



TẬP ĐOÀN CÔNG NGHIỆP VIỄN THÔNG QUÂN ĐỘI

**TÀI LIỆU TÍCH HỢP DỊCH VỤ CHỮ KÝ SỐ
MYSIGN VIETTEL**

Mã hiệu dự án: Mysign

Phiên bản: 1.8

Hà Nội, 02/2025

Mục lục

1. GIỚI THIỆU:	4
1.1. Mục đích và ý nghĩa của Tài liệu	4
1.2. Phạm vi tài liệu	4
1.3. Thuật ngữ và các từ viết tắt	4
1.4. Cấu trúc tài liệu	4
2. GIỚI THIỆU CHUNG:	5
2.1. Giới thiệu chung về dịch vụ Mysign:	5
2.2. Luồng thực hiện ký số của người dùng sử dụng ứng dụng Mysign:	5
2.3. Luồng thực hiện ký số của người dùng sử dụng ứng dụng bên thứ ba:	6
3. CHI TIẾT LUỒNG API DỊCH VỤ MYSIGN:	7
3.1. Luồng API đăng ký thiết bị làm thiết bị xác thực	7
3.2. Luồng API gửi yêu cầu ký số trên hệ thống ứng dụng (đồng bộ)	9
3.3. Luồng API gửi yêu cầu ký số trên hệ thống ứng dụng (bất đồng bộ)	10
3.4. Luồng API xác thực yêu cầu ký trên App mobile	11
3.5. Luồng API ký bất đồng bộ có callback	13
Phụ lục 1: API gửi yêu cầu ký tới dịch vụ Mysign	14
1. Đặc tả giao tiếp ký đồng bộ	14
1.1. Các loại giao dịch	14
1.2. Đặc tả chi tiết các giao dịch	14
2. Đặc tả giao tiếp ký bất đồng bộ	26
2.1. Các loại giao dịch	26
2.2. Đặc tả chi tiết các giao dịch	27
3. Đặc tả API ký bất đồng bộ có callback (async = 3)	44
4. Danh sách mã lỗi chung	49
Phụ lục 2: API Cloud-CA dành cho mobile app xác thực ký	61
1. Xác thực Client	61
2. Xác thực User	62
3. Xác thực OTP	64
4. Renew Access Token	67
5. Danh sách thiết bị đã được đăng ký (List Registered Device)	68
6. Xóa thiết bị khỏi danh sách đăng ký	70
7. Lấy danh sách yêu cầu xác thực đang chờ (Get Pending Authorization Request)	71
8. Lấy yêu cầu xác thực đang chờ sử dụng transaction_id (Get Pending Authorization Request)	73
9. Hủy bỏ yêu cầu xác thực (Cancel a Pending Authorisation Request)	76
10. Thông tin tài khoản (Users Profile)	77
11. Đăng ký thiết bị để nhận notification	78
12. Xóa thiết bị khỏi danh sách nhận notification	79
Phụ lục 3: Hướng dẫn tích hợp CloudCA SDK Lite trên Android	81
1. Hướng dẫn cài đặt	81

2.	Cấu hình thêm vào class Application	85
3.	Hướng dẫn sử dụng API	86
3.1.	<i>BaseModel</i>	86
3.2.	<i>Đăng ký thiết bị</i>	88
3.3.	<i>Xác thực yêu cầu ký</i>	90
Phụ lục 4: Hướng dẫn tích hợp CloudCA SDK Lite trên iOS		92
1.	Thêm thư viện CloudCA SDK	92
2.	Thực hiện import	93
3.	Thực hiện cài đặt Base URL, User ID cho SDK	93
4.	Thực hiện lấy Device ID, CSR	94
5.	Thực hiện cài đặt API	95
5.1.	<i>API đăng ký thiết bị</i>	95
5.2.	<i>API Xác thực yêu cầu ký</i>	97
Phụ lục 5: Thông tin code demo, SDK Mysign		100

1. GIỚI THIỆU:

1.1. Mục đích và ý nghĩa của Tài liệu

Tài liệu này dùng để hướng dẫn các hệ thống, cũng như ứng dụng bên thứ 3 tích hợp sử dụng dịch vụ chữ ký số Mysign – Viettel. Trong đó, tài liệu hướng dẫn tích hợp cho các đối tượng sau:

- Tích hợp dịch vụ ký số sử dụng ứng dụng Mysign để ký
- Tích hợp dịch vụ ký số, trong đó sử dụng ứng dụng di động bên thứ 3 đã tích hợp SDK Mobile để ký

1.2. Phạm vi tài liệu

Tài liệu làm rõ các thông tin về các API gửi yêu cầu ký và các API, hàm có sẵn trong SDK Mobile; làm rõ luồng nghiệp vụ API gửi yêu cầu ký, luồng nghiệp vụ API đăng ký thiết bị xác thực và xác thực yêu cầu trên ứng dụng mobile.

1.3. Thuật ngữ và các từ viết tắt

Thuật ngữ	Định nghĩa	Ghi chú
Mysign	Dịch vụ chữ ký số từ xa của Viettel	
Remote signing, CloudCA, Cloud-CA	Các từ định nghĩa khác nhau của dịch vụ ký số từ xa, trong tài liệu này là dịch vụ Mysign của Viettel	
CTS	Chứng thư số	
App, app mobile	Ứng dụng di động	

1.4. Cấu trúc tài liệu

Tài liệu sẽ có cấu trúc như sau:

- Giới thiệu chung về luồng sử dụng dịch vụ Mysign
- Giới thiệu chi tiết về các luồng API sử dụng dịch vụ
- Phụ lục:
 - o Phụ lục chi tiết các API của dịch vụ và SDK mobile
 - o Phụ lục hướng dẫn tích hợp SDK
 - o Phụ lục thông tin user, client test dịch vụ

2. GIỚI THIỆU CHUNG:

2.1. Giới thiệu chung về dịch vụ Mysign:

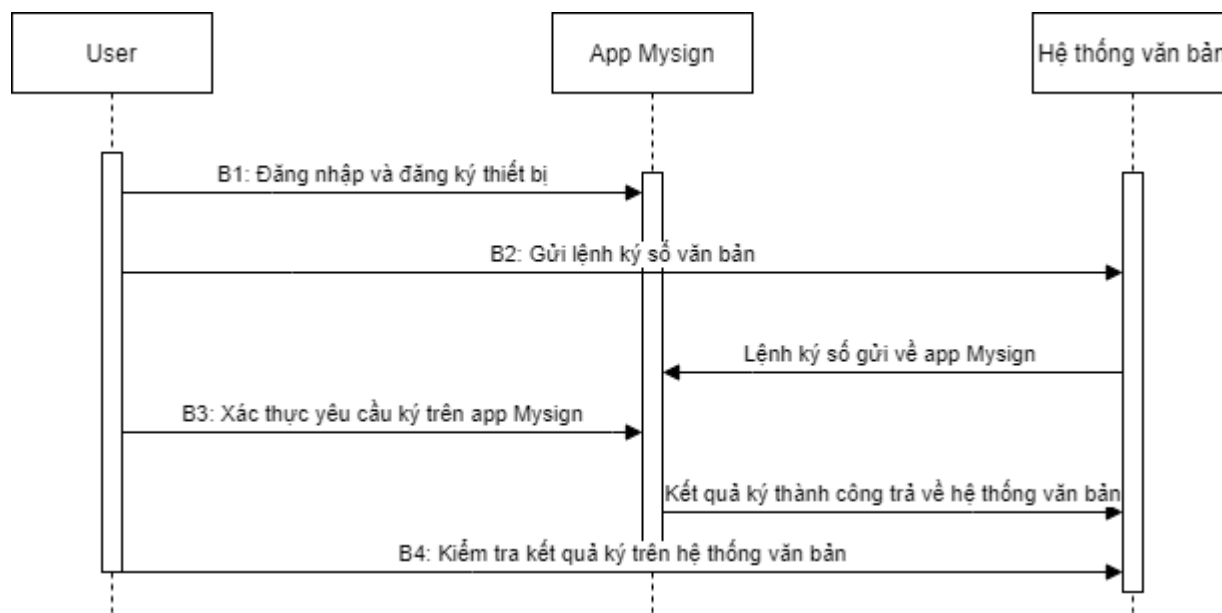
Dịch vụ Mysign là dịch vụ ký số từ xa của Viettel thuộc dịch vụ chứng thư số Viettel-CA, với một số điểm chính như sau:

- Mỗi khách hàng được cấp phát chứng thư số Viettel RS (Remote Signing), có giá trị pháp lý được công nhận bởi dịch vụ Viettel-CA và Bộ Thông tin và Truyền thông.
- Mỗi khách hàng sử dụng dịch vụ sẽ được cấp phát 1 tài khoản Mysign để thực hiện ký số trên ứng dụng Mysign hoặc ứng dụng bên thứ 3 được ủy quyền.
- Khách hàng khi thực hiện lệnh ký văn bản trên các hệ thống văn bản hỗ trợ dịch vụ Mysign, sẽ đăng nhập vào ứng dụng Mysign hoặc ứng dụng mobile của hệ thống đó (đã tích hợp ký) để thực hiện xác thực/ký số sử dụng sinh trắc học của thiết bị di động.

2.2. Luồng thực hiện ký số của người dùng sử dụng ứng dụng Mysign:

Khi được cấp phát tài khoản, người dùng sẽ sử dụng thông tin đó để đăng nhập vào ứng dụng Mysign, thực hiện đăng ký thiết bị đang sử dụng thành thiết bị xác thực của tài khoản. Ngoài ra có thể kiểm tra thông tin CTS, mua gói cước,...

Với hệ thống văn bản đã tích hợp dịch vụ Mysign, khi người dùng thực hiện ký, một yêu cầu ký sẽ được gửi tới ứng dụng Mysign trên thiết bị của người dùng để xác thực. Sau khi xác thực thành công, kết quả sẽ được trả lại cho hệ thống văn bản.



B1: Người dùng đăng nhập vào ứng dụng Mysign, thực hiện đăng ký thiết bị xác thực bằng mã OTP được gửi về thiết bị. Sau khi xác thực thành công, người dùng được truy cập vào ứng dụng Mysign và có thể thực hiện các thao tác tiếp theo (ký, xem thông tin CTS, cài đặt ngôn ngữ,...)

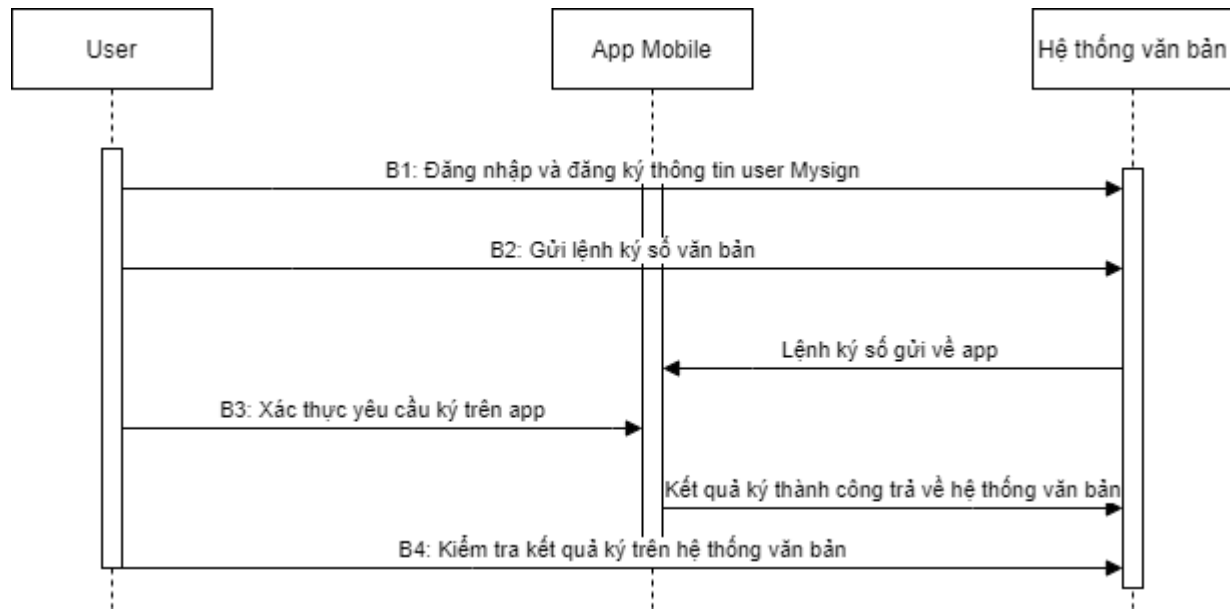
B2: Người dùng thực hiện truy cập vào hệ thống văn bản, và thực hiện gửi lệnh ký Mysign từ hệ thống đó.

B3: Người dùng vào ứng dụng Mysign để xác thực yêu cầu ký được gửi tới bằng sinh trắc học.

B4: Sau khi xác thực thành công, kết quả được trả về hệ thống văn bản. Người dùng kiểm tra văn bản đã ký trên hệ thống văn bản.

2.3. Luồng thực hiện ký số của người dùng sử dụng ứng dụng bên thứ ba:

Với người dùng ký số trực tiếp trên ứng dụng di động của bên thứ ba (thường là ứng dụng của hệ thống văn bản), người dùng thực hiện các thao tác trên ứng dụng đó. Các thao tác bao gồm đăng ký thiết bị, xác thực yêu cầu ký.



B1: Người dùng thực hiện đăng nhập ứng dụng di động. Sau đó, thực hiện đăng ký/liên kết thông tin tài khoản Mysign (do Viettel cấp) vào thông tin người dùng.

B2: Người dùng thực hiện truy cập vào hệ thống văn bản, và thực hiện gửi lệnh ký Mysign từ hệ thống đó.

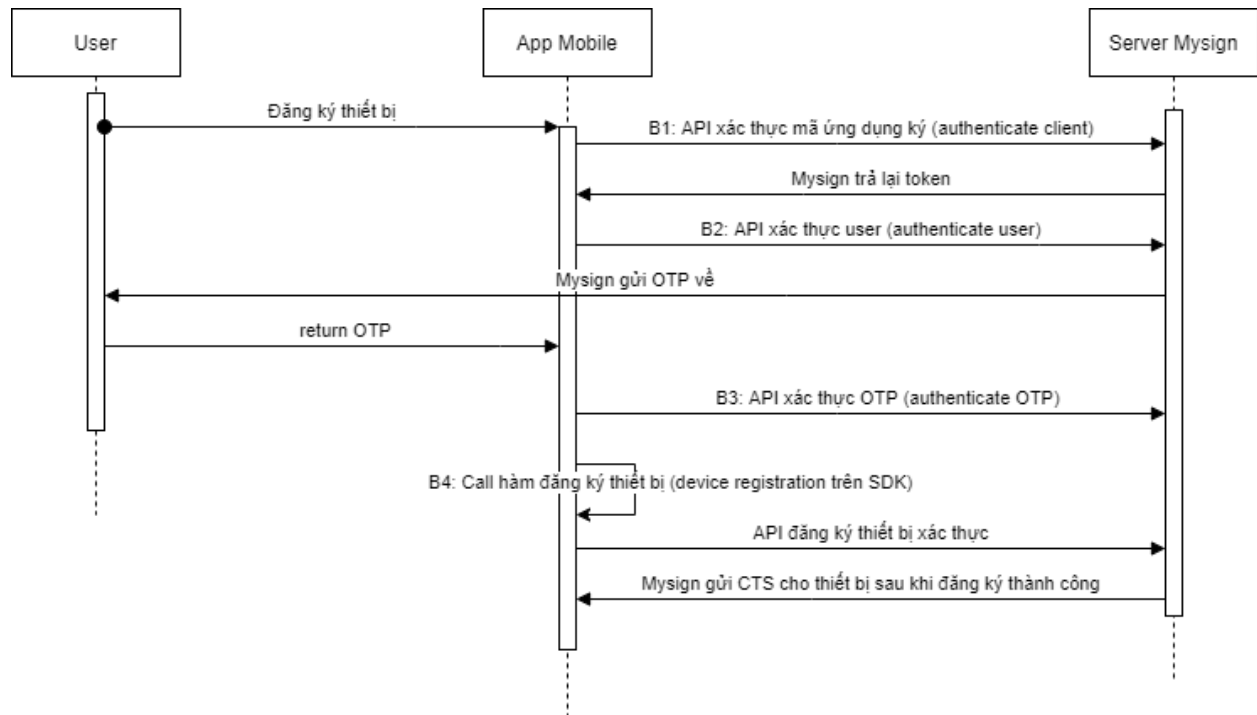
B3: Người dùng xác thực yêu cầu ký trên ứng dụng di động của hệ thống văn bản.

B4: Nhận và kiểm tra văn bản đã ký trên hệ thống văn bản.

3. CHI TIẾT LUỒNG API DỊCH VỤ MYSIGN:

3.1. Luồng API đăng ký thiết bị làm thiết bị xác thực

Lưu ý: Luồng chỉ áp dụng cho các hệ thống có ứng dụng tích hợp SDK Mobile để xác thực yêu cầu ký, không sử dụng Mysign



Với tài khoản sử dụng app trên thiết bị lần đầu, cần thực hiện xác thực tài khoản, xác thực OTP và đăng ký thiết bị. Sau đó có thể thực hiện lấy yêu cầu ký đang chờ và xác thực.

