

CỤC CHỨNG THỰC SỐ VÀ BẢO MẬT THÔNG TIN



**TÀI LIỆU HƯỚNG DẪN TÍCH HỢP DỊCH VỤ
KÝ SỐ TẬP TRUNG**

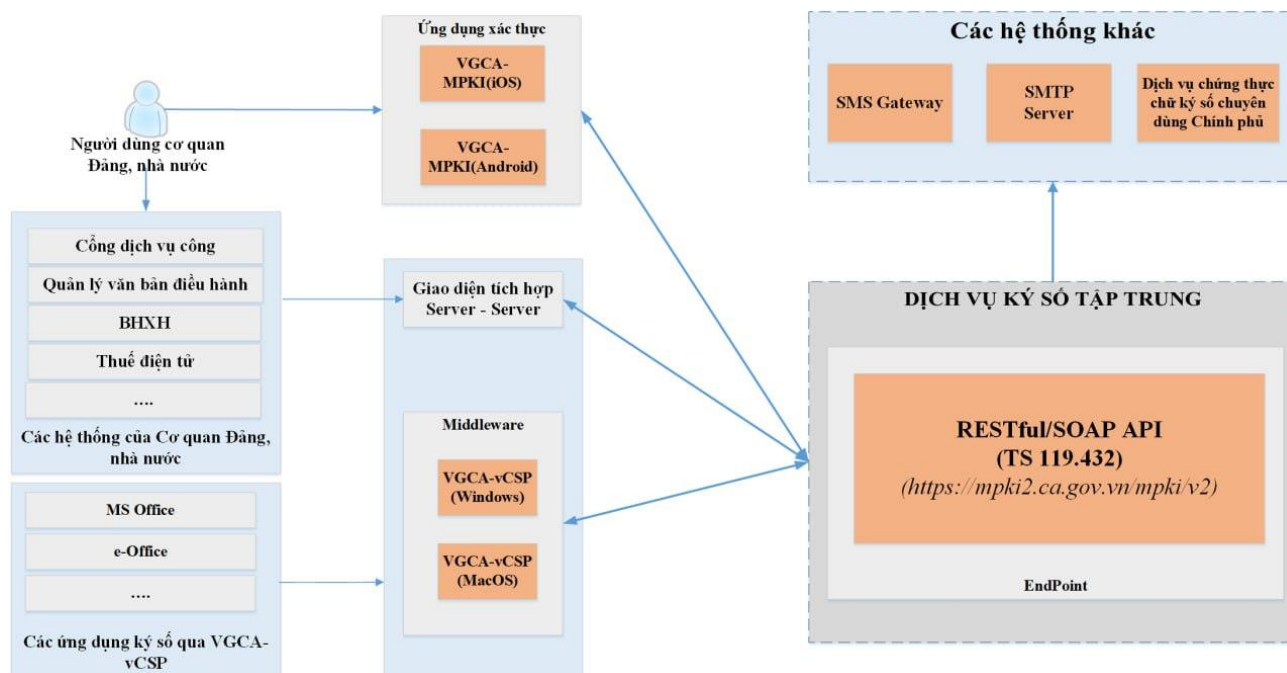
Hà Nội, 2023

MỤC LỤC

1. GIỚI THIỆU	3
2. QUY TRÌNH TÍCH HỢP.....	3
3. MÔ TẢ API.....	4
3.1. Danh sách API	4
3.2. Mô tả luồng thực thi ký số	5
3.3. Tạo kênh kết nối giữa ứng dụng và API webservice của dịch vụ ký số tập trung.....	5
3.4. API auth/login.....	6
3.5. API credentials/list.....	11
3.6. API credentials/info	15
3.7. API credentials/authorize.....	21
3.8. API signatures/signHash.....	26
4. THÔNG TIN LIÊN HỆ HỖ TRỢ	30

1. GIỚI THIỆU

a) Mô hình triển khai dịch vụ ký số tập trung



Hình 1. Mô hình triển khai dịch vụ ký số tập trung

Mô tả: Để thực hiện ký số theo mô hình ký số tập trung, các cơ quan đơn vị có thể triển khai theo hai phương thức sau:

(1) Đối với các cơ quan, đơn vị đã triển khai tích hợp giải pháp ký số sử dụng USB Token thì có thể cài đặt phần mềm vCSP trên hệ điều hành Windows, vCTK trên hệ điều hành MacOS để thực hiện ký số sử dụng dịch vụ ký số tập trung tương tự như ký số với thiết bị USB Token mà không cần phải điều chỉnh, nâng cấp phần mềm ứng dụng để tích hợp lại các API của dịch vụ ký số tập trung.

(2) Ngoài ra các cơ quan, đơn vị có thể triển khai tích hợp trực tiếp các API của dịch vụ ký số tập trung theo mô hình Server to Server.

b) Thông tin dịch vụ ký số tập trung:

- Địa chỉ dịch vụ: <https://mpki2.ca.gov.vn/mpki/v2/>
- Giao thức kết nối: SSL2
- Giao diện kết nối: RESTful/ SOAP API

2. QUY TRÌNH TÍCH HỢP

Bước 1. Đăng ký kết nối dịch vụ ký số tập trung: Cơ quan, đơn vị gửi văn bản yêu cầu triển khai tích hợp dịch vụ ký số tập trung.

Bước 2: Cục Chứng thực số và Bảo mật thông tin cung cấp tài khoản tích hợp, chứng thư số kết nối đến dịch vụ ký số tập trung.

