

CorpID Sandbox Programme Simulated API Specifications

Version: 1.0

December 2025

© The Government of the Hong Kong Special Administrative Region
of the People's Republic of China

The contents of this document remain the property of and may not be reproduced in whole or in part without the express permission of the Government of the Hong Kong Special Administrative Region of the People's Republic of China

Amendment History			
Version No.	Release Date	Section Affected	Summary of Changes

CORPID SANDBOX PROGRAMME	1
SIMULATED API SPECIFICATIONS	1
1. OVERVIEW	10
1.1. HIGH LEVEL OVERVIEW.....	10
1.2. ENVIRONMENT	12
1.2.1. CorpID Sandbox Environment.....	12
2. SECURITY.....	12
2.1. HTTPS 12	
2.2. CALLBACK VERIFICATION FROM CORPID SYSTEM.....	12
2.3. API DATA ENCRYPTION AND DECRYPTION	12
2.3.1. Encryption and Decryption Workflow.....	13
2.3.2. Proper Handling of Encryption and Decryption.....	17
2.3.3. Encryption and Decryption Scope.....	19
2.3.4. Request and Callback Examples.....	21
2.3.5. Encryption and Decryption Reference Implementation.....	24
2.3.6. API Encryption Details.....	27
2.4. COMMON PARAMETERS.....	31
2.4.1. Request Parameters.....	31
2.4.2. Response and Callback Format.....	33
2.4.3. Return Code	33
2.4.4. HTTP Status Code.....	34
2.5. AUTHORISATION SCOPES.....	34
3. API SPECIFICATION WITH IAM SMART	35
3.1. GET ACCESS TOKEN (CORPID-2).....	35
3.1.1. Request Parameters.....	35
3.1.2. Sample Request	35
3.1.3. Response Parameters.....	36
3.1.4. Sample Success Response	37
3.1.5. Sample Failed Response	37
3.2. REQUEST FORMFILLING (CORPID-3) (FOR WEB E-SERVICE)	38

3.2.1.	Request Parameters.....	38
3.2.2.	Sample Request	40
3.2.3.	Response Parameters.....	41
3.2.4.	Sample Success Response	41
3.2.5.	Sample Failed Response	42
3.3.	REQUEST FORMFILLING (CORPID-3) (FOR APP E-SERVICE)	43
3.3.1.	Request Parameters.....	43
3.3.2.	Sample Request	45
3.3.3.	Response Parameters.....	46
3.3.4.	Sample Success Response	47
3.3.5.	Sample Failed Response	47
3.4.	FORMFILLING CALLBACK (CORPID-4) (FOR E-SERVICE).....	48
3.4.1.	Callback Parameters	48
3.4.2.	Example Callback.....	50
3.5.	REQUEST ANONYMOUS FORMFILLING (CORPID-6) (FOR E-SERVICE).....	55
3.5.1.	Request Parameters.....	55
3.5.2.	Sample Request	56
3.5.3.	Response Parameters.....	57
3.5.4.	Sample Success Response	57
3.5.5.	Sample Failed Response	57
3.6.	ANONYMOUS FORMFILLING CALLBACK.....	58
3.6.1.	Callback Parameters	58
3.6.2.	Example Callback.....	59
3.7.	GET ANONYMOUS FORMFILLING RESULT	60
3.7.1.	Request Parameters.....	60
3.7.2.	Sample Request	60
3.7.3.	Response Parameters.....	61
3.7.4.	Sample Success Response	62
3.7.5.	Sample Failed Response	67
3.8.	REQUEST DIGITAL SIGNING (CORPID-21).....	68
3.8.1.	Request Parameters.....	68
3.8.2.	Sample Request	70
3.8.3.	Response Parameters.....	71

3.8.4.	Sample Success Response	71
3.8.5.	Sample Failed Response	72
3.9.	REQUEST DIGITAL SIGNING (CORPID-21) (FOR APP E-SERVICE)	73
3.9.1.	Request Parameters	73
3.9.2.	Sample Request	75
3.9.3.	Response Parameters	76
3.9.4.	Sample Success Response	77
3.9.5.	Sample Failed Response	77
3.10.	DIGITAL SIGNING CALLBACK	78
3.10.1.	Callback Parameters	78
3.10.2.	Example Callback	79
3.11.	REQUEST PDF DIGITAL SIGNING (CORPID-21)	80
3.11.1.	Request Parameters	80
3.11.2.	Sample Request	82
3.11.3.	Response Parameters	83
3.11.4.	Sample Success Response	83
3.11.5.	Sample Failed Response	84
3.12.	REQUEST PDF DIGITAL SIGNING (CORPID-21) (FOR APP E-SERVICE)	85
3.12.1.	Request Parameters	85
3.12.2.	Sample Request	87
3.12.3.	Response Parameters	88
3.12.4.	Sample Success Response	88
3.12.5.	Sample Failed Response	89
3.13.	PDF DIGITAL SIGNING CALLBACK	90
3.13.1.	Callback Parameters	90
3.13.2.	Example Callback	91
3.14.	ACKNOWLEDGES DIGITAL SIGNING RESULT	92
3.14.1.	Request Parameters	92
3.14.2.	Sample Request	92
3.14.3.	Response Parameters	93
3.14.4.	Sample Success Response	93
3.14.5.	Sample Failed Response	93
3.15.	REQUEST ANONYMOUS DIGITAL SIGNING (CORPID-12) (FOR E-SERVICE)	