B1: Gọi [API xác thực client \(authenticate client\)](#) để xác thực Client, với mã client_id và client_secret được Viettel cung cấp. Response trả về token để thực hiện gọi các API ở B2, B3.

B2: Gọi [API xác thực user \(authenticate user\)](#) để xác thực User Mysign sử dụng tên tài khoản Mysign Viettel cung cấp, sau khi xác thực tài khoản có tồn tại API sẽ gửi OTP tới các thông tin đã đăng ký (SDT, email).

B3: Gọi [API xác thực OTP \(authenticate OTP\)](#) để xác thực OTP, sử dụng token ở bước 1 và OTP ở bước 2. Sau khi xác thực thành công sẽ nhận được access_token và refresh_token.

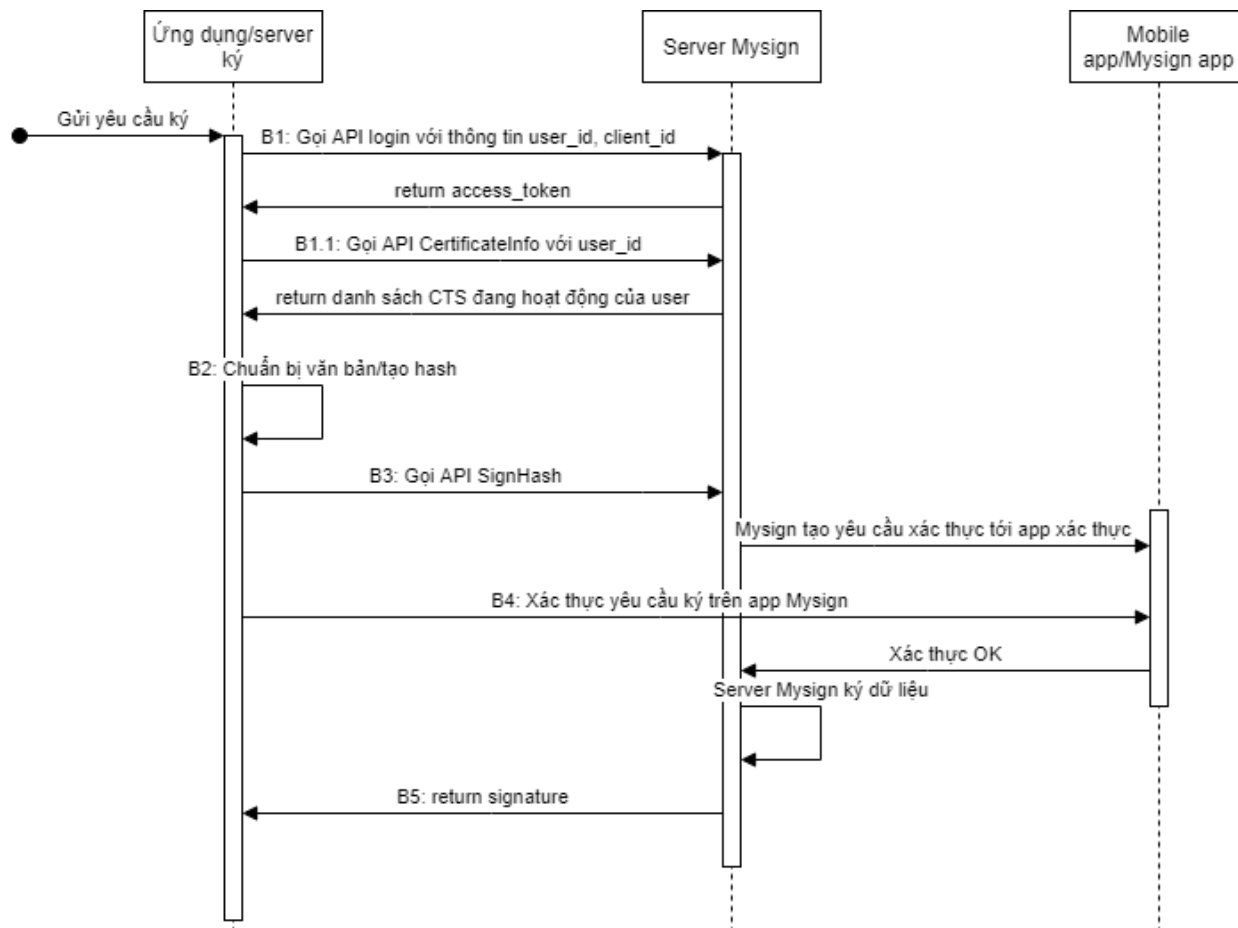
Access_token ở bước này sẽ dùng để truyền vào các API sau, được gán thông tin user đã xác thực ở bước 1-3.

Refresh_token ở bước này sẽ dùng để làm mới access_token ở các lần thực hiện sau.

B4: Gọi **hàm đăng ký thiết bị trong SDK**, thực hiện đăng ký thiết bị sử dụng App là thiết bị xác thực tin cậy. Sử dụng access_token ở bước 3.

- Android: hàm **registerDevice** (chi tiết tham khảo [Phụ lục 3, mục 3.2](#))
- iOS: hàm **API.registerDevice** (chi tiết tham khảo [Phụ lục 4, mục 5.1](#))

3.2. Luồng API gửi yêu cầu ký số trên hệ thống ứng dụng (đồng bộ)



B1: Gọi [API Login](#) với thông tin tên tài khoản Mysign (user_id), mã ứng dụng (client_id) và client_secret để lấy mã access_token.

Access_token ở bước này sẽ dùng để truyền vào các API sau, được gán thông tin user đã xác thực.

B1.1: Gọi [API CertificateInfo](#) để lấy thông tin các CTS đang hoạt động của tài khoản Mysign. Trong đó bao gồm thông tin credential_id là định danh của CTS, và nhiều thông tin khác như certificate chain, thời gian có hiệu lực – hết hạn, thông tin cá nhân/tổ chức đại diện, v.v.

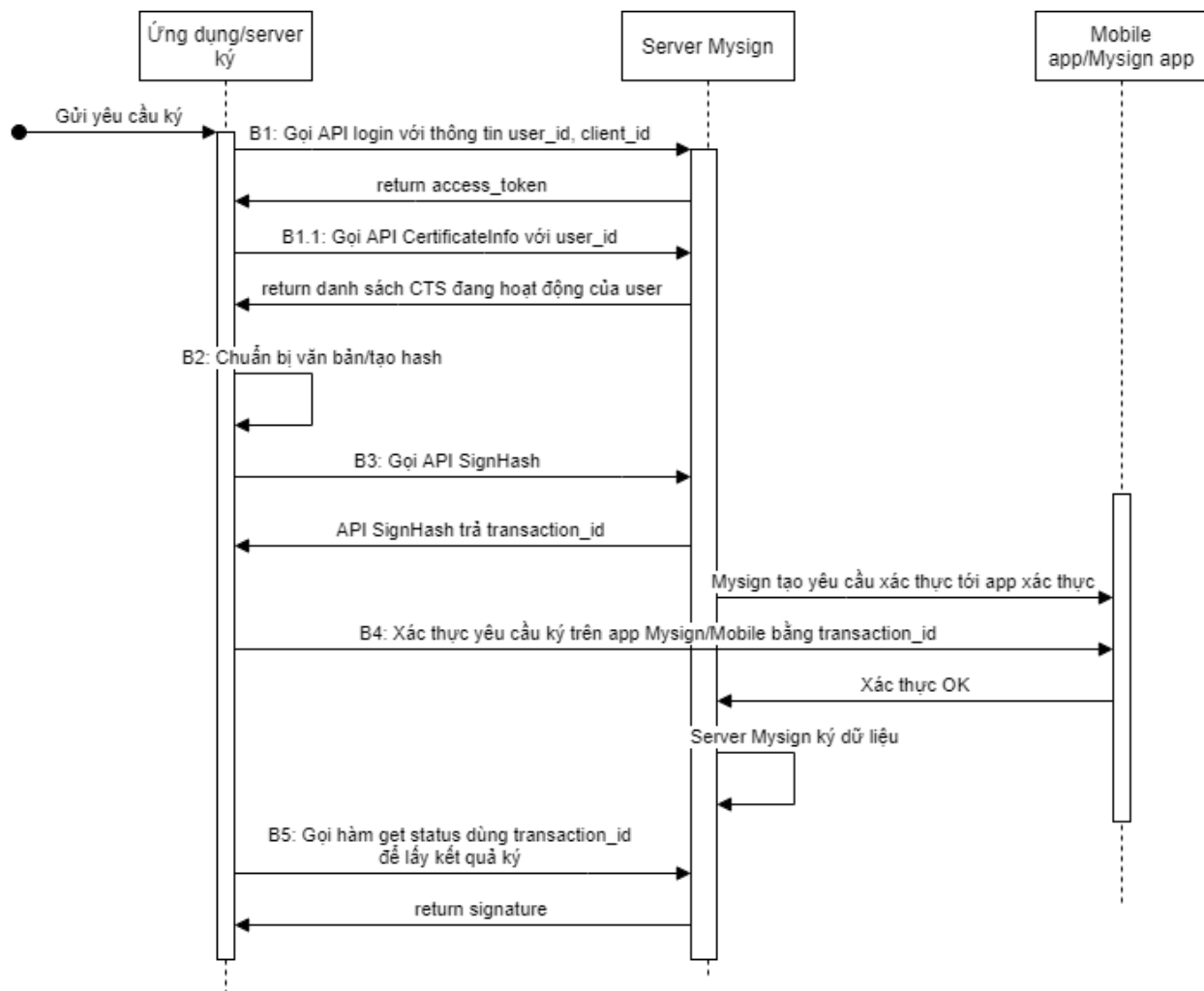
B2: Chuẩn bị các thông tin liên quan tới văn bản ký, chú ý cần có thông tin mã hash của dữ liệu cần ký (chỉ hỗ trợ mã SHA256 đã chuyển sang dạng mã base64), và tên văn bản/dữ liệu ký để hiển thị trên ứng dụng.

B3: Gọi [API SignHash](#) với các thông tin đã có ở các bước trên để gửi yêu cầu ký tới hệ thống Mysign.

B4: Sử dụng ứng dụng xác thực ký (Mysign hoặc App mobile đã tích hợp SDK) để xác thực yêu cầu ký.

B5: Với yêu cầu ký dạng đồng bộ, [API SignHash](#) ở bước 3 sẽ trả kết quả chữ ký (signature) của dữ liệu sau khi xác thực thành công; hoặc mã lỗi nếu ký gặp lỗi/thất bại.

3.3. Luồng API gửi yêu cầu ký số trên hệ thống ứng dụng (bất đồng bộ)



B1: Gọi [API Login](#) với thông tin tên tài khoản Mysign (user_id), mã ứng dụng (client_id) và client_secret để lấy mã access_token.

Access_token ở bước này sẽ dùng để truyền vào các API sau, được gán thông tin user đã xác thực.

B1.1: Gọi [API CertificateInfo](#) để lấy thông tin các CTS đang hoạt động của tài khoản Mysign. Trong đó bao gồm thông tin credential_id là định danh của CTS, và nhiều thông tin khác như certificate chain, thời gian có hiệu lực – hết hạn, thông tin cá nhân/tổ chức đại diện, v.v.

B2: Chuẩn bị các thông tin liên quan tới văn bản ký, chú ý cần có thông tin mã hash của dữ liệu cần ký (chỉ hỗ trợ mã SHA256 đã chuyển sang dạng mã base64), và tên văn bản/dữ liệu ký để hiển thị trên ứng dụng.

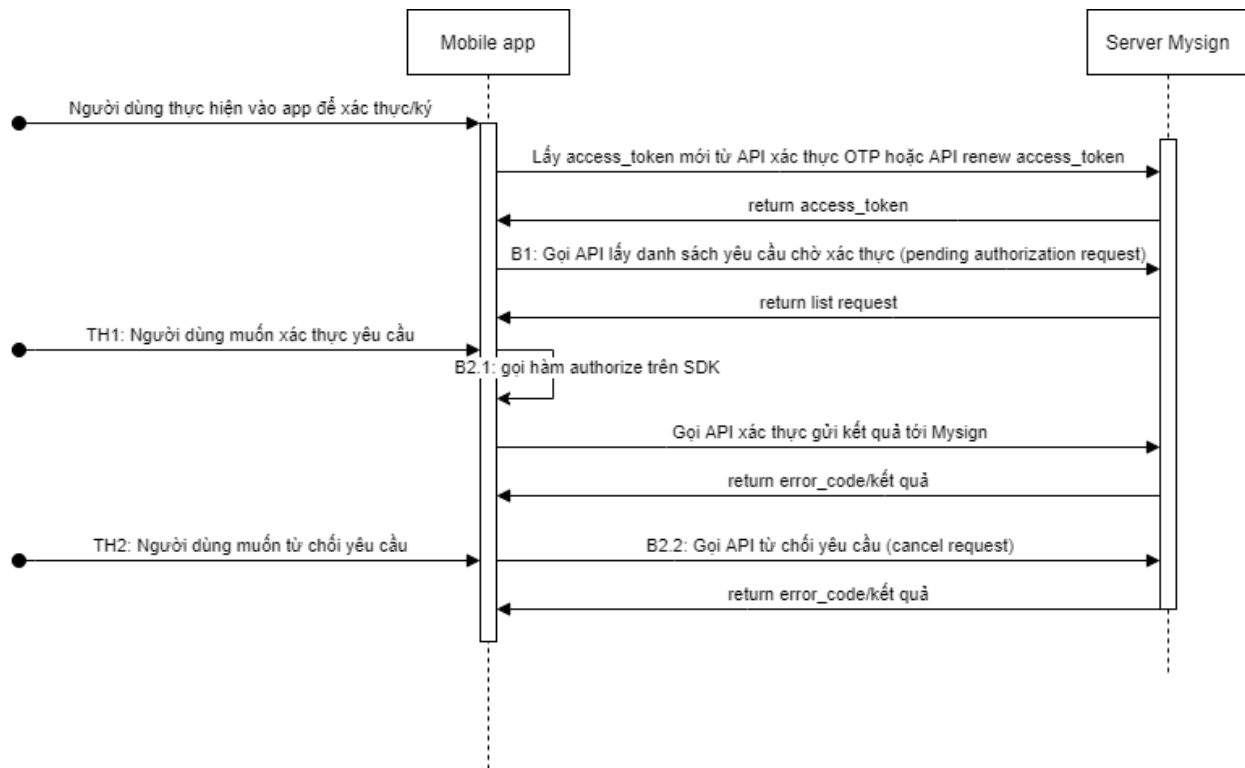
B3: Gọi [API SignHash](#) với các thông tin đã có ở các bước trên để gửi yêu cầu ký tới hệ thống Mysign. Với yêu cầu ký dạng bất đồng bộ, [API SignHash](#) sẽ trả ra mã transaction_id của giao dịch ký

B4: Sử dụng ứng dụng xác thực ký (Mysign hoặc App mobile đã tích hợp SDK) để xác thực yêu cầu ký.

B5: Gọi [API lấy trạng thái của giao dịch](#) để kiểm tra. API SignHash sẽ trả kết quả chữ ký (signature) của dữ liệu sau khi xác thực thành công; hoặc mã lỗi nếu ký gặp lỗi/thất bại.

3.4. Luồng API xác thực yêu cầu ký trên App mobile

Lưu ý: Luồng chỉ áp dụng cho các ứng dụng tích hợp SDK Mobile để xác thực yêu cầu ký, không sử dụng Mysign.



Khi thực hiện các API trong luồng xác thực yêu cầu ký, cần có mã access_token gắn với thông tin user đã xác thực để truyền vào thông tin chính xác.

Trong trường hợp tài khoản đã đăng ký thiết bị, [API xác thực OTP](#) trả lại mã refresh_token để có thể làm mới access_token và thực hiện tiếp các bước xác thực/ủy quyền ký. Sử dụng [API renew access token](#) để làm mới access_token cũ, sử dụng refresh_token từ lần đăng ký đầu tiên.

Sử dụng access_token mới để thực hiện các API ủy quyền ký.

B1: Gọi [API lấy yêu cầu chờ bằng transaction_id](#) để lấy Pending Authorization Request (Yêu cầu cần xác thực ký), sử dụng transaction_id từ bước tạo yêu cầu ký signHash dạng bất đồng bộ ([Mục 3.3](#)). Hàm này sẽ trả mã yêu cầu ký đang đợi xác thực của user. Sau khi người dùng thực hiện API signHash, sẽ có yêu cầu trả về khi thực hiện gọi API này (tham khảo tài liệu API ký của người dùng).