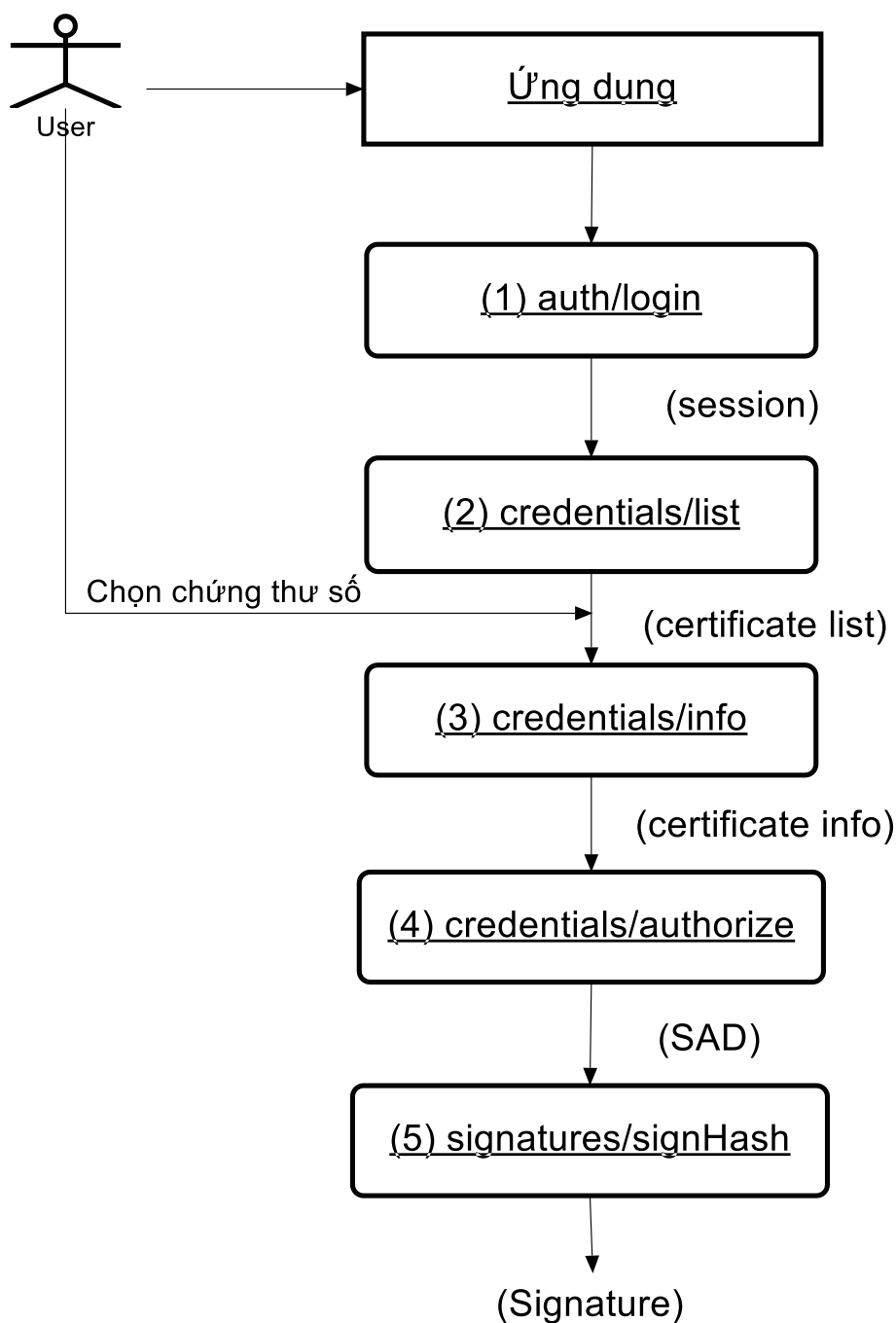
Bước 3: Cơ quan, đơn vị chủ quản ứng dụng công nghệ thông tin sử dụng tài khoản, chứng thư số được cấp để thực hiện kết nối dịch vụ ký số tập trung, tích hợp các API ký số vào trong quy trình nghiệp vụ.

3. MÔ TẢ API

3.1. Danh sách API

STT	API	Mô tả
1	auth/login	Hàm đăng nhập
2	credentials/list	Hàm lấy danh sách chứng thư số của thuê bao, kết quả trả về là thông tin ngắn gọn của các chứng thư số thuộc về thuê bao.
3	credentials/info	Hàm trả về thông tin ủy quyền, thông tin định danh và chứng thư số khóa công khai của thuê bao.
4	credentials/authorize	Chức năng yêu cầu hệ thống xác minh thông tin đăng nhập của thuê bao, hỗ trợ mã OTP, mã PIN (hoặc PassCode).
5	signatures/signHash	Hàm thực hiện ký số dữ liệu băm (hash).

3.2. Mô tả luồng thực thi ký số



Hình 2. Luồng thực thi ký số

3.3. Tạo kênh kết nối giữa ứng dụng và API webservice của dịch vụ ký số tập trung

Tạo SSL2:

Giá trị ssl2 trong header **Auhthorization** được tính:

Ssl2 = BASE64-encode(username:password:signature:timestamp:pkcs1Signature),
trong đó:

username, password, signature nhận về từ **Đăng ký thông tin tích hợp**
timestamp epoch time tính theo milliseconds.

pkcs1Signature chữ ký PKCS#1 được tính bởi khóa riêng trong file .p12 và dữ liệu:

- Mã nguồn mẫu Java:

```
String timestamp = String.valueOf(System.currentTimeMillis());
String data2sign = user + password + signature + timestamp;
String pkcs1Signature =
PKCS1Util.getPKCS1Signature(data2sign, relyingPartyKeyStore,
relyingPartyKeyStorePassword);
String ssl2 = Base64.getEncoder().encodeToString(
    user
        .concat(":")
        .concat(password).concat(":")
        .concat(signature).concat(":")
        .concat(timestamp).concat(":")
        .concat(pkcs1Signature)
        .getBytes());
```

- Mã nguồn mẫu C#:

```
string timestamp = (DateTime.UtcNow - DateTime(1970, 1, 1, 0, 0, 0,
DateTimeKind.Utc)).TotalMilliseconds.ToString();
string data2sign = relyingPartyUser + relyingPartyPassword +
relyingPartySignature + timestamp;
string pkcs1Signature = Utils.getPKCS1Signature(data2sign,
relyingPartyKeyStore, relyingPartyKeyStorePassword);
string SSL2 = relyingPartyUser + ":" + relyingPartyPassword + ":" +
relyingPartySignature + ":" + timestamp + ":" + pkcs1Signature;
string Basic = "USERNAME:" + username + ":" + password;
string BasicEncode = Utils.Base64Encode(Basic);
```

3.4. API auth/login

Ứng dụng gọi HTTP-Restful lên hệ thống để đăng nhập (server to server).

- Tên hàm: auth/login
- Phương thức: Post
- Đường dẫn: <https://mpki2.ca.gov.vn/mpki/v2/auth/login>
- Headers:

STT	Tên	Giá trị
1	Content-Type	application/json
2	Auhthorization	SSL2 {ssl2}

- Tham số đầu vào:

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	relyingParty	String	Bắt buộc	Định danh kênh tích hợp dịch vụ ký số tập trung
2	rpRequestID	String	Tùy chọn	Định danh yêu cầu của kênh
3	requestID	String	Tùy chọn	Định danh yêu cầu
4	rememberMeEnabled	Boolean	Tùy chọn	Tùy chọn để duy trì trạng thái đăng nhập và duy trì xác thực hợp lệ trên nhiều phiên: - “True”: trả về refreshToken để sử dụng trong lần gọi yêu cầu ký tiếp theo - “False”: tham số refreshToken không được trả về Nếu tham số bị bỏ qua, nó sẽ mặc định là "False".
5	clientInfo/iccid	String	Tùy chọn	Thông tin máy người dùng, sử dụng để quản lý phiên làm việc với kênh tích hợp hoặc chủ sở hữu (end-user)
6	clientInfo/imei	String	Tùy chọn	Thông tin máy người dùng, sử dụng để quản lý phiên làm việc với kênh tích hợp hoặc chủ sở hữu (end-user)
7	clientInfo/o-	String	Tùy chọn	Thông tin máy người dùng, sử dụng để quản lý phiên làm việc với kênh tích hợp hoặc chủ sở hữu (end-user)
8	lang	String	Tùy chọn	Ngôn ngữ: EN hoặc VN.