3.15.1.	Request Parameters.....	94
3.15.2.	Sample Request	96
3.15.3.	Response Parameters.....	97
3.15.4.	Sample Success Response.....	97
3.15.5.	Sample Failed Response	98
3.16.	REQUEST ANONYMOUS PDF DIGITAL SIGNING – (CORPID-13) (FOR E-SERVICE) 99	
3.16.1.	Request Parameters.....	99
3.16.2.	Sample Request	101
3.16.3.	Response Parameters.....	102
3.16.4.	Sample Success Response.....	102
3.16.5.	Sample Failed Response	102
3.17.	ANONYMOUS DIGITAL SIGNING CALLBACK.....	104
3.17.1.	Callback Parameters	104
3.17.2.	Example Callback.....	105
3.18.	GET ANONYMOUS DIGITAL SIGNING RESULT	106
3.18.1.	Request Parameters.....	106
3.18.2.	Sample Request	106
3.18.3.	Response Parameters.....	107
3.18.4.	Sample Success Response.....	108
3.18.5.	Sample Failed Response	108
3.19.	GET ANONYMOUS PDF DIGITAL SIGNING RESULT.....	109
3.19.1.	Request Parameters.....	109
3.19.2.	Sample Request	109
3.19.3.	Response Parameters.....	110
3.19.4.	Sample Success Response.....	110
3.19.5.	Sample Failed Response	111
3.20.	REQUEST BULK DIGITAL SIGNING	112
3.20.1.	Request Parameters.....	112
3.20.2.	Sample Request	115
3.20.3.	Response Parameters.....	117
3.20.4.	Sample Success Response.....	117

3.20.5.	Sample Failed Response	117
3.21.	REQUEST BULK DIGITAL SIGNING (FOR APP E-SERVICE).....	119
3.21.1.	Request Parameters.....	119
3.21.2.	Sample Request	123
3.21.3.	Response Parameters.....	124
3.21.4.	Sample Success Response	125
3.21.5.	Sample Failed Response	125
3.22.	CALLBACK TO RECEIVE BSQC TOKEN.....	126
3.22.1.	Callback Parameters	126
3.22.2.	Example Callback.....	126
3.23.	BULK DIGITAL SIGNING RESULT CALLBACK	128
3.23.1.	Callback Parameters	128
3.23.2.	Example Callback (Signing Success)	130
3.23.3.	Example Callback (Signing Failure)	131
3.24.	ACKNOWLEDGES BULK DIGITAL SIGNING RESULT.....	133
3.24.1.	Request Parameters.....	133
3.24.2.	Sample Request	134
3.24.3.	Response Parameters.....	135
3.24.4.	Sample Success Response	135
3.24.5.	Sample Failed Response	135
3.25.	ENQUIRE BULK DIGITAL SIGNING STATUS.....	136
3.25.1.	Request Parameters.....	136
3.25.2.	Sample Request	136
3.25.3.	Response Parameters.....	137
3.25.4.	Sample Success Response	137
3.25.5.	Sample Failed Response	138
3.26.	CANCEL BULK DIGITAL SIGNING REQUEST	139
3.26.1.	Request Parameters.....	139
3.26.2.	Sample Request	139
3.26.3.	Response Parameters.....	140
3.26.4.	Sample Success Response	140
3.26.5.	Sample Failed Response	140
3.27.	REQUEST ANONYMOUS BULK DIGITAL SIGNING.....	141

3.27.1.	Request Parameters.....	141
3.27.2.	Sample Request	144
3.27.3.	Response Parameters.....	145
3.27.4.	Sample Success Response	145
3.27.5.	Sample Failed Response	146
3.28.	REQUEST BSQC TOKEN.....	147
3.28.1.	Request Parameters.....	147
3.28.2.	Sample Request	147
3.28.3.	Response Parameters.....	148
3.28.4.	Sample Success Response	148
3.28.5.	Sample Failed Response	148
4.	API SPECIFICATION OF CORPID.....	149
4.1.	AUTHENICATION BY REQUESTING QR PAGE/ BROKER PAGE	149
4.1.1.	Request Parameters.....	149
4.1.2.	Sample Request	150
4.2.	REQUEST DOCUMENT LIST IN DOCUMENT WALLET	152
4.2.1.	Request Parameters.....	152
4.2.2.	Sample Request	152
4.2.3.	Response Parameters.....	153
4.2.4.	Sample Success Response	153
4.2.5.	Sample Failed Response	153
4.3.	CALLBACK TO RECEIVE DOCUMENT LIST	155
4.3.1.	Callback Parameters	155
4.3.2.	Example Callback.....	156
4.4.	GET DOCUMENT DETAIL	157
4.4.1.	Request Parameters.....	157
4.4.2.	Sample Request	157
4.4.3.	Response Parameters.....	158
4.4.4.	Sample Success Response	158
4.4.5.	Sample Failed Response	159
4.5.	UPLOAD DOCUMENT.....	160
4.5.1.	Request Parameters.....	160
4.5.2.	Sample Request	160