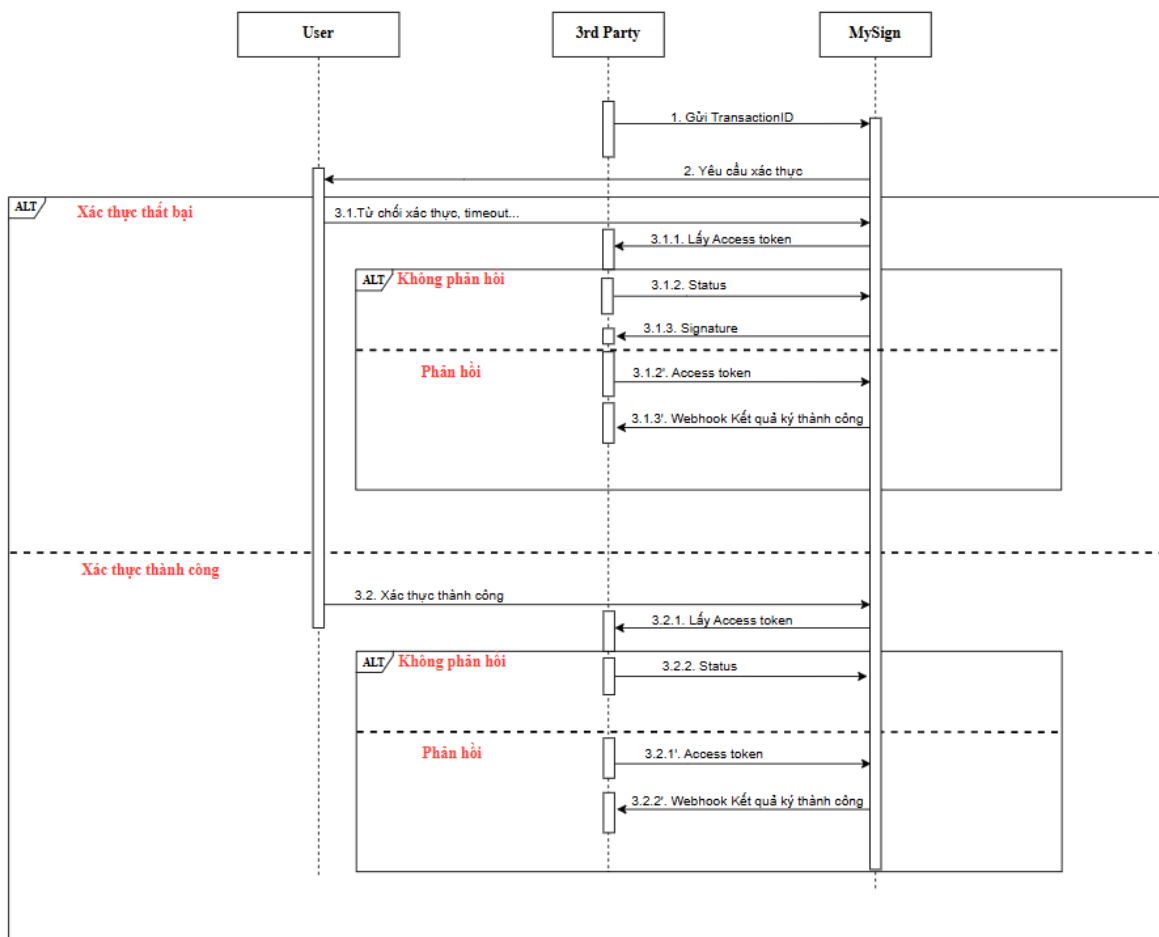
B2.1: Gọi hàm **authorize (ủy quyền ký)** trong SDK khi thực hiện ủy quyền ký cho yêu cầu trên.

- Android: hàm **authorisationPendingRequest** (xem [Phụ lục 3, mục 3.3](#))
- iOS: hàm **API.authoriseaPendingRequest** (xem [Phụ lục 4, mục 5.2](#))

Lưu ý: Nếu sau khi gọi hàm ký, API trả về mã lỗi 57091 – Device certificate is expired, cần thực hiện hủy thiết bị và đăng ký lại thiết bị (Mục 3.1).

B2.2: Gọi [API hủy bỏ yêu cầu](#) nếu muốn từ chối yêu cầu ký.

3.5. Luồng API ký bất đồng bộ có callback



Khi thực hiện các API trong luồng xác thực yêu cầu ký, cần có mã access_token gắn với thông tin user đã xác thực để truyền vào thông tin chính xác.

B1: Thực hiện yêu cầu ký bất đồng bộ với API Mysign, với cấu hình biến **async = 3** (xem [Phụ lục 1, mục 2.2 – Danh sách API ký bất đồng bộ](#)).

B2: Người dùng xác thực thông qua app Mysign hoặc app đã tích hợp SDK ký số của Mysign

B3: Sau khi xử lý xác thực, Mysign gửi lại kết quả thông qua API callback cho đơn vị tích hợp (xem [Phụ lục 2, mục 3 – Danh sách API callback trả kết quả ký](#))

Phụ lục 1: API gửi yêu cầu ký tới dịch vụ Mysign

1. Đặc tả giao tiếp ký đồng bộ

1.1. Các loại giao dịch

STT	Tên hàm	Mô tả
1	Login	Ứng dụng ký đăng nhập và lấy phiên (access_token)
2	Certificates/Info	Lấy danh sách Chứng thư số và Thông tin chi tiết từng Chứng thư số của người ký
3	SignHash	Ký mã băm

1.2. Đặc tả chi tiết các giao dịch

1.2.1. Login

Ứng dụng ký đăng nhập và lấy phiên (access_token).

Request

Link	https://remotesigning.viettel.vn/vtss/service/ras/v1/login
HTTP Verb	POST
Content Type	application/json
Accept	application/json
Request Body	<pre>{ "client_id": "adss...client", "client_secret": "fj49kl.....oOpQS", "profile_id": "ADSS RAS Profile 001", "user_id": "MST_0100109106-998" }</pre>

Parameters	Presence	Value	Description
client_id	MANDATORY	String	Client ID của Ứng dụng ký

Parameters	Presence	Value	Description
			(Tối đa 50 ký tự).
user_id	MANDATORY	String	User ID của người ký (Tối đa 50 ký tự).
client_secret	MANDATORY	String	Khóa bảo mật của Ứng dụng ký
profile_id	MANDATORY	String	Profile ID của người ký (Tối đa 50 ký tự)

Response

Status Code	Message	Response Body
200	OK	{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ", "refresh_token": "", "token_type": "Bearer", "expires_in": "3600" }
400	Bad Request	{ "error": "58071", "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed." }
401	Unauthorized	{ "error": "59033", "error_description": "Failed to process request - user ID or password is"

		<code>invalid"</code> <code>}</code>
--	--	---

Parameters	Presence	Value	Description
access_token	MANDATORY	String	Phiên giao dịch (access token)
refresh_token	CONDITIONAL	String	Token cập nhật lại phiên giao dịch
token_type	MANDATORY	String	Loại access token
expires_in	MANDATORY	String	Hạn của phiên (seconds)
error_code	CONDITIONAL	String	Mã lỗi.
error_description	CONDITIONAL	String	Mô tả lỗi.

1.2.2. CertificateInfo

Lấy danh sách Chứng thư số của người ký

Request

Link	https://remotesigning.viettel.vn/vtss/service/certificates/info
HTTP Verb	POST
Content Type	application/json
Accept	application/json
Authorization	Bearer <code>eyJhbGciOiJIUzI1NiInN...Pcxcz2hM</code>
Request Body	<code>{</code> <code>"client_id": "adss...client",</code> <code>"client_secret": "fj49kl.....oOpQS",</code> <code>"profile_id": "adss:ras:profile:001",</code> <code>"user_id": "MST_0100109106-998",</code>

	<pre>"certificates": "chain", "certInfo": true, "authInfo": true }</pre>
--	--

Parameters	Presence	Value	Description
client_id	MANDATORY	String	Client ID của Ứng dụng ký (Tối đa 50 ký tự).
user_id	MANDATORY	String	User ID của người ký (Tối đa 50 ký tự).
client_secret	MANDATORY	String	Khóa bảo mật của Ứng dụng ký
profile_id	MANDATORY	String	Profile ID của người ký (Tối đa 50 ký tự)
certificates	OPTIONAL	String	<p>Chỉ định thông tin Chứng thư số trả về:</p> <ul style="list-style-type: none"> • None: Không • Single: Chỉ chứng thư số của người ký • Chain: Cả chuỗi chứng thư số (Certificate chain). Giá trị mặc định: 'single'
certInfo	MANDATORY	Boolean	<ul style="list-style-type: none"> • True: Trả về các thông tin kèm theo Chứng thư số. • False: Không trả các thông tin kèm theo Chứng thư số. Giá trị mặc định: 'false'

Parameters	Presence	Value	Description
authInfo	MANDATORY	Boolean	<ul style="list-style-type: none"> • True: Trả về các thông tin về cơ chế ủy quyền được hỗ trợ. • False: Không trả c về các thông tin về cơ chế ủy quyền được hỗ trợ. <p>Giá trị mặc định: 'false'</p>

Response

Status Code	Message	Response Body
200	OK	<pre>[{ "description": "Go>Sign mobile based implicit credential authorization", "key": { "status": "ENABLED", "algo": ["1.2.840.113549.1.1.1"], "len": 2048, "curve": null }, "cert": { "status": "valid", "certificates": ["Base64-encoded X.509 end entity certificate", "Base64-encoded X.509 intermediate CA certificate", "Base64-encoded X.509 issuer CA certificate"], "issuerDN": "Issuer DN printable string", "SerialNumber": "5AAC41CD8FA22B953640", "subjectDN": "Subject DN printable string", "validFrom": "20180709132216+0000", "validTo": "20190709132216+0000" } }</pre>

		<pre> }, "authMode": "implicit", "multisign": "2147483647", "lang": null, "credential_id": "0100109106- 998_2475106_20221011075826", "SCAL": "2", "phone_number": "0000000000", "subscriber_id": "CA_0000000000", "sign_package": "MS_VAS_HCC" }] </pre>
400	Bad Request	<pre> { "error": "invalid_request", "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed." } </pre>

Parameters	Presence	Value	Description
description	OPTIONAL	String	Mô tả (Tối đa 255 ký tự).
key/status	MANDATORY	String	Trạng thái của cặp khóa của Chứng thư số: <ul style="list-style-type: none"> • Enabled: Khóa có thể sử dụng để ký • Disabled: Không không thể dùng để ký.
key/algo	MANDATORY	List<String>	Danh sách OIDs các thuật toán ký hỗ trợ. Ví dụ: <ul style="list-style-type: none"> • 1.2.840.113549.1.1.1=RSA encryption •

Parameters	Presence	Value	Description
			1.2.840.10045.4.3.2=ECDSA with SHA256
key/len	MANDATORY	<i>Number</i>	Độ dài khóa (bits).
key/curve	CONDITIONAL	<i>String</i>	Mã OID của ECDSA curve. Chỉ trả về khi <i>keyAlgo</i> thuộc nhóm ECDSA.
cert/status	OPTIONAL	<i>String</i>	Trạng thái của Chứng thư số.
cert/certificates	CONDITIONAL	<i>List<String></i>	Danh sách chứng thư số định dạng Base64. Theo thứ tự: - CTS của user - CTS Viettel-CA - CTS gốc (root) của Bộ TTTT
cert/issuerDN	CONDITIONAL	<i>String</i>	DN của CA cấp CTS Trả về khi <i>certInfo</i> là “true”.
cert/serialNumber	CONDITIONAL	<i>String</i>	Serial number của CTS Trả về khi <i>certInfo</i> là “true”.
cert/subjectDN	CONDITIONAL	<i>String</i>	DN của CTS Trả về khi <i>certInfo</i> là “true”.
cert/validFrom	CONDITIONAL	<i>String</i>	Thời hạn bắt đầu hợp lệ của CTS Trả về khi <i>certInfo</i> là “true”. Định dạng GeneralizedTime (RFC 5280 e.g. “YYYYMMDDHHMMSSZ”).
cert/validTo	CONDITIONAL	<i>String</i>	Thời hạn kết thúc hợp lệ của CTS Trả về khi <i>certInfo</i> là “true”. Định dạng GeneralizedTime (RFC 5280 e.g. “YYYYMMDDHHMMSSZ”).

Parameters	Presence	Value	Description
authMode	MANDATORY	<i>String</i>	Phương thức xác thực: • Implicit: Việc xác thực do người ký thực hiện.
SCAL	OPTIONAL	<i>String</i>	• “2”: SCAL2 thì mã băm sẽ được liên kết với SAD
multisign	MANDATORY	<i>Number</i>	Một số bằng hoặc cao hơn 1 đại diện cho số lượng chữ ký tối đa có thể được tạo bằng thông tin xác thực này với một yêu cầu ủy quyền duy nhất.
credential_id	MANDATORY	<i>String</i>	Mã credential_id (định danh) của CTS
lang	OPTIONAL	<i>String</i>	Mã ngôn ngữ của kết quả theo RFC 5646.
phone_number	MANDATORY	<i>String</i>	Số điện thoại của thuê bao đăng ký CTS
subscriber_id	MANDATORY	<i>String</i>	Số thuê bao
sign_package	MANDATORY	<i>String</i>	Gói cước ký
error_code	CONDITIONAL	<i>String</i>	Mã lỗi.
error_description	CONDITIONAL	<i>String</i>	Mô tả lỗi.

1.2.3. SignHash

Xác thực yêu cầu ký.

Request

Link	https://remotesigning.viettel.vn/vtss/service/signHash
HTTP Verb	POST

Content Type	application/json
Accept	application/json
Authorization	Bearer _TiHRG-bA H3XIFQZ3ndFhkXf9P24/CKN69L8gdSYp5_pw
Request Body	<pre>{ "credentialID": "JohnDoe", "client_id": "samples_test_client", "client_secret": "12345789qadewaerawer", "numSignatures": 2, "description": "Document Name", "documents": [{ "document_id": 123, "document_name": "Document Name 123", }, { "document_id": 456, "document_name": "Document Name 456", }], "hash": ["sTOgwOm+474gFj0q0x1iSNspKqbcse4IeiqlDg/HWuI=", "c1RPZ3dPbSs0NzRnRmowcTB4MWITTnNwS3FiY3NINEllaXFvRGcvSFd1ST0="], "hashAlgo": "2.16.840.1.101.3.4.2.1", "signAlgo": "1.2.840.113549.1.1.1", "async": 0 }</pre>

Request Parameters

Parameters	Presence	Value	Description
credentialID	MANDATORY	<i>String</i>	ID của Chứng thư số của người ký
client_id	MANDATORY	<i>String</i>	Mã client_id được cung cấp bởi dịch vụ
client_secret	MANDATORY	<i>String</i>	Mã client_secret được cung cấp bởi dịch vụ
numSignatures	MANDATORY	<i>Number</i>	Số lượng chữ ký cần ký
documents/document_id	MANDATORY	<i>String</i>	ID đại diện cho tài liệu ký
documents/document_name	MANDATORY	<i>String</i>	<p>* Tên của tài liệu ký, hiển thị trên yêu cầu ký app Mysign nếu không có thông tin description</p> <p>* Để hiển thị chính xác nội dung này, cần truyền thông tin như sau:</p> <ul style="list-style-type: none"> - Mã hóa nội dung dạng mã Base64, encode UTF-8 khi truyền vào API - Độ dài mã Base64 nhỏ hơn 100 kí tự - Nội dung gốc chỉ được bao hàm các nội dung sau: <ul style="list-style-type: none"> + Chữ cái hoa, thường không dấu: a-z, A-Z + Chữ số: 0-9 + Ký tự đặc biệt: dấu gạch dưới (_), dấu gạch ngang (-), dấu cách ()
hash	CONDITIONAL	<i>List<String></i>	Mã băm của tài liệu ký dạng Base64.

Parameters	Presence	Value	Description
			<p>Mỗi mã sẽ tương ứng với 1 object trên param documents theo đúng trình tự sắp xếp.</p> <p>Chỉ tiếp nhận mã hash dạng SHA256 đã chuyển đổi về dạng base64</p>
description	OPTIONAL	String	<p>* Mô tả yêu cầu ký.</p> <p>* Nội dung sẽ hiển thị trên yêu cầu ký app Mysign nếu được truyền vào</p> <p>* Để hiển thị chính xác nội dung này, cần truyền thông tin như sau:</p> <ul style="list-style-type: none"> - Mã hóa nội dung dạng mã Base64, encode UTF-8 khi truyền vào API - Độ dài mã Base64 nhỏ hơn 100 kí tự - Nội dung gốc chỉ được bao hàm các nội dung sau: <ul style="list-style-type: none"> + Chữ cái hoa, thường không dấu: a-z, A-Z + Chữ số: 0-9 + Ký tự đặc biệt: dấu gạch dưới (_), dấu gạch ngang (-), dấu cách ()
hashAlgo	CONDITIONAL	String	<p>Mã OID của thuật toán băm.</p> <p>Tham số này sẽ bị bỏ qua hoặc bỏ qua nếu thuật toán băm được chỉ định ngầm bởi thuật toán signAlgo.</p>
signAlgo	MANDATORY	String	<p>OID của thuật toán được sử dụng để ký. Nó sẽ là một trong những giá trị được thông tin xác thực cho phép như</p>

Parameters	Presence	Value	Description
			được trả về trong <i>keyAlgo</i> trong kết quả khi gọi API credentials/info.
async	MANDATORY	Integer	<p>Chế độ ký đồng bộ/bất đồng bộ</p> <p>0: Chế độ ký đồng bộ (API sẽ chờ kết quả xác thực từ server)</p> <p>1: Chế độ ký bất đồng bộ (API sẽ trả mã yêu cầu ký để người dùng gọi hàm kiểm tra)</p> <p>Ở loại yêu cầu ký đồng bộ, giá trị mã = 0</p>

Response

Status Code	Message	Response Body
200	OK	<pre>{ "signatures": ["KeTob5gl26S2tmXjqN...MRGtoew=="] }</pre>
400	Bad Request	<pre>{ "error": "invalid_request", "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed." }</pre>
400	Bad Request	<pre>{ "error": "invalid_request", "error_description": "Missing (or invalid type)" }</pre>

		<code>string parameter credentialID"</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "Invalid parameter credentialID"</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "Missing (or invalid type) integer parameter numSignatures"</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "Invalid parameter numSignatures"</code> <code>}</code>

Response Parameters

Parameters	Presence	Value	Description
signatures	MANDATORY	<i>String</i>	Danh sách chữ ký dạng base64 tương ứng với danh sách mã băm đầu vào.
error_code	CONDITIONAL	<i>String</i>	Mã lỗi.
error_description	CONDITIONAL	<i>String</i>	Mô tả lỗi.