9	profile	String	Bắt buộc	Chuỗi xác định giao thức đang sử dụng bởi ứng dụng máy khách để giao tiếp với SCS. Mặc định là "rssp-119.432-v2.0"
10	tseNotification/notificationMessage	String	Tùy chọn	Tin nhắn (thông điệp) hiển thị dưới dạng thông báo
11	tseNotification/messageCaption	String	Tùy chọn	Tiêu đề tin nhắn
12	tseNotification/message	String	Tùy chọn	Nội dung hiển thị ngoài tên dịch vụ, trước yêu cầu nhập mã PIN xác thực. Kích thước tối đa: 40 bytes
13	tseNotification/logoURI	String	Tùy chọn	URI của logo hiển thị trong quá trình xác thực
14	tseNotification/bgImageURI	String	Tùy chọn	URI của ảnh nền hiển thị trong quá trình xác thực
15	tseNotification/rpIconURI	String	Tùy chọn	URL của icon hiển thị trong quá trình xác thực
16	tseNotification/rpName	String	Tùy chọn	Tên của Kênh tích hợp hiển thị trong quá trình xác thực
27	tseNotification/vcEnabled	boolean	Tùy chọn	Bật VerificationCode với vcEnabled='TRUE' Mặc định là False
18	tseNotification/scaIdentity	String	Tùy chọn	Tên của ứng dụng ký gọi đến RP/vCSP
19	tseNotification/confirmationPolicy	String	Tùy chọn	Chính sách xác nhận: <ul style="list-style-type: none"> ▪ PIN ▪ TAP ▪ SWIPE ▪ BIOMETRIC ▪ IDENTITY
20	tseNotification/validityPeriod	int	Tùy chọn	Thời gian tối đa tính bằng mili giây
21	tseNotification/hashes	String	Tùy chọn	Giá trị băm mã hóa Base64 để tính mã xác minh
22	tseNotification/hashAlgorithmOID	String	Tùy chọn	Thuật toán băm OID sử dụng: <ul style="list-style-type: none"> • "1.3.14.3.2.26": SHA-1. • "2.16.840.1.101.3.4.2.1": SHA-256. • "2.16.840.1.101.3.4.2.2": SHA-384.

				<ul style="list-style-type: none"> • "2.16.840.1.101.3.4.2.3": SHA-512. • "2.16.840.1.101.3.4.2.8": SHA3-256. • "2.16.840.1.101.3.4.2.9": SHA3-384. • "2.16.840.1.101.3.4.2.10": SHA3-512.
--	--	--	--	--

- Tham số đầu ra

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	error	int	Bắt buộc	Mã kết quả Dự kiến: 0
2	errorDescription	String	Bắt buộc	Mô tả chi tiết kết quả
3	responseID	String	Bắt buộc	Định danh giao dịch do hệ thống trả về
4	accessToken	String	Bắt buộc	Mã truy cập dịch vụ tồn tại trong thời gian ngắn được sử dụng để xác thực các yêu cầu API tiếp theo trong cùng một phiên. Khi nhận lệnh gọi API có mã truy cập đã hết hạn, hệ thống sẽ trả về lỗi và yêu cầu xác thực/yêu cầu đăng nhập mới.
5	refreshToken	String	Điều kiện	Mã làm mới tồn tại lâu được sử dụng để xác thực người dùng trong phiên tiếp theo. refreshToken được trả về nếu tham số "rememberMeEnabled"= True
6	expiresIn	Int	Tùy chọn	Thời gian hiệu lực của mã truy cập dịch vụ tính bằng giây. Nếu bỏ qua thời gian mặc định là 3600 giây
7	remainingCounter	Int	Điều kiện	Số lần thử đăng nhập còn lại. Trả về nếu có sẵn trên tiêu đề Authorization và nhập sai mật khẩu
8	tempLockoutDuration	Int	Điều kiện	Thời gian khóa tài khoản, thử lại khi hết thời gian. Tính bằng giây
9	ownerInfo/fullName	String	Điều kiện	Tên đầy đủ của tài khoản

10	ownerInfo/phone	String	Tùy chọn	Số điện thoại của tài khoản
11	ownerInfo/email	String	Tùy chọn	Email của tài khoản
12	ownerInfo/oauth2	String	Tùy chọn	Thông tin OAUTH2
13	authorizeToken	String	Điều kiện	Mã thông báo sử dụng để hoàn tất tiến trình đăng nhập cho thuê bao không nằm trong danh sách tin cậy

Mã nguồn mẫu:

```
[MethodImpl(MethodImplOptions.Synchronized)]
public void login()
{
    Console.WriteLine("_____auth/login_____");
    String authHeader;
    if (refreshToken != null)
    {
        authHeader = refreshToken;
    }
    else
    {
        retryLogin++;
        authHeader = property.getAuthorization(this.username, this.password);
    }
    Console.WriteLine("Login-retry: " + retryLogin);
    LoginRequest loginRequest = new LoginRequest();
    loginRequest.rememberMeEnabled = true;
    loginRequest.relyingParty = property.relyingParty;
    loginRequest.lang = this.lang;
    string jsonReq = JsonConvert.SerializeObject(loginRequest, new
JsonSerializerSettings { NullValueHandling = NullValueHandling.Ignore });
    //, DefaultValueHandling = DefaultValueHandling.Ignore });
    string jsonResp = HTTPUtils.sendPost(property.baseUrl + "auth/login", jsonReq,
authHeader);
    //Console.WriteLine(jsonResp);
    LoginResponse signCloudResp =
JsonConvert.DeserializeObject<LoginResponse>(jsonResp);
    if (signCloudResp.error == 3005 || signCloudResp.error == 3006)
    {
        refreshToken = null;
        if (retryLogin >= 5)
        {
            retryLogin = 0;
            throw new APIException(signCloudResp.error,
signCloudResp.errorDescription);
        }
        login();
    }
    else if (signCloudResp.error != 0)
    {
        throw new APIException(signCloudResp.error, signCloudResp.errorDescription);
    }
    else
    {
        this.bearer = "Bearer " + signCloudResp.accessToken;
        if (signCloudResp.refreshToken != null)
```

```

    {
        this.refreshToken = "Bearer " + signCloudResp.refreshToken;
        Console.WriteLine("Response code: " + signCloudResp.error);
        Console.WriteLine("Response Dessioncription: " +
signCloudResp.errorDescription);
        Console.WriteLine("Response ID: " + signCloudResp.responseID);
        Console.WriteLine("AccessToken: " + signCloudResp.accessToken);
    }
}

```

3.5. API credentials/list

Sử dụng hàm này để lấy danh sách chứng thư số của thuê bao, kết quả trả về là thông tin ngắn gọn của các chứng thư số thuộc về thuê bao.