4.5.3.	Response Parameters.....	161
4.5.4.	Sample Success Response.....	161
4.5.5.	Sample Failed Response	161
4.6.	REQUEST DOCUMENT SHARING	162
4.6.1.	Request Parameters.....	162
4.6.2.	Sample Request	162
4.6.3.	Response Parameters.....	163
4.6.4.	Sample Success Response.....	163
4.6.5.	Sample Failed Response	164
4.7.	CALLBACK TO RECEIVE DOCUMENT SHARING	165
4.7.1.	Callback Parameters	165
4.7.2.	Example Callback.....	166
5.	APPENDIX A	168
5.1.	DETAILS OF “CORPUSERPROFILEFIELDS”	168
5.2.	DETAILS OF “CORPPROFILEFIELDS”	168
5.3.	DETAILS OF “ECORPFIELDS”	168
5.4.	DETAILS OF “PROFILEFIELDS” AND “EMEFIELDS”	169
6.	APPENDIX B	169
6.1.	SUPPORTED VALUE AT SOURCE PARAMETER.....	169
7.	APPENDIX C	171
7.1.	IDENTIFY DOCUMENT TYPE	171
8.	APPENDIX D	171
8.1.	RETURN CODE.....	171
8.2.	HTTP STATUS CODE	173

1. OVERVIEW

1.1. HIGH LEVEL OVERVIEW

To accelerate the digital economy development in Hong Kong through strengthening digital infrastructure and fostering digital transformation, the Digital Policy Office (DPO) is developing the Digital Corporate Identity (CorpID) Platform, thereby facilitating and bringing ease to corporations in their use of e-Government services. The DPO will launch the CorpID Platform in end 2026 and gradually extending the range of services.

CorpID can serve as a common digital key for identity authentication, enabling corporations to use the e-government services and conduct online business transactions at ease. Successful applicant will be issued a CorpID. The responsible person of the corporation can set the authorised representatives and their authorities to conduct electronic commercial activities on behalf of the corporation.

The DPO, collaborates with the Cyberport, has launched the CorpID Sandbox Programme in December 2025 for corporations and government bureaux/departments interested in adopting CorpID to conduct Proof-of-Concept (PoC) testing and develop their applications in order to design application scenarios and solutions that can better meet the market demands. Typical use cases include remote account opening where corporate identities and authorisation of authorised representatives can be verified, login to various online service platforms in a simple, secure and convenient manner, verifiable digital signing of application documents, funding agreements or contracts, and automatic form pre-filling without having to repeatedly provide corporate information, thereby reducing business processing time and human error.

The CorpID Sandbox Programme provides a thematic website and a CorpID Sandbox Developer Portal in which technical references such as this Developer Guide, CorpID Sandbox Application Programming Interface (API) Specifications, CorpID sample applications, information on pre-requisites and relevant configurations are available for developers to understand the technical requirements, and preview the CorpID Simulated APIs. These CorpID Simulated APIs can be tested using mock-up data and predefined responses in the CorpID Sandbox Developer Portal without coding, facilitating smooth and efficient adoption of CorpID and deployment of real applications.

Besides, a dedicated and isolated testing environment (Sandbox Platform) is also developed, offering a suite of API functionalities and simulated workflows that mirror the actual operations for core CorpID functions (namely Corporate Identity Verification, Digital Signing, Form Pre-filling and Document Wallet). While the CorpID Sandbox Developer Portal contains information of all CorpID Simulated APIs, the Sandbox Platform together with a CorpID mock-up mini-program integrated with “iAM Smart” ITE environment and “iAM Smart” testing mobile app, provides executable CorpID Simulated APIs in selected scenarios for illustration purpose. These scenarios allow a desktop web application to access four selected CorpID functionalities, including corporate identity verification (i.e. integrated authentication with “iAM Smart”), form pre-filling after integrated authentication, digital signing after integrated authentication, as well as anonymous form pre-filling. Developers may develop its PoC application to understand the true benefits of CorpID, and ascertain the technical requirements and resource implications in adopting CorpID. In addition, the Sandbox Platform also provides testing on simulated API endpoints (supporting

partial flows) other than these scenarios for verification by developers.

Please note that APIs in the CorpID Sandbox Developer Portal and the Sandbox Platform are tuned for illustration purpose. These APIs are being enhanced continuously following the actual implementation of the production CorpID Platform. These portal and platforms including the simulated APIs and sample applications are intended solely for development and validation purposes and should not be regarded as completely indicative of live production behaviour.

1.2. ENVIRONMENT

1.2.1. CorpID Sandbox Environment

The CorpID system provides a dedicated Sandbox Environment for e-service and its developers to perform application development, integration testing, and pre-production verification.

This Sandbox Environment is completely isolated from the future Production Environment.

For the purpose of this this Sandbox version of the API documentation, the placeholder <CorpID_domain> will be used when constructing website URLs and API endpoints.

B/Ds and developers shall replace <CorpID_domain> with the corresponding value listed in the table below according to the environment to which their system is connecting.

	Sandbox environment
API_HOST	Internet: https://corpid.cyberport.hk

2. SECURITY

2.1. HTTPS

e-service endpoint shall support TLS 1.2 or compatible for using CorpID APIs.

2.2. CALLBACK VERIFICATION FROM CORPID SYSTEM

For Production Environment, CorpID System supports e-service to verify on the transport layer that the callback request it receives actually comes from CorpID System. If e-service chooses to verify, it can request the public key certificate of CorpID System which from the Recognized Certification Authorities in Hong Kong during the onboarding process and configures the certificate in the corresponding location of its HTTPS server to implement the client authentication function. Specific configurations vary depending on the HTTPS server or the reverse proxy used, and the corresponding administration manual needs to be consulted. For example, if using Nginx, e-service should set `ssl_client_certificate` and `ssl_verify_client` in its configuration file to authenticate client requests.