2. Đặc tả giao tiếp ký bất đồng bộ

2.1. Các loại giao dịch

STT	Tên hàm	Mô tả
1	Login	Ứng dụng ký đăng nhập và lấy phiên (access_token)
2	Certificates/Info	Lấy danh sách Chứng thư số và Thông tin chi tiết từng Chứng thư số của người ký
3	SignHash	Ký mã băm (bất đồng bộ)
4	Get signing request status	Lấy kết quả yêu cầu ký

2.2. Đặc tả chi tiết các giao dịch

2.2.1. Login

Ứng dụng ký đăng nhập và lấy phiên (access_token).

Request

Link	https://remotesigning.viettel.vn/vtss/service/ras/v1/login
HTTP Verb	POST
Content Type	application/json
Accept	application/json
Request Body	<pre>{ "client_id": "adss...client", "client_secret": "fj49kl.....oOpQS", "profile_id": "ADSS RAS Profile 001", "user_id": "MST_0100109106-998" }</pre>

Parameters	Presence	Value	Description
client_id	MANDATORY	String	Client ID của Ứng dụng ký

Parameters	Presence	Value	Description
			(Tối đa 50 ký tự).
user_id	MANDATORY	String	User ID của người ký (Tối đa 50 ký tự).
client_secret	MANDATORY	String	Khóa bảo mật của Ứng dụng ký
profile_id	MANDATORY	String	Profile ID của người ký (Tối đa 50 ký tự)

Response

Status Code	Message	Response Body
200	OK	{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ6MmM", "refresh_token": "", "token_type": "Bearer", "expires_in": "3600" }
400	Bad Request	{ "error": "58071", "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed." }
401	Unauthorized	{ "error": "59033", }

		<pre>"error_description": "Failed to process request - user ID or password is invalid" }</pre>
--	--	--

Parameters	Presence	Value	Description
access_token	MANDATORY	String	Phiên giao dịch (access token)
refresh_token	CONDITIONAL	String	Token cập nhật lại phiên giao dịch
token_type	MANDATORY	String	Loại access token
expires_in	MANDATORY	String	Hạn của phiên (seconds)
error_code	CONDITIONAL	String	Mã lỗi.
error_description	CONDITIONAL	String	Mô tả lỗi.

2.2.2. CertificateInfo

Lấy danh sách Chứng thư số của người ký

Request

Link	https://remotesigning.viettel.vn/vtss/service/certificates/info
HTTP Verb	POST
Content Type	application/json
Accept	application/json
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJhM
Request Body	<pre>{ "client_id": "adss...client", "client_secret": "fj49kl.....oOpQS", }</pre>

	<pre> "profile_id": "adss:ras:profile:001", "user_id": "MST_0100109106-998", "certificates": "chain", "certInfo": true, "authInfo": true </pre>
--	--

Parameters	Presence	Value	Description
client_id	MANDATORY	String	Client ID của Ứng dụng ký (Tối đa 50 ký tự).
user_id	MANDATORY	String	User ID của người ký (Tối đa 50 ký tự).
client_secret	MANDATORY	String	Khóa bảo mật của Ứng dụng ký
profile_id	MANDATORY	String	Profile ID của người ký (Tối đa 50 ký tự)
certificates	MANDATORY	String	<p>Chỉ định thông tin Chứng thư số trả về:</p> <ul style="list-style-type: none"> • None: Không • Single: Chỉ chứng thư số của người ký • Chain: Cả chuỗi chứng thư số (Certificate chain). Giá trị mặc định: 'single'
certInfo	MANDATORY	Boolean	<ul style="list-style-type: none"> • True: Trả về các thông tin kèm theo Chứng thư số. • False: Không trả các thông tin kèm theo Chứng thư số.

Parameters	Presence	Value	Description
			Giá trị mặc định: 'false'
authInfo	MANDATORY	Boolean	<ul style="list-style-type: none"> • True: Trả về các thông tin về cơ chế ủy quyền được hỗ trợ. • False: Không trả về các thông tin về cơ chế ủy quyền được hỗ trợ. <p>Giá trị mặc định: 'false'</p>

Response

Status Code	Message	Response Body
200	OK	<pre>[{ "description": "Go>Sign mobile based implicit credential authorization", "key": { "status": "ENABLED", "algo": ["1.2.840.113549.1.1.1"], "len": 2048, "curve": null }, "cert": { "status": "valid", "certificates": ["Base64-encoded X.509 end entity certificate", "Base64-encoded X.509 intermediate CA certificate", "Base64-encoded X.509 issuer CA certificate"] } }]</pre>

		<pre> "issuerDN": "Issuer DN printable string", "SerialNumber": "5AAC41CD8FA22B953640", "subjectDN": "Subject DN printable string", "validFrom": "20180709132216+0000", "validTo": "20190709132216+0000" }, "authMode": "implicit", "multisign": "2147483647", "lang": null, "credential_id": "0100109106-998_2475106_20221011075826", "SCAL": "2", "phone_number": "00000000000", "subscriber_id": "CA_000000000000", "sign_package": "MS_VAS_HCC" }] </pre>
400	Bad Request	<pre> { "error": "invalid_request", "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed." } </pre>

Parameters	Presence	Value	Description
description	OPTIONAL	<i>String</i>	Mô tả (Tối đa 255 ký tự).
key/status	MANDATORY	<i>String</i>	Trạng thái của cặp khóa của Chứng thư số:

Parameters	Presence	Value	Description
			<ul style="list-style-type: none"> • Enabled: Khóa có thể sử dụng để ký • Disabled: Không thể dùng để ký.
key/algo	MANDATORY	String	<p>Danh sách OIDs các thuật toán ký hỗ trợ. Ví dụ:</p> <ul style="list-style-type: none"> • 1.2.840.113549.1.1.1=RSA encryption • 1.2.840.10045.4.3.2=ECDSA with SHA256
key/len	MANDATORY	Number	Độ dài khóa (bits).
key/curve	CONDITIONAL	String	<p>Mã OID của ECDSA curve.</p> <p>Chỉ trả về khi <i>keyAlgo</i> thuộc nhóm ECDSA.</p>
cert/status	OPTIONAL	String	Trạng thái của Chứng thư số.
cert/certificates	CONDITIONAL	List<String>	<p>Danh sách chứng thư số định dạng Base64.</p> <p>Theo thứ tự:</p> <ul style="list-style-type: none"> - CTS của user - CTS Viettel-CA - CTS gốc (root) của Bộ TTTT

Parameters	Presence	Value	Description
cert/issuerDN	CONDITIONAL	String	DN của CA cấp CTS Trả về khi <i>certInfo</i> là “true”.
cert/serialNumber	CONDITIONAL	String	Serial number của CTS Trả về khi <i>certInfo</i> là “true”.
cert/subjectDN	CONDITIONAL	String	DN của CTS Trả về khi <i>certInfo</i> là “true”.
cert/validFrom	CONDITIONAL	String	Thời hạn bắt đầu hợp lệ của CTS Trả về khi <i>certInfo</i> là “true”. Định dạng GeneralizedTime (RFC 5280 e.g. “YYYYMMDDHHMMSS Z”).
cert/validTo	CONDITIONAL	String	Thời hạn kết thúc hợp lệ của CTS Trả về khi <i>certInfo</i> là “true”. Định dạng GeneralizedTime (RFC 5280 e.g. “YYYYMMDDHHMMSS Z”).
authMode	MANDATORY	String	Phương thức xác thực: • Implicit: Việc xác thực do người ký thực hiện.

Parameters	Presence	Value	Description
SCAL	OPTIONAL	<i>String</i>	<ul style="list-style-type: none"> “2”: SCAL2 thì mã băm sẽ được liên kết với SAD
multisign	MANDATORY	<i>Number</i>	Một số bằng hoặc cao hơn 1 đại diện cho số lượng chữ ký tối đa có thể được tạo bằng thông tin xác thực này với một yêu cầu ủy quyền duy nhất.
credential_id	MANDATORY	<i>String</i>	Mã credential_id (định danh) của CTS
lang	OPTIONAL	<i>String</i>	Mã ngôn ngữ của kết quả theo RFC 5646.
phone_number	MANDATORY	<i>String</i>	Số điện thoại của thuê bao đăng ký CTS
subscriber_id	MANDATORY	<i>String</i>	Số thuê bao
sign_package	MANDATORY	<i>String</i>	Gói cước ký
error_code	CONDITIONAL	<i>String</i>	Mã lỗi.
error_description	CONDITIONAL	<i>String</i>	Mô tả lỗi.

2.2.3. SignHash

Xác thực yêu cầu ký.

Request

Link	https://remotesigning.viettel.vn/vtss/service/signHash
HTTP Verb	POST
Content Type	application/json
Accept	application/json

Authorization	Bearer _TiHRG-bA H3XIFQZ3ndFhkXf9P24/CKN69L8gdSYp5_pw
Request Body	{ "credentialID": "JohnDoe", "client_id": "samples_test_client", "client_secret": "12345789qadewaerawer", "numSignatures": 2, "description": "Document Name", "documents": [{ "document_id": 123, "document_name": "Document Name 123", },{ "document_id": 456, "document_name": "Document Name 456", }], "hash": ["sTOgwOm+474gFj0q0x1iSNspKqbcse4IeiqIDg/ HWuI=", "c1RPZ3dPbSs0NzRnRmowcTB4MWITTnNwS3 FiY3NINEllaXFfsRGcvSFd1ST0=",], "hashAlgo": "2.16.840.1.101.3.4.2.1", "signAlgo": "1.2.840.113549.1.1.1", "async": 2 }

Request Parameters

Parameters	Presence	Value	Description
credentialID	MANDATORY	String	ID của Chứng thư số của người ký
client_id	MANDATORY	String	Mã client_id được cung cấp bởi dịch vụ

Parameters	Presence	Value	Description
client_secret	MANDATORY	<i>String</i>	Mã client_secret được cung cấp bởi dịch vụ
numSignatures	MANDATORY	<i>Number</i>	Số lượng chữ ký cần ký
documents/document_id	MANDATORY	<i>String</i>	ID đại diện cho tài liệu ký
documents/document_name	MANDATORY	<i>String</i>	<p>* Tên của tài liệu ký, hiển thị trên yêu cầu ký app Mysign nếu không có thông tin description</p> <p>* Để hiển thị chính xác nội dung này, cần truyền thông tin như sau:</p> <ul style="list-style-type: none"> - Mã hóa nội dung dạng mã Base64, encode UTF-8 khi truyền vào API - Độ dài mã Base64 nhỏ hơn 100 kí tự - Nội dung gốc chỉ được bao hàm các nội dung sau: <ul style="list-style-type: none"> + Chữ cái hoa, thường không dấu: a-z, A-Z + Chữ số: 0-9 + Ký tự đặc biệt: dấu gạch dưới (_), dấu gạch ngang (-), dấu cách ()
hash	CONDITIONAL	<i>List<String></i>	Mã băm của tài liệu ký dạng Base64.

Parameters	Presence	Value	Description
			<p>Mỗi mã sẽ tương ứng với 1 object trên param documents theo đúng trình tự sắp xếp.</p> <p>Chỉ tiếp nhận mã hash dạng SHA256 đã chuyển đổi về dạng base64</p>
description	MANDATORY	String	<p>* Mô tả yêu cầu ký.</p> <p>* Nội dung sẽ hiển thị trên yêu cầu ký app Mysign nếu được truyền vào</p> <p>* Để hiển thị chính xác nội dung này, cần truyền thông tin như sau:</p> <ul style="list-style-type: none"> - Mã hóa nội dung dạng mã Base64, encode UTF-8 khi truyền vào API - Độ dài mã Base64 nhỏ hơn 100 kí tự - Nội dung gốc chỉ được bao hàm các nội dung sau: <ul style="list-style-type: none"> + Chữ cái hoa, thường không dấu: a-z, A-Z + Chữ số: 0-9 + Ký tự đặc biệt: dấu gạch dưới (_), dấu gạch ngang (-), dấu cách ()
hashAlgo	CONDITIONAL	String	<p>Mã OID của thuật toán băm.</p> <p>Tham số này sẽ bị bỏ qua hoặc bỏ qua nếu thuật toán băm được chỉ</p>

Parameters	Presence	Value	Description
			định ngầm bởi thuật toán signAlgo.
signAlgo	MANDATORY	String	OID của thuật toán được sử dụng để ký. Nó sẽ là một trong những giá trị được thông tin xác thực cho phép như được trả về trong <i>keyAlgo</i> trong kết quả khi gọi API credentials/info.
async	MANDATORY	Integer	<p>Chế độ ký đồng bộ/bất đồng bộ</p> <p>0: Chế độ ký đồng bộ (API sẽ chờ kết quả xác thực từ server)</p> <p>2: Chế độ ký bất đồng bộ (API sẽ trả mã yêu cầu ký để người dùng gọi hàm kiểm tra)</p> <p>Ở loại yêu cầu ký bất đồng bộ, giá trị mã = 2</p>

Response

Status Code	Message	Response Body
200	OK	<pre>{ "transactionId": "de67f948-0498-4919-af28-dff54a6a4e77" }</pre>
400	Bad Request	<pre>{ "error": "invalid_request", "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed." }</pre>

400	Bad Request	{ "error": "invalid_request", "error_description": "Missing (or invalid type) string parameter credentialID" }
400	Bad Request	{ "error": "invalid_request", "error_description": "Invalid parameter credentialID" }
400	Bad Request	{ "error": "invalid_request", "error_description": "Missing (or invalid type) integer parameter numSignatures" }
400	Bad Request	{ "error": "invalid_request", "error_description": "Invalid parameter numSignatures" }

Response Parameters

Parameters	Presence	Value	Description
signatures	MANDATORY	<i>String</i>	Danh sách chữ ký dạng base64 tương ứng với danh sách mã băm đầu vào.
error_code	CONDITIONAL	<i>String</i>	Mã lỗi.
error_description	CONDITIONAL	<i>String</i>	Mô tả lỗi.

2.2.4. Get signing request status

Xác thực yêu cầu ký.

Request

Link	https://remotesigning.viettel.vn/vtss/service/request s/status
HTTP Verb	POST
Content Type	application/json
Accept	application/json
Authorization	Bearer _TiHRG-bA H3X1FQZ3ndFhkXf9P24/CKN69L8gdSYp5_pw
Request Body	{ "transactionId": "de67f948-0498-4919-af28- dff54a6a4e77" }

Request Parameters

Parameters	Presence	Value	Description
transactionId	MANDATORY	String	ID của Yêu cầu ký cần tra kết quả

Response

Status Code	Message	Response Body
200	OK	{ "signatures": ["KeTob5gl26S2tmXjqN...MRGtoew=="], }

		<code>"status": "1"</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed."</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "Missing (or invalid type) string parameter credentialID"</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "Invalid parameter credentialID"</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "Missing (or invalid type) integer parameter numSignatures"</code> <code>}</code>
400	Bad Request	<code>{</code> <code>"error": "invalid_request",</code> <code>"error_description": "Invalid parameter numSignatures"</code> <code>}</code>

Response Parameters

Parameters	Presence	Value	Description
signatures	MANDATORY	<i>String</i>	Danh sách chữ ký dạng base64 tương ứng với danh sách mã băm đầu vào.
status	MANDATORY	<i>String</i>	Mã trạng thái của yêu cầu ký đang tra cứu: 1 – Success 4000 – Chờ người dùng xác nhận 6000 – Yêu cầu đã được người dùng xác nhận, hệ thống đang ký 4001 – Yêu cầu ký hết thời gian chờ (Timeout) 4002 – Người dùng từ chối ký 4004 – Ký thất bại (có lỗi xảy ra) 4005 – Tài khoản không đủ số dư/lượt ký 13004 – Chứng thư số hết hạn hoặc bị thu hồi 50000 – Có lỗi xảy ra khi lấy thông tin
error_code	CONDITIONAL	<i>String</i>	Mã lỗi.
error_description	CONDITIONAL	<i>String</i>	Mô tả lỗi.