- Tên hàm: credentials/list
- Phương thức: Post
- Đường dẫn: <https://mpki2.ca.gov.vn/mpki/v2/credentials/list>
- Headers:

STT	Tên	Giá trị
1	Content-Type	application/json
2	Auhtorization	Bearer { accessToken or refresh token }

- Tham số đầu vào:

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	rpRequestID	String	Tùy chọn	Định danh yêu cầu của kênh
2	requestID	String	Tùy chọn	Định danh yêu cầu
3	agreementUUID	String	Điều kiện	Định danh hợp đồng, duy nhất cho mỗi HIS. Tham số này yêu cầu khi access_token được truy xuất từ đăng nhập chỉ với SSL2.
4	searchConditions/ certificateStatus	String	Tùy chọn	Trạng thái chứng thư số: <ul style="list-style-type: none"> • “ALL”, • “GOOD”, • “REVOKED”. Mặc định là “ALL”.
5	searchConditions/ certificatePurpose	String	Tùy chọn	Mục đích sử dụng chứng thư số: <ul style="list-style-type: none"> • “ALL”, • “SIGNATURE”, • “ENCRYPTION”. Mặc định là “ALL”

6	certInfoEnabled	Boolean	Tùy chọn	Trả về thông tin chi tiết của chứng thư số. Điều này hữu ích khi ứng dụng ký muốn lấy một số thông tin chứng thư số mà không giải mã trước. Giá trị mặc định là “False”
7	certificates	String	Tùy chọn	Chỉ định chứng thư số trả về từ chuỗi chứng thư số. • “none”: Không trả về chứng thư số. • “single”: Trả về chứng thư số cuối cùng. • “chain”: trả về toàn bộ chứng thư số. Giá trị mặc định là “single”.
8	authInfoEnabled	Boolean	Tùy chọn	Yêu cầu trả về các tham số khác nhau có chứa thông tin về cơ chế ủy quyền được hỗ trợ bởi thông tin xác thực này (nhóm PIN và OTP). Mặc định là “false”.
9	lang	String	Tùy chọn	Ngôn ngữ: EN hặc VN .
10	profile	String	Bắt buộc	Chuỗi xác định giao thức đang sử dụng bởi ứng dụng máy khách để giao tiếp với SCS. Mặc định là “rssp-119.432-v2.0”

- Tham số đầu ra

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	error	int	Bắt buộc	Mã kết quả Dự kiến: 1007
2	errorDescription	String	Bắt buộc	Mô tả chi tiết kết quả
3	responseID	String	Bắt buộc	Định danh giao dịch do hệ thống trả về
4	certs/issuerDN	String	Điều kiện	Tên tổ chức phát hành chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
5	certs/serialNumber	String	Điều kiện	Số serial chứng thư số, hiển thị định dạng chuỗi mã Hex. Giá

				trị được trả về nếu tham số certInfoEnabled = “true”.
6	certs/credentialID	String	Điều kiện	Định danh liên kết khóa riêng với chứng thư số tương ứng
7	certs/subjectDN	String	Điều kiện	Thông tin nhận dạng thuê bao chứng thư số ở định dạng chuỗi mã UTF-8. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
8	certs/validFrom	String	Điều kiện	Ngày bắt đầu có hiệu lực của chứng thư số ở định dạng chuỗi. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
9	certs/validTo	String	Điều kiện	Ngày hết hiệu lực chứng thư số ở định dạng chuỗi. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
10	certs/certificateProfile/ name	String	Điều kiện	Tên của Profile chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
11	certs/certificateProfile/ description	String	Điều kiện	Mô tả Profile chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
12	certs/purpose	String	Điều kiện	Mục đích sử dụng chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
13	certs/multisign	int	Điều kiện	Số lượng chữ ký được tạo tối đa trên mỗi lần xác thực. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
14	certs/ remainingSigningCounter	int	Bắt buộc	Số lượt ký còn lại của chứng thư số: <ul style="list-style-type: none"> • -1: Không giới hạn. • 0: Hết lượt ký.
15	certs/version	int	Bắt buộc	Phiên bản chứng thư số, ứng dụng thứ ba phải dựa vào giá trị này để quyết định thực hiện credentials/info để cập nhật thông tin chứng thư số

16	certs/certificates	String[]	Điều kiện	<p>Một hoặc nhiều chứng thư số X.509v3 mã hóa Base64 từ chuỗi chứng thư số. Nếu tham số “certificates” = “chain” thì toàn bộ chuỗi chứng thư số PHẢI được trả về cùng với chứng thư số thực thể cuối (end entity certificate) ở đầu mảng</p> <p>Nếu tham số “certificates” = “single” chỉ trả về chứng thư số thực thể cuối (end entity certificate)</p> <p>Nếu tham số “certificates” = “none” không trả về giá trị</p> <p>Mặc định là “none”.</p>
27	certs/authorizationEmail	String	Điều kiện	Email ủy quyền. Giá trị được trả về nếu tham số authInfoEnabled = “true”.
18	certs/authorizationPhone	String	Điều kiện	Số điện thoại ủy quyền. Giá trị được trả về nếu tham số authInfoEnabled = “true”.
19	certs/status	String	Bắt buộc	<p>Trạng thái chứng thư số</p> <ul style="list-style-type: none"> • "not enrolled" • "valid" • "expired" • "revoked" • "suspended" • "unknown"
20	certs/statusDesc	String	Bắt buộc	Mô tả về trạng thái tương ứng với ngôn ngữ.
21	certs/trialEnabled	Boolean	Điều kiện	“True” nếu chứng thư số thử nghiệm, “False” nếu chứng thư số kinh doanh. Nếu thiếu trialEnabled=false

Mã nguồn mẫu:

```
public List<ICertificate> listCertificates(string agreementUUID, string certificate, bool
certInfoEnabled, bool authInfoEnabled, SearchConditions conditions)
{
    Console.WriteLine("_____credentials/list_____");
    String authHeader = bearer;
```

```

CredentialListRequest credentialListRequest = new CredentialListRequest();
credentialListRequest.agreementUUID = agreementUUID;
credentialListRequest.certificates = certificate;
credentialListRequest.certInfoEnabled = certInfoEnabled;
credentialListRequest.authInfoEnabled = authInfoEnabled;
credentialListRequest.searchConditions = conditions;
credentialListRequest.lang = this.lang;

string jsonReq = JsonConvert.SerializeObject(
    credentialListRequest, new JsonSerializerSettings { NullValueHandling =
        NullValueHandling.Ignore });
string jsonResp = HTTPUtils.sendPost(property.baseUrl + "credentials/list",
    jsonReq, authHeader);

CredentialListResponse signCloudResp =
    JsonConvert.DeserializeObject<CredentialListResponse>(jsonResp);
if (signCloudResp.error == 3005 || signCloudResp.error == 3006)
{
    login();
    return listCertificates(agreementUUID, certificate, certInfoEnabled,
        authInfoEnabled, conditions);
}
else if (signCloudResp.error != 0)
{
    throw new APIException(signCloudResp.error, signCloudResp.errorDescription);
}
List<BaseCertificateInfo> listCert = signCloudResp.certs;
List<ICertificate> listCertificate = new List<ICertificate>();

foreach (var item in listCert)
{
    ICertificate icrt = new Certificate(item, agreementUUID, this);
    listCertificate.Add(icrt);
}

Console.WriteLine("Error code: " + signCloudResp.error);
Console.WriteLine("Error description: " + signCloudResp.errorDescription);
return listCertificate;
}

```

3.6. API credentials/info

Hàm này sẽ trả về thông tin ủy quyền, thông tin định danh và chứng thư số khóa công khai của thuê bao.