This feature is not supported in CorpID Sandbox environment.

2.3. API DATA ENCRYPTION AND DECRYPTION

HTTPS will be used to secure communication channels between e-services and the CorpID System. However, there is still a chance that the data transferred by HTTPS could be eavesdropped or manipulated by malicious service providers in some special situations. To better protect the data

in transit, an additional layer of data encryption will be applied to all APIs POST request (except the one that e-service request for getting symmetric encryption key). Callbacks from CorpID System will also be encrypted using the method described herein.

The overall idea of API data encryption is:

1. e-service requests for data encryption key from CorpID System;
2. CorpID System generates a symmetric data encryption key;
3. CorpID System stores the generated key and returns to the e-service who initiated the request in secure manner. i.e. Encrypted by the e-service's public key KEK;
4. For subsequent interactions between e-service and CorpID System, business data will be encrypted by the generated key;
5. Both CorpID System and e-service will use the same key to decrypt the API data.
6. If the generated key is expired, e-service have to request a new key and repeat the above process.

2.3.1. Encryption and Decryption Workflow

The following diagram describes detailed workflow on how e-service can request data encryption key, perform encryption and decryption with the key, and request for another new encryption key if the existing key is expired:

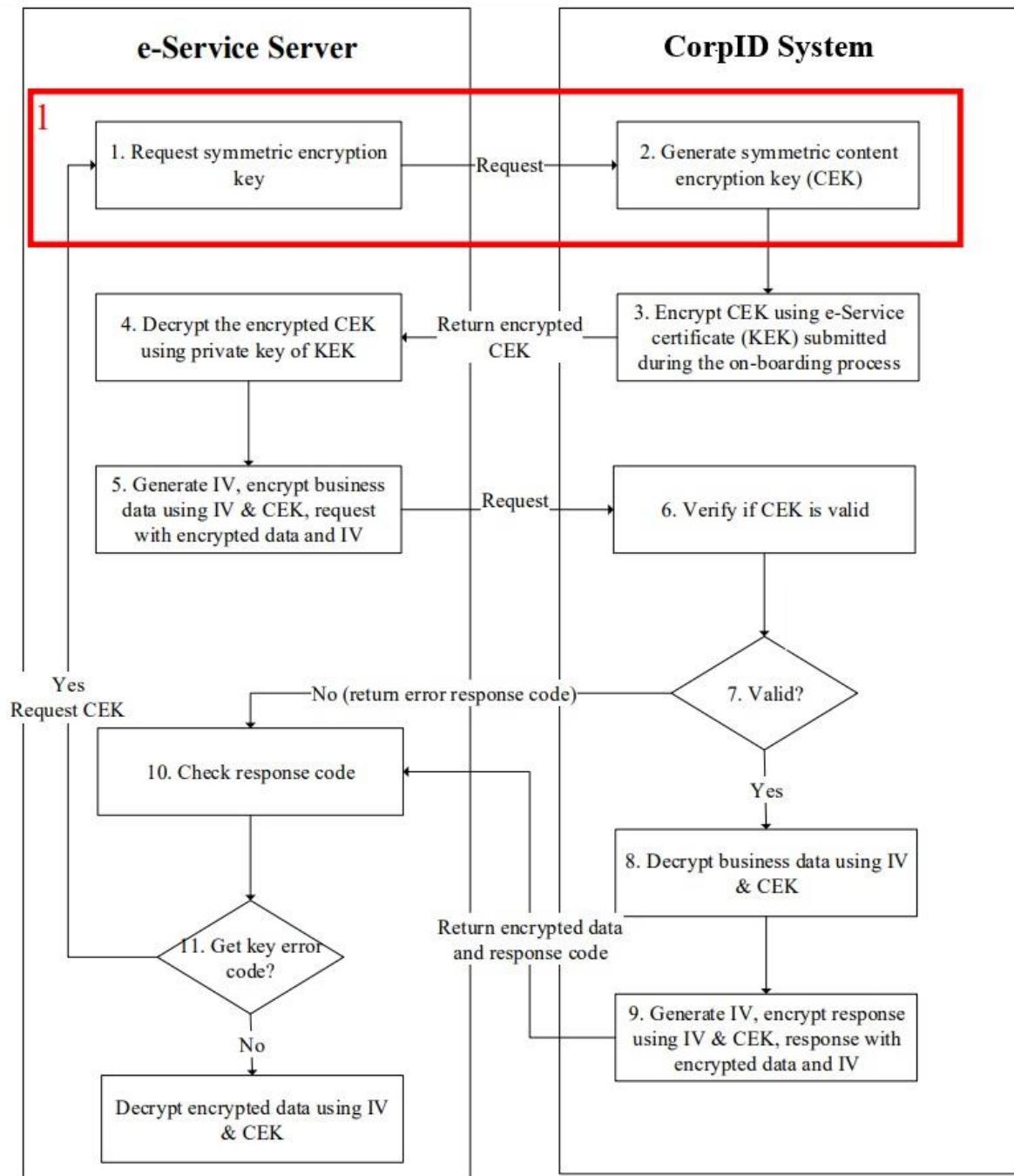


Figure - 1 Application of Content Encryption Key and Procedures of Encryption and Decryption (Perform by e-service)

The APIs for e-service to request Content Encryption Key are marked in red.