3. Đặc tả API ký bất đồng bộ có callback (async = 3)

3.1. API Xác thực của đối tác

Đối tác cần chuẩn bị API để Mysign gọi sang lấy token: Link theo IP + Domain

API xác thực client_id, client_secret là các thông tin xác thực của client được hệ thống Mysign cấp. Sau khi xác thực thành công, hệ thống trả lại mã access_token xác thực cho các API 1.2 và 1.3. (application_access_token)

Request

https://<API URL> (API URL cho đối tác cung cấp - Lưu ý sử dụng https)	
HTTP Verb	GET
Content-Type	application/json
Accept	application/json
Request body (json)	{ "grant_type": "client_credentials", "client_id": "simple_callback_code", "client_secret": "205640fd6ea8c7d80bb91c630b52d286d21ee511" }
Response Header	
authentication_methods	true

Response

Status Code	Message	Response Body
200	OK	{ "status_code": "00", "message": "Thành công", "data": { "access_token": "Ex2YHUSZqQQB8Smh62zcZDUbxIsbMVR8PKmgK70Z", "expires_in": "3600000",

		"consented_on": "1735035122" } }
400	Bad Request	Message lỗi
500	Internal Server Error	Message lỗi

Request Parameters

Parameters	Presence	Value	Description
grant_type	forward	String	Mặc định là: client_credentials
client_id	forward	String	Do đối tác cung cấp
client_secret	forward	String	Do đối tác cung cấp

Response Parameters

Parents Parameters	Parameters	Presence	Value	Description
	status_code	REQUIRED	String	Mã lỗi của request
	message	REQUIRED	String	Mô tả mã lỗi request
data		REQUIRED	Object	Data
data	access_token	REQUIRED	String	Access_token sử dụng cho API 3.2
data	expires_in	REQUIRED	Number	Khoảng thời gian access_token còn hiệu lực
data	consented_on	REQUIRED	Timestamp	Thời điểm access_token được sinh ra (Dựa vào Unix time second)

Bảng mã lỗi

Hệ thống Mysign chấp nhận mã lỗi 00 là yêu cầu thực hiện thành công.

status_code	message	Mô tả
00	Thành công	Lấy thông tin access token thành công
Khác 00	Bên thứ 3 trả mô tả lỗi ở trường message	Không thành công, dừng luồng callback trả kết quả

3.2. API Trả kết quả ký số cho bên thứ 3

Đối tác cần chuẩn bị API để Mysign gọi sang trả kết quả ký số: Link theo IP + Domain

API trả kết quả ký số cho bên thứ 3.

- Request:

https://<API URL> (API URL cho đối tác cung cấp - Lưu ý sử dụng https)	
HTTP Verb	GET
Content-Type	application/json
Accept	application/json
Authorization	Bearer {access_token}
Request body (json)	<pre>{ "meta_data": { "request_id": "VIETTEL_6fa0d53c-e225-45bd-8a78-95246c9bb590_1735543766152", "transaction_id": "6fa0d53c-e225-45bd-8a78-95246c9bb590", "description": "Ký thành công", "status": "1" }, "data": { "action_timestamp": 1735543766, "signature_timestamp": 1735543766, "results": [{ "document_id": "123asdasd", "signature": "dQA1ZLHKM+9qJicJ9XgzlFwi56Gq19q+guok5nSaGTfpXuBnQRwosPHGKZm7 vs6S\r\n8jX2VxY0etyaVmJW8osDFb83oxvdtPIw+fY3ItxQir80YT3+Hc8ZfRBFInu 07oja\r\nizirsTz2P6SDm37a+JplPgBt5NBW3T5gACbkZlZAWRx5RE5nkbCELi3p HKOZnJNd\r\nCobzBD/Kc7yKuJB6e6CL25A8z3Zli+F3EJcF5jyw4b7wsu1/17tWM"</pre>

	<pre> WMEDotCYYLB\r\ndwTYaHRz7UiQhLLoN960LaunFmc0TLhN8KD7Jhu3w6rmJ Rhmf29s3zq/PNFibwff\r\nb9CJALmsIy69nLiTICO4Q==" }] } } </pre>
Response Header	
authentication_methods	true

- **Response:**

Status Code	Message	Response Body
200	OK	<pre> { "status_code": "00", "message": "Thành công" } </pre>
500	Internal Server Error	Message lỗi

Request Parameters

Parent Parameter	Parameters	Presence	Value	Description
meta_data		MANDATORY	<i>Object</i>	
meta_data	request_id	MANDATORY	<i>String</i>	Do đối tác cung cấp
meta_data	transaction_id	MANDATORY	<i>String</i>	Do đối tác cung cấp
meta_data	description	MANDATORY	<i>String</i>	Mô tả người ký
meta_data	status	MANDATORY	<i>String</i>	Trạng thái
data		MANDATORY	<i>Object</i>	
data	action_timestamp	MANDATORY	Timestamp	Thời điểm xác thực yêu cầu ký (Dựa vào Unix time second)

data	signature_timestamp	MANDATORY	Timestamp	Thời điểm ký thành công (Dựa vào Unix time second)
data	results (List)	MANDATORY	<i>Object</i>	Kết quả chữ ký trả về
results (List)	document_id	MANDATORY	<i>String</i>	Mã ID document ký
results (List)	signature	MANDATORY	<i>String</i>	Chữ ký dạng base64 tương ứng với Document ID trên

Bảng mã trạng thái trường meta_data.status (trạng thái yêu cầu ký)

meta_data.status	Mô tả
1	success
4000	Chờ người dùng xác nhận
6000	Yêu cầu đã được người dùng xác nhận, hệ thống đang ký
4001	Yêu cầu ký hết thời gian chờ (Timeout)
4002	Người dùng từ chối ký
4003	Chờ hệ thống sinh chữ ký
4004	Ký thất bại (có lỗi xảy ra)
4005	Tài khoản không đủ số dư/lượt ký
4006	Chứng thư số chưa nghiệm thu
4007	Chờ hệ thống ghép chữ ký vào file
4008	Thuê bao không hợp lệ
13003	Chứng thư số hết hạn
13004	Chứng thư số không hợp lệ
50000	Có lỗi xảy ra khi lấy thông tin

Response Parameters

Parent Parameter	Parameters	Presence	Value	Description
	status_code	MANDATORY	String	Mã lỗi của request
	message	MANDATORY	String	Mô tả mã lỗi request

Bảng mã lỗi

Hệ thống Mysign chấp nhận mã lỗi 00 là yêu cầu thực hiện thành công.

status_code	message	Mô tả
00	Thành công	Ghi nhận kết quả ký đã trả về thành công
Khác 00	Bên thứ 3 trả mô tả lỗi ở trường message	Không thành công, dừng luồng callback trả kết quả

4. Danh sách mã lỗi chung

Mã lỗi	Mô tả lỗi
58001	An internal server error occurred while processing the request – see the RAS service debug logs for details
58002	Service is not available - Try later
58003	Failed to process request - RAS service is not enabled in license
58004	Failed to process request - RAS service license has expired
58005	Failed to process request - RAS service is not enabled in system
58006	Failed to process request - RAS service is not allowed
58007	Failed to process request - Client ID does not exist
58008	Failed to process request - User ID does not exist

Mã lỗi	Mô tả lỗi
58009	Failed to process request - User ID already exists
58010	Failed to process request - Key alias does not exist
58011	Failed to process request - Transaction ID does not exist
58012	Failed to process request - Client ID not found in the request
58013	Failed to process request - User ID not found in the request
58014	Failed to process request - Key alias not found in the request
58015	Failed to process request - Subject DN not found in the request
58016	Failed to process request - User password not found in the request
58017	Failed to process request - Key length not found in the request
58018	Failed to process request - Key algorithm not found in the request
58019	Failed to process request - User name not found in the request
58020	Failed to process request - User password not found in the request
58021	Failed to process request - User mobile number not found in the request

Mã lỗi	Mô tả lỗi
58022	Failed to process request - Key alias exceeds the allowed limit
58023	Failed to process request - User ID exceeds the allowed limit
58024	Failed to process request - User name exceeds the allowed limit
58025	Failed to process request - User password exceeds the allowed limit
58026	Failed to process request - Invalid user mobile number
58027	Failed to process request - Invalid user email
58028	Failed to process request - Invalid user status
58029	Failed to process request - RAS profile does not exist or marked inactive
58030	Failed to process request - User certificate not found in the request
58031	Failed to process request - Profile ID not found in the request
58032	Failed to process request - Invalid client ID
58033	Failed to process request - User's new password not found in the request

Mã lỗi	Mô tả lỗi
58034	Failed to process request - SMS OTP not found in the request
58035	Failed to process request - Email OTP not found in the request
58036	Invalid string parameter - refresh_token
58037	Failed to process request - Invalid refresh token
58038	Failed to process request - Invalid access token
58039	The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed.
58040	Failed to process request - Basic authentication is not enabled in RAS profile
58041	Failed to process request - SAML authentication is not enabled in RAS profile
58042	Failed to process request - Missing (or invalid type) string parameter token
58043	Failed to process request - Invalid string parameter token_type_hint
58044	Failed to process request - Invalid string parameter token
58045	Failed to process request - Client ID is not configured for certification service settings in RAS service manager

Mã lỗi	Mô tả lỗi
58046	Failed to process request - Certification profile is not configured for certification service settings in RAS service manager
58047	Failed to process request - Certification service address is not configured for certification service settings in RAS service manager
58048	Failed to process request - Unable to get device certificate from certification service
58049	Failed to generate access token - HMAC Key not configured in RAS service manager
58050	Failed to process request - User Email not found in the request
58051	Failed to process request - Client ID is not configured for default settings in RAS service manager
58052	Failed to process request - Device ID not found in the request
58053	Failed to process request - Push notification token not found in the request
58054	Failed to process request - OS type not found in the request
58055	Missing (or invalid type) string parameter credentialID
58056	Missing (or invalid type) integer parameter numSignatures

Mã lỗi	Mô tả lỗi
58057	Invalid parameter numSignatures
58058	Invalid request parameter - numSignatures doesn't match with no of hashes in hash array
58059	Invalid request parameter - no of documents in clientData doesn't match with no of hashes in hash array
58060	Missing parameter hash
58061	Failed to authorise user credentials - Request timeout for mobile authorisation
58062	Failed to authorise user credentials - User cancelled mobile authorisation
58063	Invalid parameter credentialID
58064	Missing (or invalid type) string parameter SAD
58065	Invalid parameter SAD
58066	Empty hash array
58067	Invalid Base64 hash string parameter
58068	Missing (or invalid type) string parameter signAlgo
58069	Invalid parameter signAlgo
58070	Missing (or invalid type) string parameter hashAlgo
58071	Invalid parameter hashAlgo

Mã lỗi	Mô tả lỗi
58072	Failed to validate SAML assertion - Invalid base64 data
58073	Failed to validate SAML assertion - Not comply with SAML 2.0 schema
58074	Failed to validate SAML assertion - Unable to parse SAML assertion
58075	Failed to validate SAML assertion - Validity period expired or not yet valid
58076	Failed to validate SAML assertion - Server certificate does not match with the certificate configured in RAS Profile
58077	Failed to validate SAML assertion - Multiple or no attributeValue found
58078	Failed to validate SAML assertion - Invalid Signature
58079	Failed to process request - User status is blocked or inactive
58080	Failed to process request - Certificate chain not found in the request
58081	Failed to process request - Invalid certificate chain
58082	Failed to process request - Client secret not found in the request
58083	Failed to authorise user credentials - An internal server error occurred during signature computation

Mã lỗi	Mô tả lỗi
58084	Failed to process request - Device CSR not found in the request
58085	Failed to process request - Invalid device CSR
58086	Failed to process request - Device information not found in the request
58087	Failed to process request - Device ID not found in the request
58088	Failed to process request - Device name not found in the request
58089	Failed to process request - SAD not found in the request
58090	Failed to process request - Request ID not found in the request
58091	Failed to process request - Invalid request
58092	Failed to process request - Either request ID is invalid or the transaction is expired
58093	An internal server error occurred - please contact your service provider
58094	Failed to process request - User mobile exceeds the allowed limit

Mã lỗi	Mô tả lỗi
58095	Failed to process request - User email exceeds the allowed limit
58096	Failed to process request - Configurations for SMS/Email OTP(s) not available
58097	Failed to process request - No OTP(s) found in request
58098	Failed to process request - QR Code authentication is not allowed for this RAS profile
86000	Failed to authenticate client - TLS client authentication certificate has expired
86001	Failed to authenticate client - TLS certificate CN does not match with Client ID
86002	Failed to authenticate client - TLS client certificate is revoked
86003	Failed to authenticate client - revocation status for TLS client certificate is unknown
86004	Failed to authenticate client - Client ID does not match with the client identified by TLS client certificate
86005	Failed to authenticate client - TLS client certificate does not match with the configured client certificate
86006	Failed to authenticate client - request signing certificate has expired

Mã lỗi	Mô tả lỗi
86007	Failed to authenticate client - request signing certificate is revoked
86008	Failed to authenticate client - revocation status for request signing certificate is unknown
86009	Failed to authenticate client - request signing certificate does not match with the configured client certificate
86010	Failed to authenticate client - Client ID does not match with the client identified by the request signing certificate
86011	Failed to authenticate client - Client ID does not exist
86012	An error occurred while communicating with database - see the service debug logs for details
86013	An error occurred when checking the certificate revocation status see the service debug logs for details
86014	An internal error occurred while authenticating the client - see the service debug logs for details
86015	Failed to authenticate client - Client ID is not found in the request
86016	Failed to process request - Request signing certificate is not trusted
86017	Failed to authenticate client - client is marked inactive

Mã lỗi	Mô tả lỗi
86018	Failed to authorise client - service is not allowed to this client
86019	Failed to authorise client - service profile does not exist
86020	Failed to authorise client - service profile is inactive
86021	Failed to authorise client - profile is not allowed to this client
86022	Failed to authorise client - default profile not configured and neither found in request
86023	Failed to authorise client - default profile is inactive
86024	Failed to authorise client - client secret is invalid
internal_error	An internal server error occurred while processing the request
invalid_csr	CSR is invalid
invalid_otp	OTP is either invalid or expired
missing_csr	CSR is missing in the request
missing_device_id	Device ID is missing in the request
missing_device_info	Device information is missing in the request

Mã lỗi	Mô tả lỗi
missing_device_name	Device name is missing in the request
missing_request_id	Request ID is missing in the request
refresh_token_revoked	Refresh token is either invalid or expired

Phụ lục 2: API Cloud-CA dành cho mobile app xác thực ký

1. Xác thực Client

API xác thực client_id, client_secret là các thông tin xác thực của client được hệ thống Mysign cấp. Sau khi xác thực thành công, hệ thống trả lại mã access_token xác thực cho các API 1.2 và 1.3. (application_access_token)

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authenticate	
HTTP Verb	POST
Content-Type	application/x-www-form-urlencoded
Accept	application/json
Authorization	
Request Body	client_id=samples_test_client & client_secret=121212 & grant_type=client_credentials
Response Header	
authentication_methods	true

Response

Status Code	Message	Response Body
200	OK	{ "access_token": "2YotnFZFEjr1zCsicMWpAA", "expires_in": 3600 }
400	Bad Request	Tham khảo tài liệu OAuth RFC 6749

Request Parameters

client_id	MANDATORY	String	Thông tin client_id (mã ID của client) được cấp bởi hệ thống Mysign
client_secret	MANDATORY	String	Thông tin client_secret (mã bí mật của client) được cấp bởi hệ thống Mysign

grant_type	MANDATORY	String	Kiểu xác thực. Mặc định: client_credentials
------------	-----------	--------	---

Response Parameters

access_token	MANDATORY	String	Mã access_token được sử dụng trong các API 1.2, 1.3
expires_in	MANDATORY	String	Thời gian hết hạn của mã

2. Xác thực User

Xác thực user có tồn tại trên hệ thống hay không. Nếu có thông tin, một mã OTP sẽ được trả về

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/user/enrol	
HTTP Verb	POST
Content-Type	application/json
Accept	application/json
Authorization	Bearer {application_access_token} --- Token lấy được từ hàm 1.1
Request Body	{ "user_id": "John_Doe", }
Response Header	
authentication_methods	true

Response

Status Code	Message	Response Body
-------------	---------	---------------

200	OK	<p>Nếu OTP được gửi tới email và SĐT của user</p> <pre>{ "auth_type": "OTP", "access_token": "eyJhbGciOiJI...FdR_3eyoY4", "token_type": "bear", "expires_in": 3600, "otp_info": [{ "otp_type": "EMAIL_OTP", "sent_to": "john.doe@sample.som" }, { "otp_type": "SMS_OTP", "sent_to": "+448007720442" }] }</pre> <p>Nếu OTP chỉ được gửi tới email của user:</p> <pre>{ "auth_type": "OTP", "access_token": "eyJhbGciOiJI...FdR_3eyoY4", "token_type": "bear", "expires_in": 3600, "otp_info": [{ "otp_type": "EMAIL_OTP", "sent_to": "john.doe@sample.som" }] }</pre>
-----	----	--

		Nếu OTP chỉ được gửi tới SĐT của user: <pre>{ "auth_type": "OTP", "access_token": "eyJhbGciOiJI...FdR_3eyoY4", "token_type": "bear", "expires_in": 3600, "otp_info": [{ "otp_type": "SMS_OTP", "sent_to": "+448007720442" }] }</pre>
400	Bad Request	
500	Internal Server Error	

Request Parameters

Parameters	Presence	Value	Description
user_id	MANDATORY	String	Thông tin user_id (tên user Mysign) được cấp

Response Parameters

Parameters	Presence	Value	Description
auth_type	MANDATORY	String	Loại cấu hình OTP xác thực được cấp.
otp_info	CONDITIONAL	String	Thông tin loại hình OTP và địa chỉ gửi OTP (email/SĐT) được cấu hình trên user_id (chỉ có khi loại hình xác thực là OTP)
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

3. Xác thực OTP

Xác thực OTP đã gửi vào thông tin đăng ký của tài khoản (SMS, Email). Sau khi xác thực thành công, hệ thống trả về mã access_token và refresh_token để thực hiện các giao dịch tiếp theo.

