127711

2 测试结果