High Level Workflow Description of encrypting API request data using Content Encryption Key:

- During the onboarding process, e-service have to download the Sandbox KEK certificate from Self-Service Portal. The certificate contains a public key (denoted as KEK, the Key Encryption Key) that will be used in the following steps.

- CorpID System provides API to e-service for retrieving a symmetric key (denoted as CEK, the Content Encryption Key). Such key will be used to encrypt/decrypt subsequent API data. AES256 symmetric encryption algorithm will be adopted for encrypting the data. CEK will be expired after specified period (refer to section 2.3.6.1) of generation.
- When CorpID System received request from e-service for retrieving CEK, CorpID System will employ the RSA algorithm and use e-service's KEK to encrypt the CEK. Then CorpID System will send the encrypted CEK to e-service. e-service can use the private key of the KEK to decrypt and obtain the symmetric key CEK.
- Before e-service calling APIs of CorpID System, e-service should check whether the symmetric data encryption key (CEK) exists and is valid. If e-service has no such CEK or the CEK is expired, then e-service should request a new CEK from CorpID System. The CEK will be used to encrypt subsequent API data until the key expires.
- Upon receiving the request from e-service, CorpID System will check whether e-service's data encryption key CEK exists and is still valid. If CEK does not exist or expired, error message will be returned to e-service. If CEK is still valid, the key will be used to decrypt the request data. Same CEK will also be used to encrypt respective response or callback data for returning to e-service.

High Level Workflow Description of Callback Encryption and Decryption:

- Before CorpID System initiates a callback, it will check if the data encryption key (CEK) of the corresponding e-service is still valid. If the CEK is expired or does not exist, CorpID System will regenerate a new data encryption key for that e-service.
- CorpID System uses the CEK to encrypt the callback body. Then such CEK will be encrypted by KEK of the e-service. The encrypted callback data and the encrypted CEK will be returned to e-service through the callback.

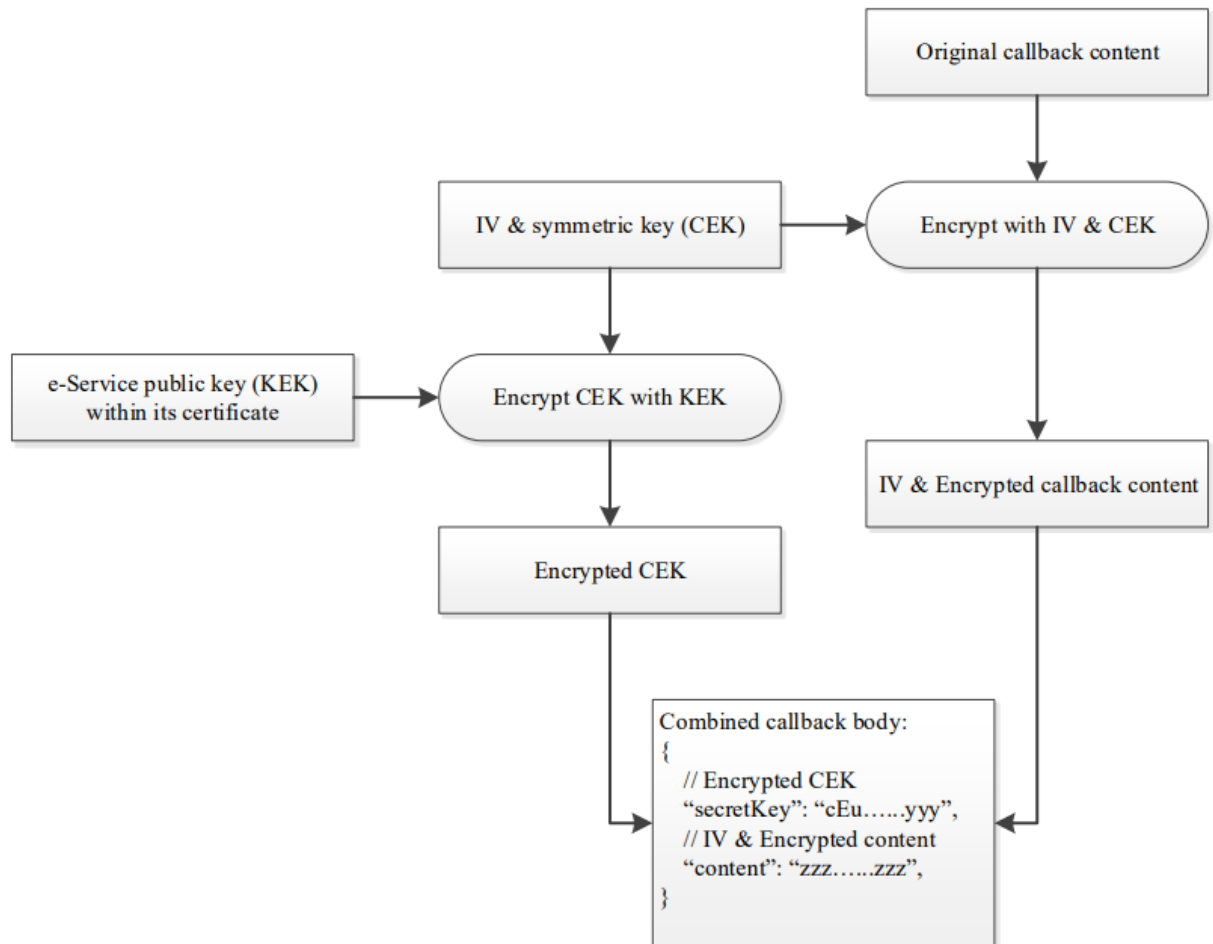


Figure - 2 Callback Encryption Process (Perform by “iAM Smart” System)

- After e-service receives the callback, e-service should use the private key of the KEK to decrypt the encrypted data encryption key (CEK). Then the decrypted data encryption key (CEK) can be used to decrypt the callback data.