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authentication/otp/verify	
HTTP Verb	POST
Content-Type	application/json
Accept	application/json
Authorization	Bearer {application_access_token}
Request Body	<p>Nếu người dùng nhận 2 OTP:</p> <pre>{ "user_id": "User ID", "otp_info": [{ "otp": "258456987", "otp_type": "SMS_OTP" }, { "otp": "258456987", "otp_type": "EMAIL_OTP" }] }</pre>
	<p>Nếu người dùng nhận 1 OTP trên SĐT:</p> <pre>{ "user_id": "User ID", "otp_info": [{ "otp": "258456987", "otp_type": "SMS_OTP" }] }</pre>

Nếu người dùng nhận 1 OTP từ Email:

```
{
  "user_id": "User ID",
  "otp_info": [
    {
      "otp": "258456987",
      "otp_type": "EMAIL_OTP"
    }
  ]
}
```

Response

Status Code	Message	Response Body
200	Ok	{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5XmJGp-E", "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5XmJGp-E", "token_type": "bearer", "expires_in": 3600 }
400	Bad Request	
401	Unauthorized	
403	Forbidden	
500	Internal Server Error	

Request Parameters

Parameters	Presence	Value	Description
user_id	MANDATORY	String	Tên tài khoản/ID user Mysign
otp_info	MANDATORY	JSON Object	Object thông tin OTP nhận được
otp	MANDATORY	String	Mã OTP nhận được

otp_type	MANDATORY	String	Loại OTP nhận được (EMAIL_OTP/SMS_OTP)
----------	-----------	--------	--

Response Parameters

Parameters	Presence	Value	Description
access_token	MANDATORY	String	Mã access_token dùng để xác thực các API tiếp theo. Mã gắn với thông tin user_id đã xác thực
refresh_token	MANDATORY	String	Refresh token dùng để làm mới mã access_token ở trên mà không cần xác thực lại (xem API 1.4 trong tài liệu này)
expires_in	MANDATORY	String	Thời gian hết hạn của mã access_token
error	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

4. Renew Access Token

Sử dụng refresh_token ở mục 1.3 để cung cấp 1 access_token mới, không cần phải xác thực lại các bước 1.1-1.3.

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authenticate	
HTTP Verb	POST
Content-Type	application/x-www-form-urlencoded
Accept	application/json
Request Body	grant_type=refresh_token&refresh_token=tGzv3JOkF0XG5Qx

Response

Status Code	Message	Response Body
200	Ok	{ "access_token": "2YotnFZFEjr1zCsicMWpAA", "refresh_token": "TRVfHThcedfJGJFLGKKJ", "expires_in": 3600,

		}
400	Bad Request	

Request Parameters

Parameters	Presence	Value	Description
grant_type	MANDATORY	String	Loại mã sử dụng để refresh. Mặc định : refresh_token
refresh_token	MANDATORY	String	Mã refresh_token được cấp (xem response API 1.3 trong tài liệu này)

Response Parameters

Parameters	Presence	Value	Description
access_token	MANDATORY	String	Mã access_token mới
refresh_token	MANDATORY	String	Mã refresh_token mới
expires_in	MANDATORY	String	Thời gian hết hạn mã access_token.
error	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

5. Danh sách thiết bị đã được đăng ký (List Registered Device)

Trả về danh sách thiết bị đã đăng ký làm thiết bị xác thực của tài khoản

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/devices?user_id={user_id}	
HTTP Verb	GET
Accept	application/json
Authorization	Bearer {application_access_token} --- Token lấy được từ hàm 1.1
Request Body	

Response

Status Code	Message	Response Body
200	OK	[{ "device_id": "id-001", "device_name": "iPhone", "secure_element": true, "biometric": true }, { "device_id": "id-002", "device_name": "Samsung", "secure_element": true, "biometric": true }]
400	Bad Request	
500	Internal Server Error	

Request Parameters

Parameters	Presence	Value	Description
user_id	MANDATORY	String	Thông tin tên tài khoản/ID User Mysign cần lấy danh sách thiết bị đã đăng ký

Response Parameters

Parameters	Presence	Value	Description
device_id	MANDATORY	String	ID của thiết bị.
device_name	MANDATORY	String	Tên alias của thiết bị
secure_element	MANDATORY	String	“True” nếu thiết bị chứa các yếu tố bảo mật
biometric	MANDATORY	String	“True” nếu thiết bị hỗ trợ tính năng xác thực bằng sinh trắc học
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

6. Xóa thiết bị khỏi danh sách đăng ký

Đăng xuất thiết bị/bỏ thiết bị khỏi danh sách thiết bị bảo mật (xem API 5 trong Phụ lục)

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/devices/{device_id}	
HTTP Verb	DELETE
Accept	application/json
Access Token	Bearer {access_token} --- Token lấy được từ hàm 1.3 khi bật xác thực OTP
Request Body	

Response

Status Code	Message	Response Body
200	OK	
404	Not Found	
403	Forbidden	
500	Internal Server Error	

Request Parameters

Parameters	Presence	Value	Description
{device_id}	MANDATORY	String	Thông tin Device ID được đăng ký trên hệ thống Mysign (Xem Response API 1.5)

Response Parameters

Parameters	Presence	Value	Description
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

7. Lấy danh sách yêu cầu xác thực đang chờ (Get Pending Authorization Request)

Lấy danh sách yêu cầu cần xác thực đang trong trạng thái chờ. Yêu cầu xác thực được gửi từ các API gửi yêu cầu ký của người dùng (tham khảo tài liệu API cho người dùng)

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/request/list	
HTTP Verb	GET
Accept	application/json
Authorization	Bearer {access_token} --- Token lấy được từ hàm 1.3 khi bật xác thực OTP
Request Body	

Response

Status Code	Message	Response Body
200	OK	[{ "transaction_id": "932469001521668267", "request": "PEFDRj48Y2VydEFs[...]9BQ0Y+", "hash_algorithm": "SHA256"

		}, {...}]
400	Bad Request	
500	Internal Server Error	

Response Parameters

Parameters	Presence	Value	Description
transaction_id	MANDATORY	String	Mã ID giao dịch
request	MANDATORY	String	<p>Yêu cầu xác thực được mã hóa base64. Định dạng dữ liệu theo dạng xml bao gồm các thông tin của yêu cầu ký, theo mẫu như sau :</p> <pre> <AuthorisationData> <OriginatorID>Virtual_CSP_Client</OriginatorID> <UserID>olcayatli@gmail.com</UserID> <CertificateID>416edc72-6c63-45aa-bb34a373102234df</CertificateID> <TransactionID>980551837300673581</TransactionID> <Salt>924552495291565632</Salt> ADSS RAS - Developers Guide Ascertia Limited Commercial-in-Confidence Page 85 of 102 <MetaData> <DisplayText>Data to be displayed</DisplayText> <DeviceID></DeviceID> </MetaData> <Documents> </pre>

			<pre> <Document id="b81e040a-a4d8-4134-92ff-2d4bf5e9116d"> <Name>b81e040a-a4d8-4134-92ff-2d4bf5e9116d</Name> <DigestValue>ypkP9L2tZO2Jdf Nr4X4X5SRur529uJqykdc5q 5HDSiLNiYcLrysO0S/H31yb8QZS&# xD;SOBYsFIVSj9/SKUqrh sUC5oEc/gr</DigestValue> </Document> </Documents> <ValidityPeriod> <ValidFrom>2019-12-07T18:25:37</ValidFrom> <ValidTo>2019-12-07T18:42:17</ValidTo> </ValidityPeriod> <Signature> <DigestMethod>SHA256</DigestMethod> </AuthorisationData> </pre>
hash_algorithm	MANDATORY	String	Thuật toán hash được sử dụng để ký
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

8. Lấy yêu cầu xác thực đang chờ sử dụng transaction_id (Get Pending Authorization Request)

Lấy yêu cầu cần xác thực đang trong trạng thái chờ, sử dụng transaction_id đã có (được cung cấp khi gửi yêu cầu qua API SignHash dạng bất đồng bộ). Yêu cầu xác thực được gửi từ các API gửi yêu cầu ký của người dùng (tham khảo tài liệu API cho người dùng)

Request

<a href="https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/request/<transaction_id>">https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/request/<transaction_id>	
HTTP Verb	GET
Accept	application/json
Authorization	Bearer {access_token} --- Token lấy được từ hàm 1.3 khi bật xác thực OTP
Request Body	

Response

Status Code	Message	Response Body
200	OK	[{ "transaction_id": "932469001521668267", "request": "PEFDRj48Y2VydEFs[...]9BQ0Y+", "hash_algorithm": "SHA256" }]
400	Bad Request	
500	Internal Server Error	

Response Parameters

Parameters	Presence	Value	Description
transaction_id	MANDATORY	String	Mã ID giao dịch

request	MANDATORY	String	<p>Yêu cầu xác thực được mã hóa base64. Định dạng dữ liệu theo dạng xml bao gồm các thông tin của yêu cầu ký, theo mẫu như sau :</p> <pre> <AuthorisationData> <OriginatorID>Virtual_CSP_Client</OriginatorID> <UserID>olcayatli@gmail.com</UserID> <CertificateID>416edc72-6c63-45aa-bb34a373102234df</CertificateID> <TransactionID>980551837300673581</TransactionID> <Salt>924552495291565632</Salt> ADSS RAS - Developers Guide Ascertia Limited Commercial-in-Confidence Page 85 of 102 <MetaData> <DisplayText>[REV:hash_code][ID:transaction_id]</DisplayText> <DeviceID></DeviceID> </MetaData> <Documents> <Document id="b81e040a-a4d8-4134-92ff-2d4bf5e9116d"> <Name>b81e040a-a4d8-4134-92ff-2d4bf5e9116d</Name> <DigestValue>ypkP9L2tZO2JdfNr4X4X5SRur529uJqykdc5q5HDSiLNiYcLrysO0S/H31yb8QZS&#xD;SOBYsFIVSj9/SKUqrhsUC5oEc/gr</DigestValue> </Document> </Documents> <ValidityPeriod> <ValidFrom>2019-12-07T18:25:37</ValidFrom> </pre>
---------	-----------	--------	--

			<ValidTo>2019-12-07T18:42:17</ValidTo> </ValidityPeriod> <Signature> <DigestMethod>SHA256</DigestMethod> </AuthorisationData>
hash_algorithm	MANDATORY	String	Thuật toán hash được sử dụng để ký
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

9. Hủy bỏ yêu cầu xác thực (Cancel a Pending Authorisation Request)

Hủy yêu cầu xác thực trong trạng thái chờ được chỉ định

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/request/{request_id}	
HTTP Verb	DELETE
Authorization	Bearer {access_token} --- Token lấy được từ hàm 1.3 khi bật xác thực OTP
Accept	application/json
Request Body	

Response

Status Code	Message	Response Body
200	OK	
400	Bad request	
500	Internal Server Error	

Request Parameters

Parameters	Presence	Value	Description
{request_id}	MANDATORY	String	ID của yêu cầu muốn hủy (xem transaction_id tại API 1.7)

Response Parameters

Parameters	Presence	Value	Description
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

10. Thông tin tài khoản (Users Profile)

Thông tin của tài khoản

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/users/profile	
HTTP Verb	GET
Content-Type	application/json
Accept	application/json
Authorization	Bearer {access_token} --- Token lấy được từ hàm 1.3 khi bật xác thực OTP

Response

Status Code	Message	Response Body
200	OK	{ "user_id": "Alice", "user_name": "Alice", "app_name": "samples_test_client", "user_email": "abc@ascertia.com", "user_mobile": "+9230XXXXXXXXX" }
400	Bad Request	
500	Internal	

	Server Error	
--	--------------	--

Response Parameters

Parameters	Presence	Value	Description
user_id	MANDATORY	String	Thông tin tài khoản/ID user Mysign
user_name	MANDATORY	String	Tên người dùng tài khoản
App_name	CONDITIONAL	String	Tên ứng dụng gắn theo tài khoản
user_email	MANDATORY	String	Email của tài khoản
user_mobile	MANDATORY	String	SĐT tài khoản
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả mã lỗi

11. Đăng ký thiết bị để nhận notification

Đăng ký để thiết bị nhận notification

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/push/notification	
HTTP Verb	POST
Content-Type	application/json
Accept	application/json
Authorization	Bearer {access_token} --- Token lấy được từ hàm 1.3 khi bật xác thực OTP
Request Body	{ "device_token": "2YotnFZFEjr1zCsicMWpAA", "os_type": "ANDROID/IOS" }

Response

Status Code	Message	Response Body
200	OK	
401	Unauthorized	Invalid or expired user access token
400	Bad Request	Device token is missing
500	Internal Server Error	

Request Parameters

Parameters	Presence	Value	Description
device_token	MANDATORY	String	Token của thiết bị dùng để đăng ký với server để nhận notification
os_type	MANDATORY	String	Android/iOS.

Response Parameters

Parameters	Presence	Value	Description
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả lỗi

12. Xóa thiết bị khỏi danh sách nhận notification

Loại bỏ thiết bị khỏi danh sách nhận notification (API 10 trong Phụ lục)

Request

https://remotesigning.viettel.vn/vtss/service/ras/v1/authorization/push/notification/{device_token}	
HTTP Verb	DELETE
Content-Type	application/json
Accept	application/json
Authorization	Bearer {access_token} --- Token lấy được từ hàm 1.3 khi bật xác thực OTP

Request Body	{ "device_token": "2YotnFZFEjr1zCsicMWpAA" }
--------------	--

Response

Status Code	Message	Response Body
200	OK	

Request Parameters

Parameter	Presence	Value	Description
device_token	MANDATORY	String	Token thiết bị đã dùng để đăng ký nhận notification

Response Parameters

Parameters	Presence	Value	Description
error_code	CONDITIONAL	String	Mã lỗi
error_description	CONDITIONAL	String	Mô tả lỗi

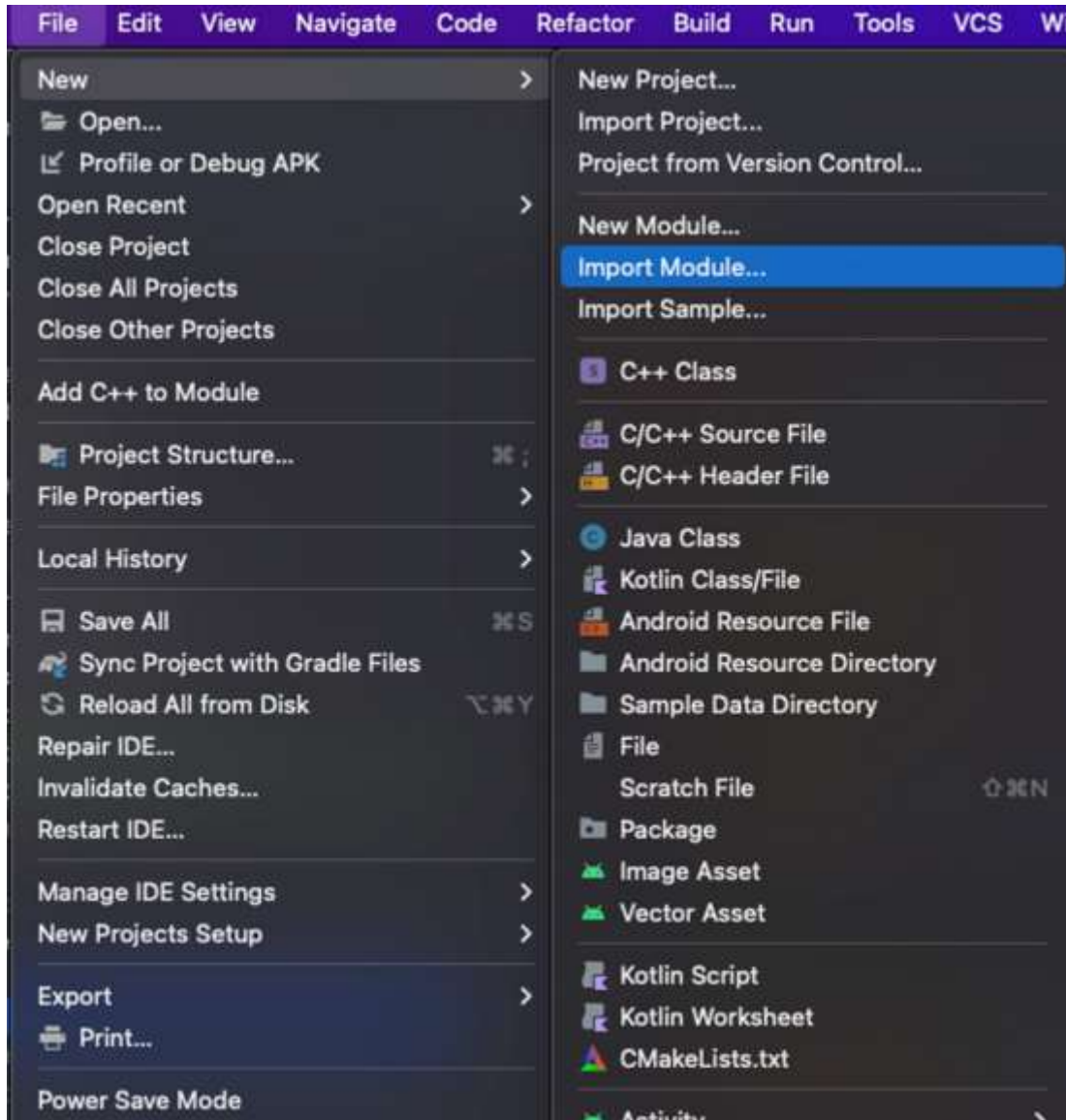
Phụ lục 3: Hướng dẫn tích hợp CloudCA SDK Lite trên Android