- Tên hàm: credentials/info
- Phương thức: Post
- Đường dẫn: <https://mpki2.ca.gov.vn/mpki/v2/credentials/info>
- Headers:

STT	Tên	Giá trị
1	Content-Type	application/json
2	Auhthorization	Bearer {accessToken or refresh token}

- Tham số đầu vào:

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	rpRequestID	String	Tùy chọn	Định danh yêu cầu của kênh
2	requestID	String	Tùy chọn	Định danh yêu cầu
3	lang	String	Tùy chọn	Ngôn ngữ: EN hặc VN.
4	agreementUUID	String	Điều kiện	Định danh hợp đồng, duy nhất cho mỗi HIS. Tham số này yêu cầu khi access_token được truy xuất từ đăng nhập chỉ với SSL2.
5	credentialID	String	Bắt buộc	Định danh liên kết khóa riêng với chứng thư số tương ứng
6	certificates	String	Tùy chọn	Chỉ định chứng thư số trả về từ chuỗi chứng thư số. <ul style="list-style-type: none"> • “none”: Không trả về chứng thư số. • “single”: Trả về chứng thư số cuối cùng. • “chain”: trả về toàn bộ chứng thư số. Giá trị mặc định là “single”.
7	certInfoEnabled	Boolean	Tùy chọn	Trả về thông tin chi tiết của chứng thư số. Điều này hữu ích khi ứng dụng ký muốn lấy một số thông tin chứng thư số mà không giải mã trước. Giá trị mặc định là “False”
8	authInfoEnabled	Boolean	Tùy chọn	Yêu cầu trả về các tham số khác nhau có chứa thông tin về cơ chế ủy quyền được hỗ trợ bởi thông tin xác thực này (nhóm PIN và OTP). Mặc định là “false”.
9	profile	String	Bắt buộc	Chuỗi xác định giao thức đang sử dụng bởi ứng dụng máy khách để giao tiếp với SCS. Mặc định là “rssp-119.432-v2.0”

- Tham số đầu ra

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
----	---------	--------------	----------	-------

1	error	String	Tùy chọn	Mã kết quả
2	errorDescription	String	Điều kiện	Mô tả chi tiết kết quả
3	responseID	String	Điều kiện	Định danh cho mỗi giao dịch
4	cert/status	String	Bắt buộc	Trạng thái chứng thư số ký: <ul style="list-style-type: none"> ▪ “not enrolled” ▪ “initialized” ▪ “generated” ▪ “valid” ▪ “renewed” ▪ “revised” ▪ “revoked” ▪ “expired” ▪ “declined” ▪ “suspended”
5	cert/statusDesc	String	Điều kiện	Mô tả trạng thái với ngôn ngữ tương ứng
6	cert/certificates	String	Tùy chọn	Một hoặc nhiều chứng thư số X.509v3 mã hóa Base64 từ chuỗi chứng thư số. Nếu tham số “certificates” = “chain” thì toàn bộ chuỗi chứng thư số PHẢI được trả về cùng với chứng thư số thực thể cuối (end entity certificate) ở đầu mảng Nếu tham số “certificates” = “single” chỉ trả về chứng thư số thực thể cuối (end entity certificate) Nếu tham số “certificates” = “none” không trả về giá trị
7	csr	int	Bắt buộc	Nếu cert/certificates không có sẵn thì máy chủ sẽ trả về csr để thay thế.
8	cert/issuerDN	String[]	Điều kiện	Tên tổ chức phát hành chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled = “true”.
9	cert/serialNumber	String	Điều kiện	Số serial chứng thư số, hiển thị định dạng chuỗi mã Hex. Giá

				trị được trả về nếu tham số certInfoEnabled =“true”.
10	cert/thumbprint	String	Tùy chọn	Chuỗi băm tạo từ khóa công khai của chứng thư số. Giá trị này được tính bằng cách băm nhị phân chứng thư số đã mã hóa, thường sử dụng thuật toán SHA-256
11	cert/subjectDN	String	Tùy chọn	Thông tin nhận dạng thuê bao chứng thư số ở định dạng chuỗi mã UTF-8. Giá trị được trả về nếu tham số certInfoEnabled =“true”.
12	cert/validFrom	String	Tùy chọn	Ngày bắt đầu có hiệu lực của chứng thư số ở định dạng chuỗi. Giá trị được trả về nếu tham số certInfoEnabled =“true”.
13	cert/validTo	String	Tùy chọn	Ngày hết hiệu lực chứng thư số ở định dạng chuỗi. Giá trị được trả về nếu tham số certInfoEnabled =“true”.
14	cert/certificateProfile/type	String	Tùy chọn	Loại profile của chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled =“true”.
15	cert/certificateProfile/name	String	Tùy chọn	Tên profile của chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled =“true”.
16	cert/certificateProfile/description	String	Tùy chọn	Mô tả profile của chứng thư số. Giá trị được trả về nếu tham số certInfoEnabled =“true”.
17	cert/purpose	String	Tùy chọn	Mục đích sử dụng chứng thư số
18	cert/version	String	Tùy chọn	Phiên bản chứng thư số, ứng dụng thứ ba phải dựa vào giá trị này để quyết định cập nhật thông tin chứng thư số mà nó lưu trữ.
19	sharedMode	String	Tùy chọn	Chế độ chia sẻ được sử dụng trong ký số tập trung: ▪ “PRIVATE_MODE”

				<ul style="list-style-type: none"> ▪ “RP_SHARED_MODE” ▪ “AGREEMENT_SHARED_MODE” Mặc định là PRIVATE_MODE.
20	createdRP	String	Tùy chọn	Tên kênh kết nối dịch vụ ký số tập trung, chỉ ra chứng thư số được tạo ra bởi kênh này.
21	multisign	Boolean	Tùy chọn	Số lượng chữ ký được tạo tối đa trên mỗi lần xác thực. Mặc định là 1.
22	authModes	Boolean	Tùy chọn	<p>Chỉ định một trong các chế độ ủy quyền.</p> <ul style="list-style-type: none"> ▪ “EXPLICIT/PIN”: mã passphrase (PIN). ▪ “EXPLICIT/OTP-SMS”: OTP SMS. ▪ “EXPLICIT/OTP-EMAIL”: OTP Email. ▪ “IMPLICIT/TSE”: ủy quyền được quản lý bởi dịch vụ một cách tự động ▪ “IMPLICIT/BIP-CATTP”: ủy quyền được quản lý bởi Cyber-ID – một ứng dụng di động. ▪ “EXPLICIT/OTP-MOBILE”: OTP Mobile Application. ▪ “OAUTH2”: chưa triển khai
23	authMode	String	Tùy chọn	Một giá trị trong các giá trị của No.21, là chế độ ủy quyền được kích hoạt tại thời điểm này. Để thay đổi phải được thực hiện bởi quản trị viên hệ thống.
24	SCAL	String	Tùy chọn	<ul style="list-style-type: none"> ▪ 1: Dữ liệu băm to-be-sign không liên kết với dữ liệu kích hoạt chữ ký