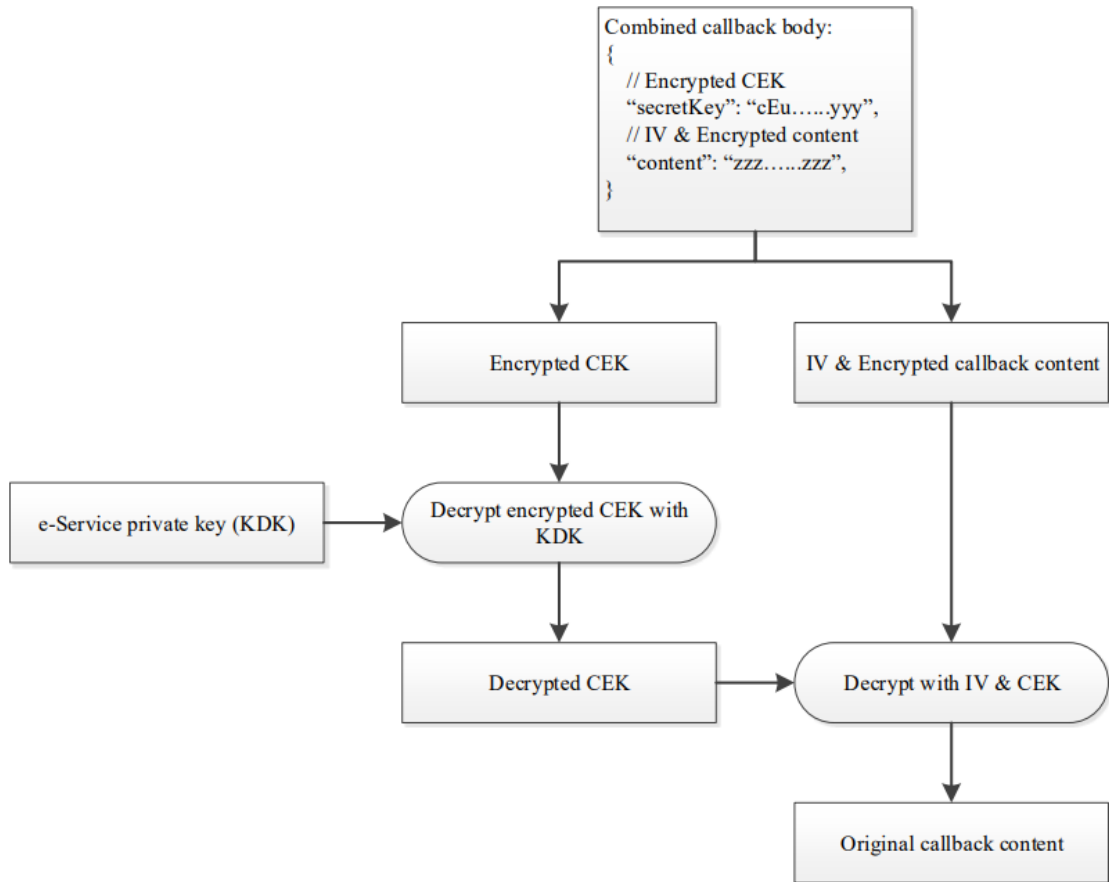


Figure - 3 Callback Decryption Process (Perform by e-service)

2.3.2. Proper Handling of Encryption and Decryption

e-service shall handle the following scenarios properly in their system design:

- Due to unexpected reasons, e-service may receive Encryption/Decryption Error Code. e-service shall perform retry of the “Request Symmetric Content Encryption Key” API specified in section 2.3.6.1 of this document to obtain the latest CEK and try to decrypt content with such CEK when it is received.
- Due to operational needs, the “expiresIn” response parameter of the “Request Symmetric Content Encryption Key” API specified in section 2.3.6.1 of this document may dynamically change, e-service shall calculate the expiry time of the CEK with “issueAt” and “expiresIn” response parameters.
- Under certain circumstances, CorpID API service may be unavailable for a certain period of time. The e-service shall prepare the fallback procedures (e.g. switch to paper mode, prevent users from accessing CorpID functions, etc.).
- e-service shall store the CEK to avoid unnecessary “Request Symmetric Content Encryption Key” API request (see section 2.3.6.1 of this document).

- Even a new KEK is configured for effective in a specified period, the existing CEK will still be returned through the “Request Symmetric Content Encryption Key” API specified in section 2.3.6.1 of this document and encrypted by the old KEK until the expiry of the existing CEK. To use the new KEK, e-service shall request a new CEK through the “Revoke Symmetric Content Encryption Key” and “Request Symmetric Content Encryption Key” APIs specified in sections 2.3.6.2 and 2.3.6.1 respectively of this document or to request a new CEK after its expiry through the same APIs.
- In CorpID Sandbox environment, the public key (denoted as KEK, the Key Encryption Key) is provided by CorpID Sandbox Developer Portal.