1. Hướng dẫn cài đặt

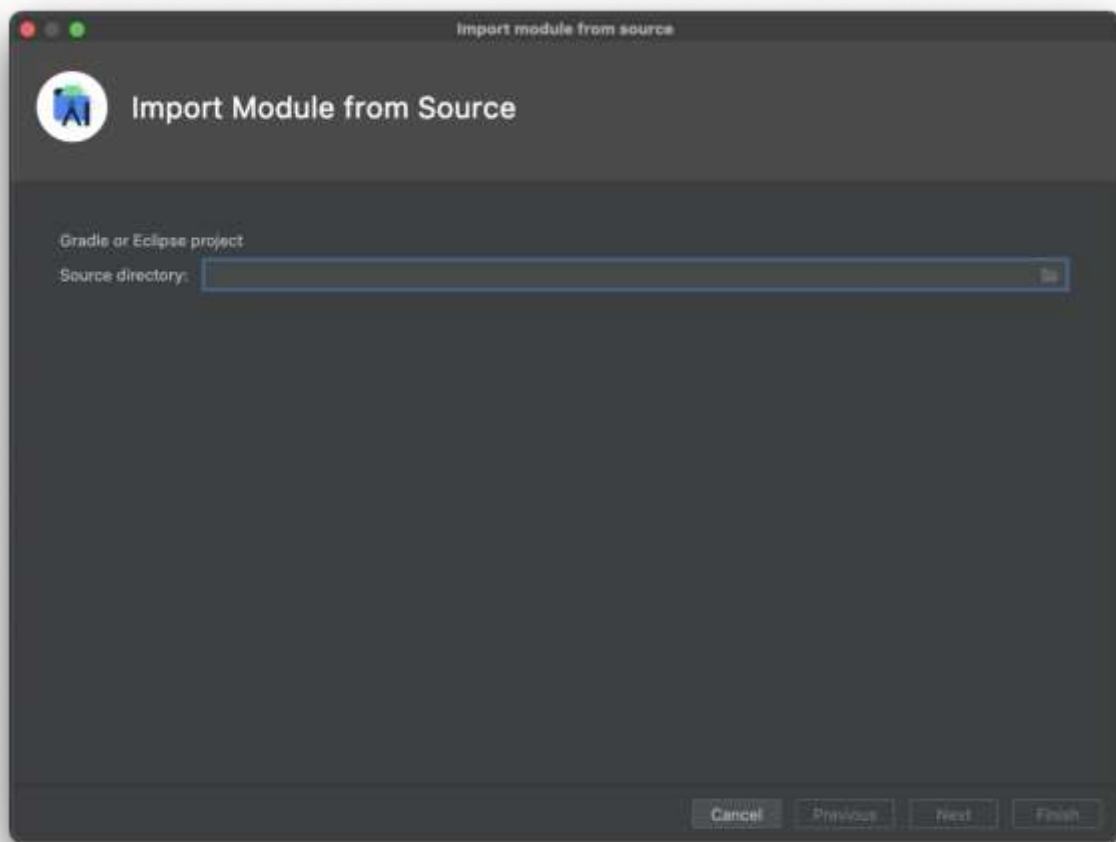
- Đưa file GoSignSDK.aar vào thư mục libs của dự án

Import SDK vào gradle app:

- Đường dẫn: File -> New -> Import module



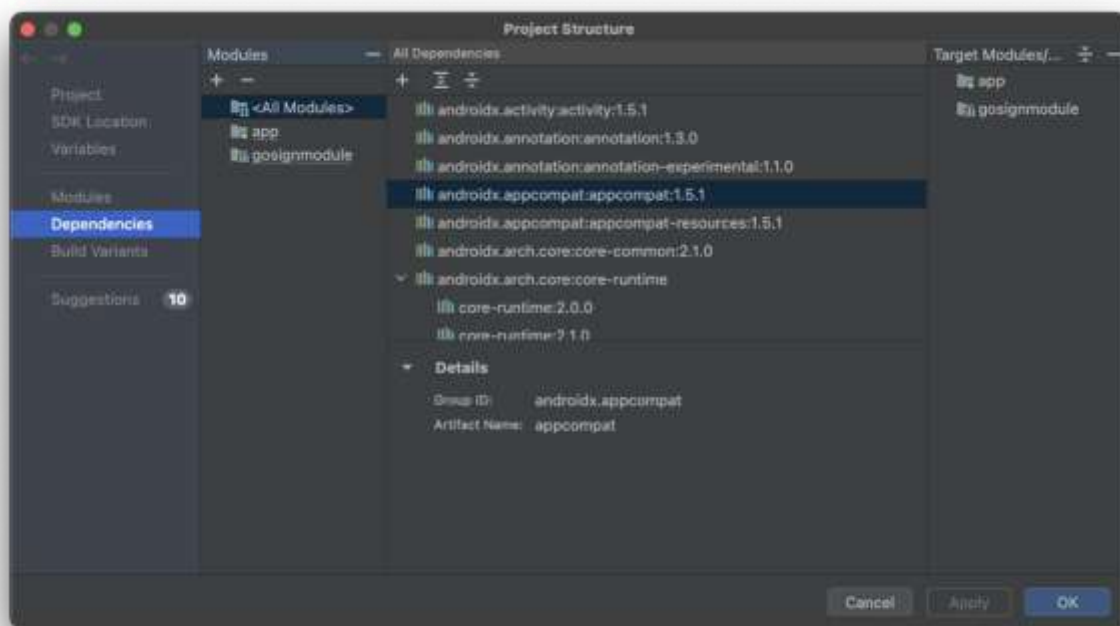
Chọn đường dẫn của SDK:



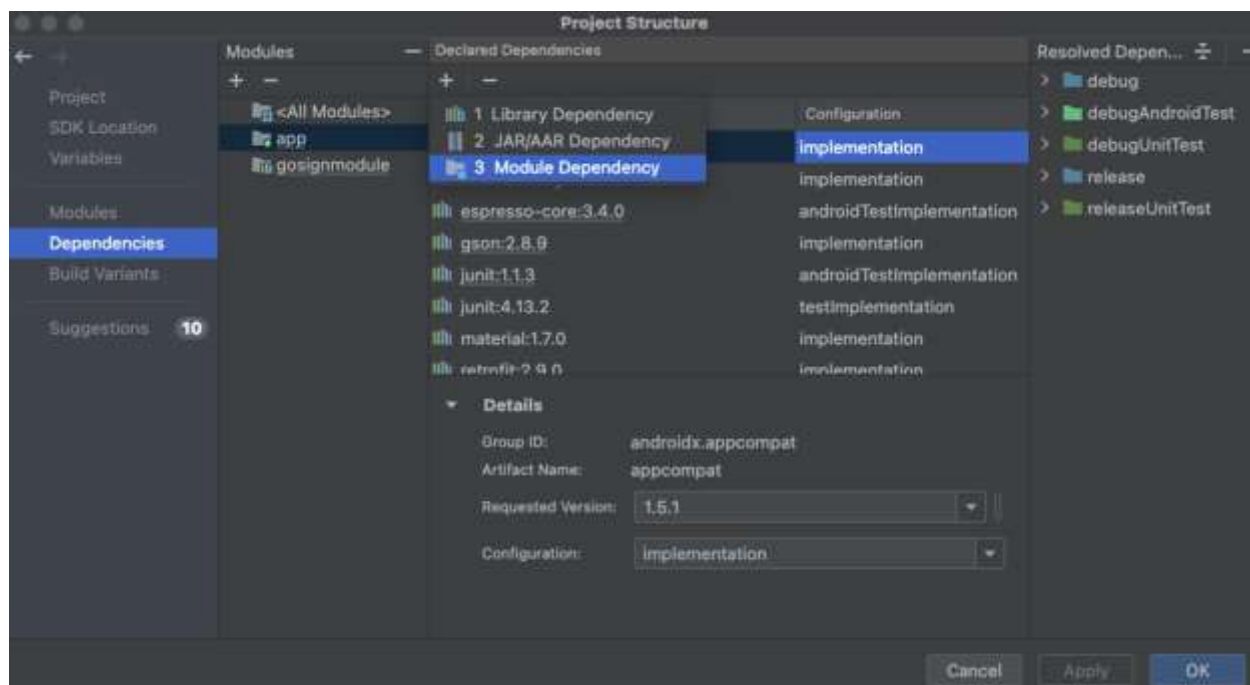
- Chọn đúng đường dẫn đến thư mục chứa module GoSignSDK
- Sau đó chọn “Finish”

Import thư viện vào gradle của project

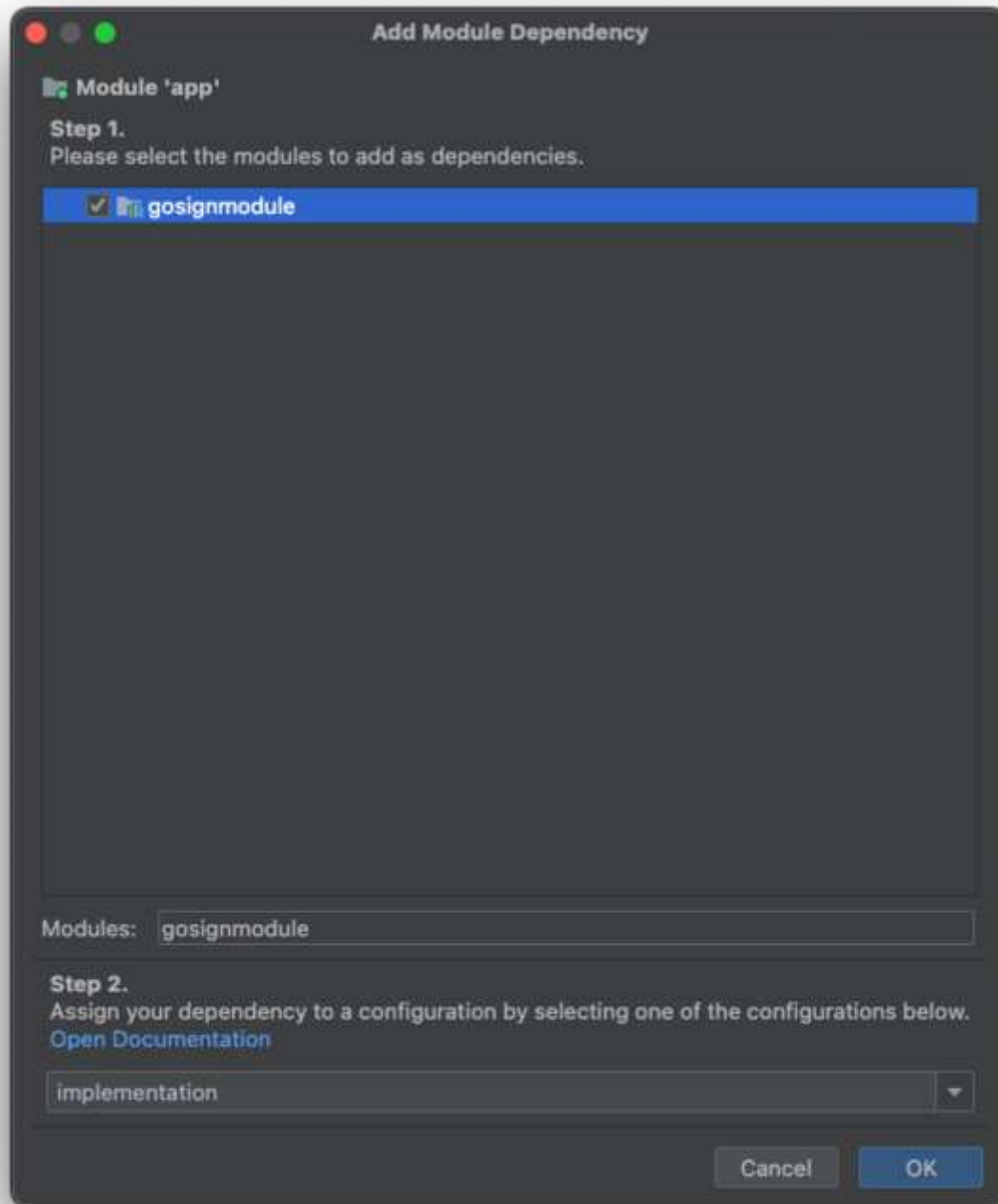
- Sau khi thêm module vào project, ta cần implement dependency để sử dụng các hàm trong đó, làm như hình sau.



- Đường dẫn: File -> Project structure -> Dependencies.
 - Mở dependencies lên, lúc này tùy từng project sẽ hiển thị khác nhau, dựa theo số modules được implement, chỉ quan tâm đến 1 mục duy nhất là “app”



- Chọn app -> nhấn vào biểu tượng “+” trong mục “Declared Dependencies” -
> Chọn vào tùy chọn thứ 3 “Module Dependency”



- Sau đó nhấn vào nút “OK”

Import dependencies:

// Network

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
implementation 'com.google.code.gson:gson:2.8.9'  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
implementation 'com.squareup.retrofit2:adapter-rxjava2:2.9.0'  
implementation 'com.squareup.okhttp3:okhttp:4.9.1'  
implementation 'com.squareup.okhttp3:logging-interceptor:4.9.1'  
implementation "io.reactivex.rxjava2:rxandroid:2.1.1"  
implementation "io.reactivex.rxjava2:rxjava:2.2.18"
```

// Spongy Castle

```
implementation "com.madgag.spongycastle:core:1.58.0.0"  
implementation "com.madgag.spongycastle:prov:1.58.0.0"  
implementation "com.madgag.spongycastle:bcpkix-jdk15on:1.58.0.0"
```

// Mockito

```
testImplementation "org.mockito:mockito-core:2.26.0"
```

// Biometric

```
implementation "androidx.biometric:biometric:1.1.0"  
implementation "dev.skomlach:biometric:2.1.52"
```

2. Cấu hình thêm vào class Application

- Application/Context: Nếu gửi vào 1 context, cần chắc chắn rằng context này sẽ không bị destroy trong quá trình sử dụng SDK ⇒ Tốt nhất là nên gửi vào application
- BaseURL: host (VD: <https://remotesigning.viettel.vn>)
- BiometricTitle: Tiêu đề của Dialog sinh trắc học
- VD: GoSignSDKSetup.initialize(application, baseURL, biometricTitle)

```
2 usages
public static void initialize(
    Application application,
    String baseURL,
    String biometricTitle
) {
    initialize(application, baseURL, biometricTitle);
}

public static void initialize(
    Context context,
    String baseURL,
    String biometricTitle
) {
```

3. Hướng dẫn sử dụng API

3.1. BaseModel

```
public interface ServiceApiListenerEmpty {
    void showLoading();
    void hideLoading();
    void onSuccess();
    void onFail(ResponseError error);
}
```

```
public interface ServiceApiListener<T> {  
    void showLoading();  
    void hideLoading();  
    void onSuccess(T data);  
    void onFail(ResponseError error);  
}
```

- **ResponseError**

- **ErrorType:**

- *BAD_SERVER_RESPONSE* : HTTP không trả lại mã 200
 - *SOME_THING_WENT_WRONG* : Lỗi trả về từ máy chủ.
 - *DECODING_ERROR* : Không thể phân tích cú pháp json.
 - *HARDWARE_ERROR* : Lỗi hệ thống nhận dạng.
 - *AUTHENTICATE_FAILURE* : Lỗi trả về khi xác thực

- **Biometric**

- **BiometricApiType:**

- *FACE_ID* ⇒ *Xác thực bằng khuôn mặt*
 - *FINGER_PRINT* ⇒ *Xác thực bằng vân tay*
 - *AUTO*
 - Kiểm tra phương thức xác thực hiện tại của thiết bị (*FACE_ID* => *FINGER_PRINT*)
 - Thỏa mãn phương pháp nào trước thì dùng phương pháp đó

- **Biometrics Error:**

- *HARDWARE_NON_SUPPORT* : Phần cứng không hỗ trợ.
 - *NO_ENROLLED*: Chưa đăng ký sinh trắc học.
 - *BIOMETRIC_SENSOR_TEMPORARY_LOCKED* : Sinh trắc học tạm thời bị khóa.
 - *BIOMETRIC_SENSOR_PERMANENTLY_LOCKED* : Sinh trắc học bị khóa vĩnh viễn.
 - *CANCELED* : *Xác thực đã bị hủy.*
 - *NO_BIOMETRICS_REGISTERED* : Người dùng chưa đăng ký bất kỳ dấu vân tay nào với hệ thống.

- **SENSOR_FAILED** : Cảm biến không thể đọc dấu vân tay, có thể do ngón tay di chuyển quá nhanh hoặc cảm biến bị bẩn.
- **TIMEOUT** :
 - Cảm biến đã chạy quá lâu mà không đọc được gì.
 - Khoảng thời gian mà cảm biến có thể chạy trước khi hết thời gian chờ là tùy theo hệ thống và cảm biến cụ thể, nhưng thường là khoảng 30 giây. Việc bắt đầu một lần xác thực khác ngay lập tức là an toàn.
- **AUTHENTICATION_FAILED** : Một dấu vân tay đã được đọc thành công, nhưng dấu vân tay đó chưa được đăng ký trên thiết bị.
- **UNKNOWN** : Xác thực không thành công vì một lý do không xác định.
- **INTERNAL_ERROR** : Lỗi API nội bộ.
- **NOT_INITIALIZED_ERROR** : API không được khởi tạo.
- **MISSING_PERMISSIONS_ERROR** : Không thể bắt đầu sinh trắc học do thiếu quyền.

3.2. Đăng ký thiết bị

- API này được sử dụng để đăng ký thiết bị di động của người dùng cho mục đích ủy quyền chữ ký từ xa và yêu cầu chứng chỉ cho khóa công khai ủy quyền của thiết bị.