				<ul style="list-style-type: none"> 2: Dữ liệu băm to-be-sign được liên kết với dữ liệu kích hoạt chữ ký
25	contractExpirationDate	String	Tùy chọn	Ngày hết hạn hợp đồng, để cập lại chứng thư số
26	remainingSigningCounter	String	Tùy chọn	Số lượt ký còn lại của chứng thư số: <ul style="list-style-type: none"> -1: Không giới hạn. 0: Hết lượt ký.
27	authorizationEmail	Int	Tùy chọn	Email ủy quyền. Giá trị được trả về nếu tham số authInfoEnabled = "true".
28	authorizationPhone	String	Bắt buộc	Số điện thoại ủy quyền. Giá trị được trả về nếu tham số authInfoEnabled = "true".
29	defaultPassphraseEnabled	String[]	Điều kiện	Đúng nếu cụm mật khẩu là mặc định và người dùng phải thay đổi cụm mật khẩu trước khi thực hiện mã hóa
30	trialEnabled	String	Tùy chọn	Đúng nếu chứng thư số là thử nghiệm, sai nếu chứng thư số là kinh doanh. Nếu thiếu nó có nghĩa là sai.
31	cert/certificateAuthority/name	String	Tùy chọn	Tên cơ quan phát hành chứng thư
32	cert/certificateAuthority/description		Tùy chọn	Mô tả thẩm quyền của cơ quan phát hành chứng thư

Mã nguồn mẫu:

```

public ICertificate certificateInfo(String agreementUUID, String credentialID, String
certificate, bool certInfoEnabled, bool authInfoEnabled)
{
    Console.WriteLine("_____credentials/info_____");
    CredentialInfoRequest credentiallistRequest = new CredentialInfoRequest();
    credentiallistRequest.agreementUUID = agreementUUID;
    credentiallistRequest.credentialID = credentialID;
    credentiallistRequest.certificates = certificate;
    credentiallistRequest.certInfoEnabled = certInfoEnabled;
    credentiallistRequest.authInfoEnabled = authInfoEnabled;
    credentiallistRequest.lang = this.lang;
    string jsonReq = JsonConvert.SerializeObject(
        credentiallistRequest, new JsonSerializerSettings { NullValueHandling =
        NullValueHandling.Ignore });
    string jsonResp = HTTPUtils.sendPost(property.baseUrl + "credentials/info",
    jsonReq, bearer);
    //Console.WriteLine(jsonResp);
}

```

```

        CredentialInfoResponse signCloudResp =
JsonConvert.DeserializeObject<CredentialInfoResponse>(jsonResp);
        if (signCloudResp.error == 3005 || signCloudResp.error == 3006)
        {
            login();
            return certificateInfo(agreementUUID, credentialID, certificate,
certInfoEnabled, authInfoEnabled);
        }
        else if (signCloudResp.error != 0)
        {
            throw new APIException(signCloudResp.error, signCloudResp.errorDescription);
        }

        ICertificate icrt = new Certificate(signCloudResp.cert, agreementUUID, this);
        signCloudResp.cert.authorizationEmail = signCloudResp.authorizationEmail;
        signCloudResp.cert.authorizationPhone = signCloudResp.authorizationPhone;
        signCloudResp.cert.sharedMode = signCloudResp.sharedMode;
        signCloudResp.cert.createdRP = signCloudResp.createdRP;
        signCloudResp.cert.authModes = signCloudResp.authModes;

        signCloudResp.cert.authMode = signCloudResp.authMode;
        signCloudResp.cert.SCAL = signCloudResp.SCAL;
        signCloudResp.cert.contractExpirationDate =
signCloudResp.contractExpirationDate;
        signCloudResp.cert.defaultPassphraseEnabled =
signCloudResp.defaultPassphraseEnabled;
        signCloudResp.cert.trialEnabled = signCloudResp.trialEnabled;

        return icrt;
    }

```

3.7. API credentials/authorize

Chức năng yêu cầu hệ thống xác minh thông tin đăng nhập của thuê bao, hỗ trợ mã OTP, mã PIN (hoặc PassCode).

- Tên hàm: credentials/authorize
- Phương thức: Post
- Đường dẫn: <https://mpki2.ca.gov.vn/mpki/v2/credentials/authorize>
- Headers:

STT	Tên	Giá trị
1	Content-Type	application/json
2	Auhtorization	Bearer {accessToken or refresh token}

- Tham số đầu vào:

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	rpRequestID	String	Tùy chọn	Định danh yêu cầu của kênh
2	requestID	String	Điều kiện	Định danh yêu cầu. Giá trị này là bắt buộc nếu authorizeCode là OTP và giá trị của nó là

				responseID từ credentials/sendOTP
3	agreementUUID	String	Điều kiện	Định danh thỏa thuận, duy nhất cho mỗi máy chủ sử dụng dịch vụ. Tham số này yêu cầu khi access_token được truy xuất từ đăng nhập chỉ với SSL2.
4	credentialID	String	Bắt buộc	Định danh liên kết khóa riêng với chứng thư số tương ứng
5	authorizeCode	String	Điều kiện	Mã ủy quyền do người dùng cung cấp. Có thể là mã PIN (PassCode) hoặc OTP. Bắt buộc khi authMode trả về từ API credentials/info là EXPLICIT/PIN hoặc EXPLICIT/OTP-SMS hoặc EXPLICIT/OTP-EMAIL hoặc EXPLICIT/OTP-MOBILE
6	lang	String	Tùy chọn	Ngôn ngữ: EN hoặc VN .
7	numSignatures	int	Bắt buộc	Số lượng chữ ký để ủy quyền. Các giao dịch đa chữ ký có thể tạo bằng cách sử dụng kết hợp việc truyền một mảng các giá trị băm và gọi phương thức signatures/signHash
8	documentDigests/hashes	String[]	Điều kiện	Một hoặc nhiều giá trị băm được mã Base64 để ký. Bắt buộc khi SCAL là cấp 2
9	documentDigests/ hashAlgorithmOID	String	Điều kiện	Thuật toán băm OID sử dụng: <ul style="list-style-type: none"> ▪ "1.3.14.3.2.26": SHA-1. ▪ "2.16.840.1.101.3.4.2.1": SHA-256. ▪ "2.16.840.1.101.3.4.2.2": SHA-384. ▪ "2.16.840.1.101.3.4.2.3": SHA-512. ▪ "2.16.840.1.101.3.4.2.8": SHA3-256. ▪ "2.16.840.1.101.3.4.2.9": SHA3-384.