2.3.3. Encryption and Decryption Scope

API Type	Parameter Type	Parameter Type/Field	Will Encrypt?	Encryption Rules
e-service request and response	Request parameters	request body	YES	1. Convert the request body into json string 2. Use "AES/GCM/NoPadding" to encrypt the byte array of the json string 3. Base64 encode the encrypted result and set the encoded string to the content field of the request body
		request header	NO	N/A
		url parameters	NO	N/A
	Response parameters	txID	NO	N/A
		code	NO	N/A
		msg	NO	N/A
		content	YES	1. Convert the content field of response body into json string 2. Use "AES/GCM/NoPadding" to encrypt the byte array of the json string 3. Base64 encode the encrypted result and set the encoded string to the content field of the response body
CorpID callback	request parameters	request body	YES	1. Convert the content field of request body into json string 2. Using "AES/GCM/NoPadding" to encrypt the byte array of the json string 3. Base64 encode the encrypted result and set the encoded string to the content field of the request body

CORPID SIMULATED API SPECIFICATION

				4. Encrypt the symmetric data encryption key (CEK) with the public key certificate (KEK). Base64 encode the encryption result and set the encoded string to the key field of the request body
		request header	NO	N/A
		url parameters	NO	N/A
	response parameters	N/A	NO	N/A

2.3.4. Request and Callback Examples

- e-service calls CorpID API

Request Parameter

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header format.

POST

https://<CorpID_domain>/api/v1/auth/getToken

// Request Headers

clientID: "edae2e2529ff46228af1e4d18c8405d1"

signatureMethod: "HmacSHA256"

signature: "5X42Y1B7MEd8Mm%2BonwjiQz9VCZkkrentADskXsYntavU%3D"

timestamp: 1557048906183

nonce: "e893647dc4204eb9b7b8eddd527b687c"

// Unencrypted Request Body

// Clear text before encryption

// {

//   "code": "xxxa42e76bf4cb0846a68e6d83d6096",

//   "grantType": "authorization_code"

// }

// iv + ciphertext

{

  "content":

  "AAADMBVmPKZZmm6/DBASQjG/0Ns1l0BVKk8OcaQLsjiM13XD/c1sZAkQfQQLiTFCxutL9T

DCyVxhl1pUNQk//9VtPKXqmPbB1S/SgeNHYXhgbnQWvNW6Sx4pUcI440JbmUuU5JFWEnGVibj

KmRGZ5QJRYXG89BI0bcjmrpMOnim0TKQ="

}
```

Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "D00000",

  "message": "SUCCESS",

// Clear text before encryption

// "content": {

//   "accessToken": "0ad186353c424c64897fcc00445c9ba1",

//   "tokenType": "Bearer",

//   "issueAt": 1557053922938,

//   "expiresIn": 14400000,

//   "openID": "liR14%2BvX%2F5hSum5uf4ERczu0KcDnIJA5BM7FoM1ag9c%3D",

//   "lastModifiedDate": 1560849218006,

//   "userType": "sign",

//   "scope": "eidapi_auth eidapi_formFilling"

// }

  "content":
  "AAAADIF/FmVrY9df128k2fWjAFrRr29kTBVZfQTvJX+eN454TGl8X9SQ1DkY+W9ntIwmwXRz8f
ajpk4XxYoM9702pFjK4JBzu6+0dCHo0TINruMRD10nMstMthpT7O0hkouEWMbN8YJVpQz2eka
xTq0jWc0X0rj1B16Ir+mbs9h3qDEonbHSceDthfdKZsqLd67wfc9KwXVgVCMdr0vo3H4tdjYYfV1ut
TLNdZbhO9Z0C0Tnr64/7xZq5lZMnQ+ttEhksBeD9/zmPGVs77CzoWJihYW7S7dfZHAcZfoSleS2Ib
BeXnVb0Yg/wkN1FFEPcOSf6DkO4Yck/0iuN/iuXchGbrf5hz/mkhKdi2goxNt/vvU+N8Lk6U8LdwZN
maAZLYXd3jL9ZHnUBkuygcHN8OyaxEkLlJOJ4FFPZpCf2O19aPyCxYJDK1gzp9+k+eD8wn3XH4
0ezYYUJrIqkW9+sS6KX3"

}

}
```

Example error response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```

"code": "D30002",

"message": "content encryption key not exist or expired"

}

```

- CorpID calls back e-service

Request Parameter

```

// Line breaks are for legibility only.

POST

https://<rp_domain>/<rp_context>/<call_back_endpoint>

// Callback Body

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "D00000",

  "message": "SUCCESS",

// Clear text before encryption

// "content": {

// "businessID": "2YotnFZFEjr1zCsicMWpAA",

// "state": "unesidkd",

// "hashCode": "tGzv3JOkF0XG5Qx2TlKWIA",

// "timestamp": 1556450176000,

// "signature": "nnoadisauflanehykdjf",

// "cert": "sdfGSDGsdfaGDEHfjsgGQG.....GSGjljlkwjmh"