```
/**
 * Call generate certificate according to default device settings
 * @param activityNeed current [activity] info to show biometric authentication view
 * @param tokenUse to add header when calling API
 * @param biometricApiTypeBiometric information needs to be authenticated
 * @param listenerReturns the result of biometric authentication and register device
 */
void registerDevice(FragmentActivity activity,
                    String token,
                    BiometricApiType biometricApiType,
                    ServiceApiListener<CertificateResponse> listener);
```

- Request: **activity, token, biometricType**

- Response: `ServiceApiListener<AuthUserResponse>`

```
new MySignSDK.Builder()
    .withUserId(UserId)
    .withToken(token)
    .withActivity((MainActivity) getContext())
    .withBiometricApiType(BiometricApiType.AUTO)
    .registerDevice(new ServiceApiListener<CertificateResponse>() {
        @Override
        public void onSuccess(CertificateResponse data) {
            // Do something
        }
        @Override
        public void onFail(ResponseError error) {
            // Do something
        }
        @Override
        public void showLoading() {
        }
        @Override
        public void hideLoading() {
        }
    })
    .build();
```

- Model

```
public class CertificateResponse {
    @SerializedName("alias")
    private String alias;

    @SerializedName("certificate")
    private String certificate;

    public String getAlias() {
        return alias;
    }

    public void setAlias(String alias) {
        this.alias = alias;
    }

    public String getCertificate() {
        return certificate;
    }

    public void setCertificate(String certificate) {
        this.certificate = certificate;
    }
}
```

3.3. Xác thực yêu cầu ký

Phương pháp này cho phép xác thực nhiều yêu cầu ký cùng một lúc.

- Xác thực nhiều đơn ký

Request: **activity**, **token**, **listPendingRequest**, **biometricType**, **listener**

- Model PendingAuthorisationRequest khởi tạo giống API 4.9

Response: **ServiceApiListenerEmpty**
- **AuthorisationResponse**

```
public class AuthorisationResponse {
    @SerializedName("success")
    private List<PendingAuthorisationRequest> success;

    @SerializedName("failed")
    private List<PendingAuthorisationRequest> failed;

    public List<PendingAuthorisationRequest> getSuccess() {
        return success;
    }
    public List<PendingAuthorisationRequest> getFailed() {
        return failed;
    }
}
```

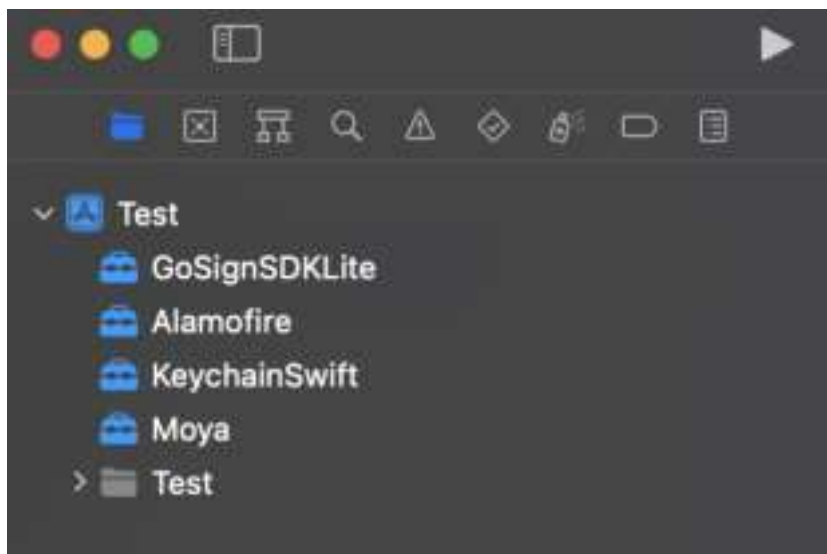
Request

```
new MySignSDK.Builder()
    .withUserId(UserId)
    .withToken(token)
    .withBiometricApiType(BiometricApiType.AUTO)
    .withActivity((MainActivity) getContext())
    .authorisationListPendingRequest(list, new ServiceApiListener<AuthorisationResponse>() {
        @Override
        public void onSuccess(AuthorisationResponse data) {
            // Do something
        }
        @Override
        public void onFail(ResponseError error) {
            // Do something
        }
        @Override
        public void showLoading() {
        }
        @Override
        public void hideLoading() {
        }
    })
    .build();
```

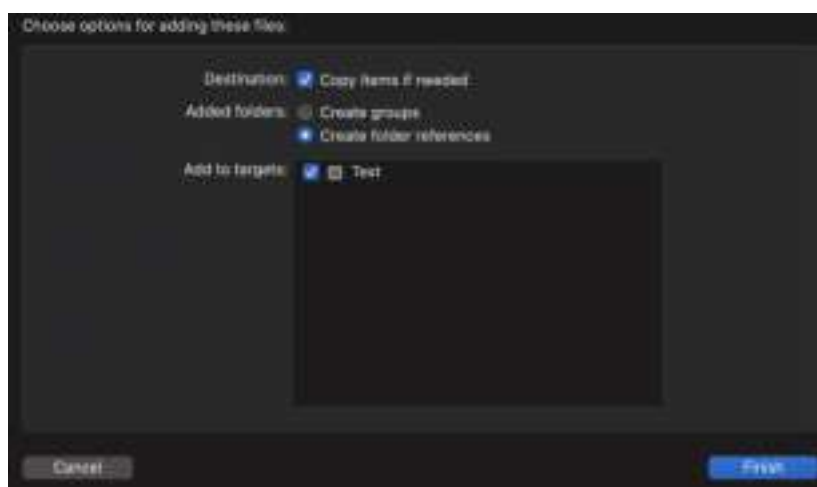
Phụ lục 4: Hướng dẫn tích hợp CloudCA SDK Lite trên iOS

1. Thêm thư viện CloudCA SDK

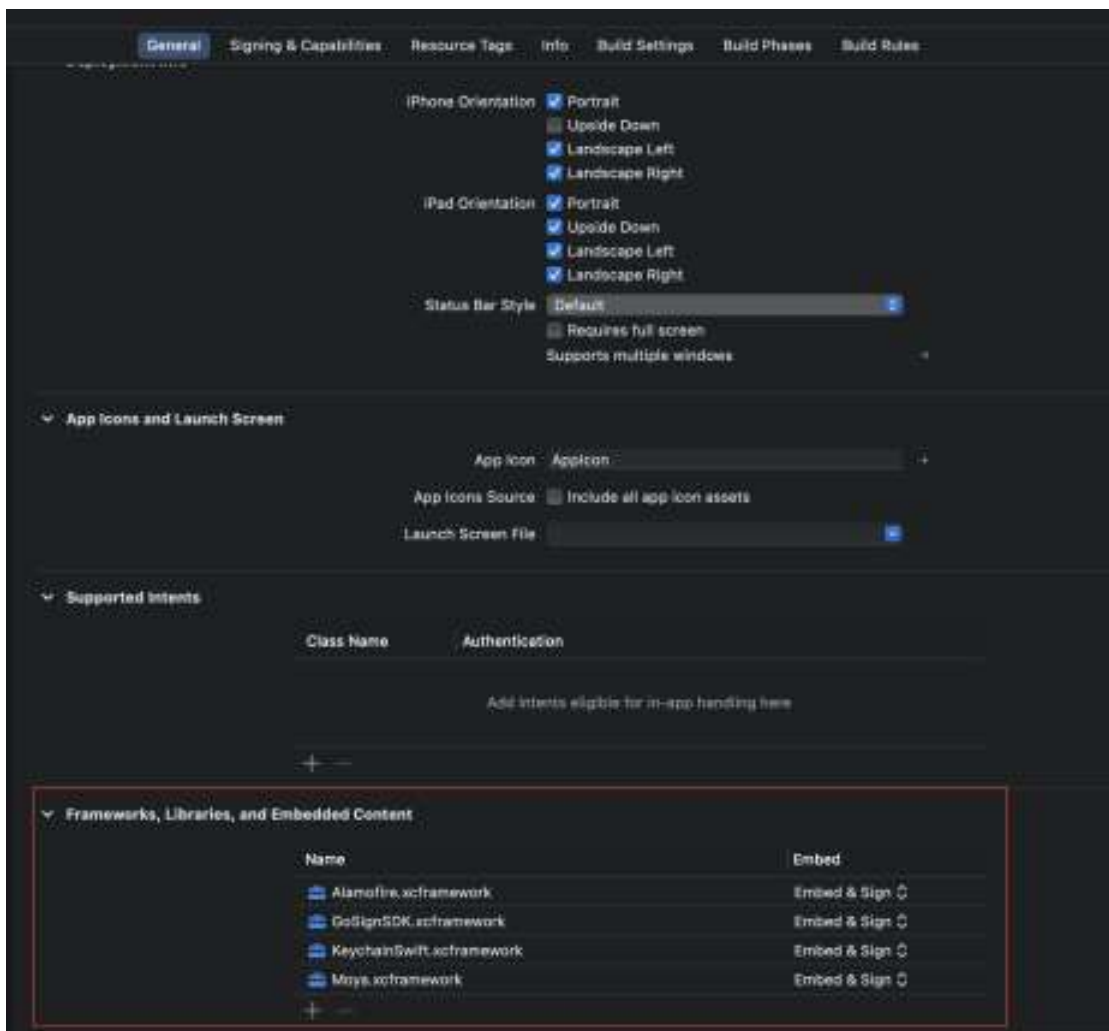
- Thực hiện kéo các file framework tới project navigator
GoSignSDKLite.xcframework, **Alamofire.xcframework**,
KeychainSwift.xcframework, **Moya.xcframework**
- Các file framework được gửi kèm với tài liệu hướng dẫn tích hợp



- Chọn “Copy items if needed”



- Trên tab General, mục Frameworks, Libraries, and Embedded Content chọn Embed & Sign



2. Thực hiện import

- Import gói GoSignSDKLite tới file cần cài đặt Swift

```
import UIKit
import GoSignSDKLite
```

Objective-C

```
#import "ViewController.h"
#import <GoSignSDKLite/GoSignSDKLite-Swift.h>
```

3. Thực hiện cài đặt Base URL, User ID cho SDK

- Base url, User ID sẽ được VTT cung cấp

- Thực hiện cài đặt Base url

- **Swift**

```
API.host = "base url"
```

- **Objective-C**

```
[API setHost:@"base url"];
```

- Thực hiện cài đặt User ID

- **Swift**

```
API.setUserId("user id")
```

- **Objective-C**

```
[API setUserId:@"user id"];
```

- Thực hiện cài đặt Device ID (Optional). Nếu app không truyền device id thì SDK sẽ tự động sinh ra device id.

- **Swift**

```
API.setDeviceId("device id")
```

- **Objective-C**

```
[API setDeviceId:@"device id"];
```

4. Thực hiện lấy Device ID, CSR

- Device ID

- **Swift**

```
let deviceId = API.getDeviceId()
```

- **Objective-C**

```
NSString *deviceId = [API getDeviceId];
```

- CSR

- **Swift**

```
let csr = API.getCSR()
```

Hoặc

```
let csr = API.getCSR(userId: "user id")
```

Hoặc

```
API.getCSR(userId: "user id", deviceId: "device id")
```

- **Objective-C**

```
NSString *csr = [API getCSR];
```

Hoặc

```
NSString *csr = [API getCSR:@"user id"];
```

Hoặc

```
NSString *csr = [API getCSR:@"user id" deviceId: @"device id"];
```

5. Thực hiện cài đặt API

5.1. API đăng ký thiết bị

- API này được sử dụng để đăng ký thiết bị di động của người dùng cho mục đích ủy quyền chữ ký từ xa và yêu cầu chứng chỉ cho khóa công khai ủy quyền của thiết bị.
- Khi thực hiện gọi API này, thiết bị bắt buộc phải cài đặt xác nhận vân tay hoặc khuôn mặt. Nếu thiết bị không có hoặc không cài đặt vân tay hoặc khuôn mặt thì sẽ trả về lỗi.
- **Swift**

```
API.registerDevice(authenToken: "authenToken") { response in
```

```
// response: Result<RegisterDeviceAPIResponse, Error>
```

```
switch response {
```

```
case .success(let result):
```

```
// handle response if needed
```

```
case .failure(let failure):
```

```
let error = failure as! ServerResponseError
```

```
let response = "Failure: \(error.message)"
```

```
// handle response if needed
```

```
}
```

```
}
```

- **Objective-C**

[API registerDeviceWithToken:@"authenToken" localizedReason:@"Unlock to add device"]

completion:^(RegisterDeviceAPIResponse * _Nullable response, NSError * _Nullable error) {

// handle registerDeviceAPIResponse or error if needed

});

- **Request**

authenToken: String: Token được trả về từ API 4.2 hoặc 4.3

localizedReason: String: Nội dung hiển thị trên popup xác nhận vân tay hoặc khuôn mặt

- **Model**

RegisterDeviceAPIResponse

Swift

```
public struct RegisterDeviceAPIResponse : Codable {  
    public let alias: String?  
    public let certificate: String?  
  
    /// Encodes this value into the given encoder  
    /// If the value fails to encode anything, `encoder` will encode an empty /  
    /// keyed container in its place.  
    /// This function throws an error if any values are invalid for the given  
    /// encoder's format.  
    /// - Parameter encoder: The encoder to write data to.  
    public func encode(to encoder: Encoder) throws  
  
    /// Creates a new instance by decoding from the given decoder.  
    /// This initializer throws an error if reading from the decoder fails, or  
    /// if the data read is corrupted or otherwise invalid.  
    /// - Parameter decoder: The decoder to read data from.  
    public init(from decoder: Decoder) throws  
}
```


Objective-C

```
@interface RegisterDeviceAPIResponse : NSObject
@property (nonatomic, readonly, copy) NSString * _Nullable alias;
@property (nonatomic, readonly, copy) NSString * _Nullable certificate;
@end
```

- **Response**

- Thành công

```
{
  "alias": "CN=4868DDB9-1421-4451-9F51-E78B476C570D",
  "certificate": "MIIEfzCCAzOgAwIBAgIUtwGusK8lu1LQwDbRMDLjFJ
    auo/AwQQYJKoZIhvcNAQEK\r\nMDSgDzANBgIlg }
```

- Thất bại

```
{
  "error": "58039",
  "error_description": "The request is missing a required parameter,
    includes an invalid parameter value, includes a parameter more than
    once, or is otherwise malformed."
}
```

error: Mã lỗi (Mô tả trong file danh sách mã lỗi)

error_description: Nội dung lỗi

5.2. API Xác thực yêu cầu ký

- API này cho phép có thể ký được nhiều uỷ quyền ký trong một lần.
- Khi thực hiện gọi API này, thiết bị bắt buộc phải cài đặt xác nhận vân tay hoặc khuôn mặt. Nếu thiết bị không có hoặc không cài đặt vân tay hoặc khuôn mặt thì sẽ trả về lỗi.

API.authoriseaPendingRequestList(authenToken: “authenToken”,

pendingAuthorisation: pendingAuthorisation) { response in //

response:

Result<AuthoriseaListPendingAPIResponse, Error>

```
switch response {
case .success(let result):
// handle response if needed
case .failure(let failure):
let error = failure as! ServerResponseError
let response = "Failure: \(error.message)"
// handle response if needed
}
}
```

- **Request**

- authToken:
- + Kiểu dữ liệu: String
- + Token được trả về từ API authenticate
- pendingAuthorisation:
- + Kiểu dữ liệu: Mảng object PendingAuthorisationAPIResponse
- + Danh sách các yêu cầu cần ký được lấy về từ API Danh sách chờ ký.

```
[
{
  "requestId": "",
  "request": "",
  "hash_algorithm": ""
},
{
  "requestId": "",
  "request": "",
  "hash_algorithm": ""
}
```

```
}
```

```
]
```

- **Model**

- PendingAuthorisationAPIResponse:

- + requestId: string

- + request: string

- + hashAlgorithm: string

- **Response**

- Thành công

```
{
```

```
"success":[],
```

```
"failed": [
```

```
"PEF1dGhvcmlzYXRpb25EYXRhPjxPcmlnaW5hdG9ySUQ+c2FtcGxlc1
```

```
90ZXN0X2NsaWVudDwvT3JpZ2luYXRvcklEP ]
```

```
}
```

success, failed: Danh sách các requestId (Theo request đầu vào).

- Thất bại

```
{
```

```
"error": "58039",
```

```
"error_description": "The request is missing a required parameter,  
includes an invalid parameter value, includes a parameter more  
than once, or is otherwise malformed."
```

```
}
```

error: Mã lỗi (Mô tả trong file danh sách mã lỗi)

error_description: Nội dung lỗi.

Phụ lục 5: Thông tin code demo, SDK Mysign

Code demo ký số theo dịch vụ Mysign: https://drive.google.com/file/d/1-4-bTcNsn4Z_UzMJW4zZKKlGMIZw8jPJ/view?usp=drive_link

SDK Mysign:

https://drive.google.com/drive/folders/1rNG1ynIU2cRiAheftaW4H4rao4gtj7Zr?usp=drive_link