				<ul style="list-style-type: none"> ▪ "2.16.840.1.101.3.4.2.10": SHA3-512.
10	clientInfo/iccid	String	Tùy chọn	Thông tin máy khách. Kênh tích hợp sử dụng để quản lý ràng buộc phiên làm việc với một thỏa thuận
11	clientInfo/imei	String	Tùy chọn	Thông tin máy khách. Kênh tích hợp sử dụng để quản lý ràng buộc phiên làm việc với một thỏa thuận
12	clientInfo/macAddr	String	Tùy chọn	Thông tin máy khách. Kênh tích hợp sử dụng để quản lý ràng buộc phiên làm việc với một thỏa thuận
13	notificationMessage	String	Tùy chọn	Tin nhắn hiển thị dưới dạng thông báo
14	messageCaption	String	Tùy chọn	Tiêu đề tin nhắn
15	message	String	Tùy chọn	Text displayed in addition to Service Name and before asking authentication PIN.
16	logoURI	String	Tùy chọn	URI của logo tùy chỉnh được hiển thị trong trình xác thực
27	bgImageURI	String	Tùy chọn	URI của ảnh nền được hiển thị trong trình xác thực
18	rpIconURI	String	Tùy chọn	URI của icon RP được hiển thị trong trình xác thực
19	rpName	String	Tùy chọn	Tên tùy chỉnh của RP hiển thị trong quá trình xác thực
20	confirmationPolicy	String	Tùy chọn	Chính sách xác nhận: <ul style="list-style-type: none"> ▪ PIN ▪ TAP ▪ SWIPE ▪ BIOMETRIC ▪ IDENTITY
21	vcEnabled	Boolean	Tùy chọn	Bật VerificationCode với vcEnabled='TRUE' Mặc định là 'TRUE'
22	acEnabled	Boolean	Tùy chọn	Bật AuthorizeCode với acEnabled='TRUE'
23	operationMode	String	Tùy chọn	Chế độ tin nhắn sử dụng:

				<ul style="list-style-type: none"> • "A": Chế độ hoạt động không đồng bộ. • "S": chế độ hoạt động đồng bộ (mặc định).
24	scaIdentity	String	Tùy chọn	Tên ứng dụng ký, gọi đến RP/vCSR
25	clientInfo/instanceUUID	String	Tùy chọn	Phiên được xác định giữa ứng dụng tạo chữ ký với máy khách (RP hoặc VCSP).
26	responseURI	String	Tùy chọn	Trường thông tin có giá trị dưới dạng URI để thông báo đường dẫn kết quả sau khi SCS thực hiện xong
27	validityPeriod	Int	Tùy chọn	Khoảng thời gian tối đa của yêu cầu tính theo giây
28	profile	String	Bắt buộc	Chuỗi xác định giao thức đang được ứng dụng khách sử dụng để giao tiếp với SCS. Mặc định là rssp-119.432-v2.0.
29	documents	String[]	Điều kiện	Dữ liệu được mã hóa Base64 để ký. Tham số này không được sử dụng nếu documentDigests được thông qua.
30	signAlgo	String	Tùy chọn	Thuật toán OID được sử dụng để ký: <ul style="list-style-type: none"> • "1.2.840.113549.1.1.1": RSA. • "1.2.840.113549.1.1.5": sha1RSA. • "1.2.840.113549.1.1.11": sha256RSA. • "1.2.840.113549.1.1.12": sha384RSA. • "1.2.840.113549.1.1.13": sha512RSA. • "2.16.840.1.101.3.4.3.14": sha3-256RSA. • "2.16.840.1.101.3.4.3.15": sha3-384RSA.

				<ul style="list-style-type: none"> "2.16.840.1.101.3.4.3.16": sha3-512RSA Bắt buộc khi SCAL ở mức 2 và tài liệu sẵn sàng
31	signAlgoParams	String	Tùy chọn	Lược đồ ký số định dạng ASN.1 Der được mã hóa Base64. Ví dụ lược đồ RSASSA-PSS= "BgkqhkiG9w0BAQo=".

- Tham số đầu ra

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	error	int	Bắt buộc	Mã kết quả Dự kiến: 0
2	errorDescription	String	Bắt buộc	Mô tả chi tiết kết quả
3	responseID	String	Bắt buộc	Định danh cho mỗi giao dịch
4	SAD	String	Điều kiện	Dữ liệu kích hoạt chữ ký (SAD) được sử dụng làm đầu vào cho phương thức signatures/signHash, sẽ bỏ lỡ nếu xảy ra lỗi trong quá trình xử lý thông tin ủy quyền, khi đó remainingCounter là bắt buộc
5	expiresIn	int	Tùy chọn	Thời gian hiệu lực của SAD tính bằng giây. Nếu bỏ qua thời gian mặc định là 3600 giây
6	remainingCounter	int	Điều kiện	Bộ đếm giảm 1 khi mã ủy quyền không hợp lệ. Chứng thư bị chặn khi bộ đếm =0 và bỏ chặn sau 300s Giá trị mặc định là 300 giây, có thể được cấu hình lại trong hệ thống
7	tempLockoutDuration	Int	Điều kiện	Thời gian khóa chứng thư (tính theo giây), thử lại khi hết thời gian
8	documentDigests/hashes	String[]	Điều kiện	Một hoặc nhiều giá trị băm được mã hóa Base64 để ký. Nó

				được trả về khi SCAL ở mức 2 và tài liệu có sẵn theo yêu cầu.
9	documentDigests/ hashAlgorithmOID	String	Điều kiện	Thuật toán băm OID được sử dụng để tính toán (các) giá trị băm của tài liệu. Nó được trả về khi SCAL ở mức 2 và tài liệu có sẵn theo yêu cầu.

Mã nguồn mẫu:

```
public string authorize(string agreementUUID, string credentialID, int numSignatures,
DocumentDigests doc, SignAlgo? signAlgo, string otpRequestID, string passCode)
{
    Console.WriteLine("_____credentials/authorize_____");
    AuthorizeRequest request = new AuthorizeRequest();
    request.agreementUUID = agreementUUID;
    request.credentialID = credentialID;
    request.numSignatures = numSignatures;
    request.documentDigests = doc;
    request.signAlgo = signAlgo;
    request.requestID = otpRequestID;
    request.authorizeCode = passCode;
    request.lang = this.lang;

    string jsonReq = JsonConvert.SerializeObject(
        request, new JsonSerializerSettings { NullValueHandling =
NullValueHandling.Ignore });
    string jsonResp = HTTPUtils.sendPost(property.baseUrl + "credentials/authorize",
jsonReq, bearer);
    //Console.WriteLine(jsonResp);
    AuthorizeResponse signCloudResp =
JsonConvert.DeserializeObject<AuthorizeResponse>(jsonResp);
    if (signCloudResp.error == 3005 || signCloudResp.error == 3006)
    {
        login();
        return authorize(agreementUUID, credentialID, numSignatures, doc, signAlgo,
otpRequestID, passCode);
    }
    else if (signCloudResp.error != 0)
    {
        throw new APIException(signCloudResp.error, signCloudResp.errorDescription);
    }
    return signCloudResp.SAD;
}
```

3.8. API signatures/signHash

Chức năng thực hiện ký số dữ liệu băm (hash).

- Tên hàm: signatures/signHash
- Phương thức: Post
- Đường dẫn: <https://mpki2.ca.gov.vn/mpki/v2/signatures/signHash>
- Headers:

STT	Tên	Giá trị
1	Content-Type	application/json
2	Auhtorization	Bearer {accessToken or refresh token}

- Tham số đầu vào:

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	rpRequestID	String	Tùy chọn	Định danh yêu cầu của kênh
2	requestID	String	Tùy chọn	Định danh yêu cầu
3	lang	String	Tùy chọn	Ngôn ngữ: EN hặc VN.
4	agreementUUID	String	Điều kiện	Định danh hợp đồng, duy nhất cho mỗi HIS. Tham số này yêu cầu khi access_token được truy xuất từ đăng nhập chỉ với SSL2.
5	credentialID	String	Bắt buộc	Định danh liên kết khóa riêng với chứng thư số tương ứng
6	SAD	String	Bắt buộc	Dữ liệu Kích hoạt Chữ ký được trả về bởi các phương thức Ủy quyền Thông tin xác thực.
7	documentDigests/ hashes	String[]	Điều kiện	Một hoặc nhiều giá trị băm mã hóa Base64 để ký
8	documentDigests/ hashAlgorithmOID	String	Điều kiện	Thuật toán băm OID sử dụng: <ul style="list-style-type: none"> "1.3.14.3.2.26": SHA-1. "2.16.840.1.101.3.4.2.1": SHA-256. "2.16.840.1.101.3.4.2.2": SHA-384. "2.16.840.1.101.3.4.2.3": SHA-512. "2.16.840.1.101.3.4.2.8": SHA3-256. "2.16.840.1.101.3.4.2.9": SHA3-384. "2.16.840.1.101.3.4.2.10": SHA3-512.
9	signAlgo	String	Bắt buộc	Thuật toán OID được sử dụng để ký: <ul style="list-style-type: none"> "1.2.840.113549.1.1.1": RSA.

				<ul style="list-style-type: none"> ▪ "1.2.840.113549.1.1.5": sha1RSA. ▪ "1.2.840.113549.1.1.11": sha256RSA. ▪ "1.2.840.113549.1.1.12": sha384RSA. ▪ "1.2.840.113549.1.1.13": sha512RSA. ▪ "2.16.840.1.101.3.4.3.14": sha3-256RSA. ▪ "2.16.840.1.101.3.4.3.15": sha3-384RSA. ▪ "2.16.840.1.101.3.4.3.16": sha3-512RSA ▪ "1.2.840.10045.4": ECDSA. ▪ "1.2.840.10045.4.3.1": sha1ECDSA. ▪ "1.2.840.10045.4.3.2": sha256ECDSA. ▪ "1.2.840.10045.4.3.3": sha384ECDSA. ▪ "1.2.840.10045.4.3.4": sha512ECDSA.
10	signAlgoParams	String	Điều kiện	Lược đồ ký số định dạng ASN.1 Der được mã hóa Base64. Ví dụ lược đồ RSASSA-PSS="BgkqhkiG9w0BAQo=".
11	operationMode	String	Tùy chọn	Chế độ tin nhắn sử dụng: <ul style="list-style-type: none"> • "A": Chế độ hoạt động không đồng bộ. • "S": chế độ hoạt động đồng bộ (mặc định).
12	scaIdentity	String	Tùy chọn	Tên của ứng dụng ký gọi đến RP/vCSP
13	clientInfo/instanceUUID	String	Tùy chọn	Phiên được xác định giữa ứng dụng tạo chữ ký với máy khách (RP hoặc VCSP).
14	responseURI	String	Tùy chọn	Trường thông tin có giá trị dưới dạng URI để thông báo

				đường dẫn kết quả sau khi SCS thực hiện xong
15	validityPeriod	Int	Tùy chọn	Khoảng thời gian tối đa tính bằng mili giây ở chế độ "A"
16	profile	String	Bắt buộc	Khoảng thời gian tối đa được biểu thị bằng mili giây ở chế độ "A"

- Tham số đầu ra

TT	Tham số	Kiểu dữ liệu	Bắt buộc	Mô tả
1	error	int	Bắt buộc	Mã kết quả Dự kiến: 0
2	errorDescription	String	Bắt buộc	Mô tả chi tiết kết quả
3	responseID	String	Bắt buộc	Định danh cho mỗi giao dịch
4	signatures	String[]	Tùy chọn	Một hoặc nhiều giá trị băm có chữ ký được mã hóa Base64. Trường hợp có nhiều chữ ký, giá trị băm trả về theo thứ tự các hàm tương ứng.
5	remainingSigningCounter	int	Tùy chọn	Số lượt ký còn lại của chứng thư số: <ul style="list-style-type: none"> -1: Không giới hạn. 0: Hết lượt ký.
6	remainingCounter	int	Điều kiện	Bộ đếm giảm 1 khi mã ủy quyền không hợp lệ. Chứng thư bị chặn khi bộ đếm =0 và bỏ chặn sau 300s Giá trị mặc định là 300 giây, có thể cấu hình lại trong hệ thống
7	tempLockoutDuration	Int	Điều kiện	Thời gian khóa chứng thư (tính theo giây), thử lại khi hết thời gian

Mã nguồn mẫu:

```

public List<byte[]> signHash(string agreementUUID, string credentialID,
DocumentDigests documentDigest, SignAlgo? signAlgo, string SAD)
{
    Console.WriteLine("_____signatures/signHash_____");
    SignHashRequest request = new SignHashRequest();
    request.agreementUUID = agreementUUID;
    request.credentialID = credentialID;
    request.documentDigests = documentDigest;
    request.signAlgo = signAlgo;
    request.SAD = SAD;
    request.lang = this.lang;
    //Console.WriteLine(request);

```

```

        string jsonReq = JsonConvert.SerializeObject(
            request, new JsonSerializerSettings { NullValueHandling =
NullValueHandling.Ignore });
        string jsonResp = HTTPUtils.sendPost(property.baseUrl + "signatures/signHash",
jsonReq, bearer);
        //Console.WriteLine(jsonResp);

        SignHashResponse signCloudResp =
JsonConvert.DeserializeObject<SignHashResponse>(jsonResp);
        if (signCloudResp.error == 3005 || signCloudResp.error == 3006)
        {
            login();
            return signHash(agreementUUID, credentialID, documentDigest, signAlgo, SAD);
        }
        else if (signCloudResp.error != 0)
        {
            throw new APIException(signCloudResp.error, signCloudResp.errorDescription);
        }
        Console.WriteLine("err code: " + signCloudResp.error);
        Console.WriteLine("error description: " + signCloudResp.errorDescription);
        return signCloudResp.signatures;
    }

```

4. THÔNG TIN LIÊN HỆ HỖ TRỢ

Cục Chứng thực số và Bảo mật thông tin

Địa chỉ: Số 23, Ngõ Như Kôn Tum, Thanh Xuân, Hà Nội

Điện thoại: 0243.773.8668

Email: ca@bcy.gov.vn

Website: <https://ca.gov.vn>