// }

// Base64 encoded encrypted CEK

"secretKey":
"XIUu98zV6/W4LPDfZZxzOVY62e4PHZ1e3BzENgA/ysO+rxmnBlqmYRUOqYKS8Fj/6EpuCA1jsn
NY/NJWh8KBz/Kxfe03I1SuD/IHemzMahXLc9x58ecGU0q59VRNO+powUdPTYu0EPH8izh8GQ+Ht
33gic14GMmXuwjmqoo4TZl6h4hPGf6al6piQ9gucsJgFpIoP9genqXqcDCmNkhLoLUMrk5t/ZsA97AP
7DHAGh5RcUT3Qktwz+D1I7Nd4oWUS6yo8ci39epiYSrG9B06OZ/kVo7k6to1Q7KVoCKj3PsMJgC

```

```

UVfOQ0HP1YnH599Cu4eyULaa3rIpafpycPvqh9g==",

// iv + ciphertext

"content":
"AAAADFaP8/FuTcoymbq2hswpgHER9BUM5itaTKWo2/UwtG4wWRsMgri9bSMZ6uLxjKgmGxK
TiR/5N8JuqgX1WJjoMJHp8wnLgSU4FFuGr5yNuGiLpW62fmf95GdF/ikInE5zkzT3hJRzTXbdTALe
XWWPu/88WyZn1REGqSs6m2AdMY5pIhgia8bCHBxSIPBGuBtAujMII7Nlb6ocFwUL0QuaZqPF/h
NJWt/EI46YGHeGKiRB6KP6hXg6K5I9Sz77uQfO60PV5833fnvU"

}

```

2.3.5. Encryption and Decryption Reference Implementation

- Data Encryption

```

private static final String AES_GCM_NOPADDING = "AES/GCM/NoPadding";

/**
 * AES GCM encryption
 * @param contentByte - Bytes of content to be encrypted
 * @param key - CEK (AES 256)
 * @return
 */
public static String encrypt(byte[] contentByte, byte[] key) {
    SecretKeySpec skeySpec = new SecretKeySpec(key, AES);

    byte[] encrypted = null;

    try {
        SecureRandom secureRandom = new SecureRandom();

        byte[] iv = new byte[12];

        // do not reuse iv with the same key

        secureRandom.nextBytes(iv);

        Cipher cipher = Cipher.getInstance(AES_GCM_NOPADDING);

        GCMParameterSpec parameterSpec = new GCMParameterSpec(128, iv);

        cipher.init(Cipher.ENCRYPT_MODE, skeySpec, parameterSpec);
    }
}

```

```

        encrypted = cipher.doFinal(contentByte);

        ByteBuffer byteBuffer = ByteBuffer.allocate(4 + iv.length + encrypted.length);

        byteBuffer.putInt(iv.length);

        byteBuffer.put(iv);

        byteBuffer.put(encrypted);

        byte[] cipherMessage = byteBuffer.array();

        return Base64Util.base64Encode(cipherMessage);

    } catch (NoSuchAlgorithmException | NoSuchPaddingException | InvalidKeyException |
InvalidAlgorithmParameterException | IllegalBlockSizeException | BadPaddingException e) {

        log.error("Encryption error: {}", e);

        throw new CorpIDException(e);

    }
}

```

- Data Decryption

```

private static final String AES_GCM_NOPADDING = "AES/GCM/NoPadding";

/**
 * AES GCM decryption
 * @param contentByte - Bytes of content to be decrypted
 * @param key - CEK (AES 256)
 * @return
 */

public static String decrypt(byte[] content, byte[] key) {

    SecretKeySpec skeySpec = new SecretKeySpec(key, AES);

    ByteBuffer byteBuffer = ByteBuffer.wrap(content);

    int ivLength = byteBuffer.getInt();

    if(ivLength != 12) {

```

```
        throw new IllegalArgumentException("invalid iv length");
    }

    try {

        byte[] iv = new byte[ivLength];

        byteBuffer.get(iv);

        byte[] cipherText = new byte[byteBuffer.remaining()];

        byteBuffer.get(cipherText);


        Cipher cipher = Cipher.getInstance(AES_GCM_NOPADDING);

        GCMParameterSpec parameterSpec = new GCMParameterSpec(128, iv);

        cipher.init(Cipher.DECRYPT_MODE, skeySpec, parameterSpec);

        byte[] original = cipher.doFinal(cipherText);

        return new String(original);

    } catch (NoSuchAlgorithmException | NoSuchPaddingException | InvalidKeyException |
InvalidAlgorithmParameterException | IllegalBlockSizeException | BadPaddingException e) {

        throw new CorpIDException(e);

    }

}
```

2.3.6. API Encryption Details

2.3.6.1. Request Symmetric Content Encryption Key

RESTful API	https://<CorpID_domain>/api/v1/security/getKey
Request Type	POST
Service Version	1.0.0
Description	Each e-service must request a data encryption key to encrypt business data for every POST request when calling the CorpID API. This API allows e-service to request symmetric Content Encryption Key for encrypting business data using AES-256 algorithm. Key expiration time will also be specified in the API response.

2.3.6.1.1. Request Parameters

Common API Parameters described 2.4

2.3.6.1.2. Sample Request

```
{
  "clientId": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX"
}
```

2.3.6.1.3. Response Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	
secretKey	M	String	Base64-encoded symmetric data encryption key encrypted by the certificate that e-service	

			submitted to CorpID System	
pubKey	M	String	Base64-encoded public key of the certificate that is used to encrypt the symmetric data encryption key.	
issueAt	M	Long	secretKey issue time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
expiresIn	M	Long	Key expiration time, expressed in milliseconds.	

2.3.6.1.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "secretKey": "XXXXXX",
    "pubKey": "XXXXXX",
    "issueAt": 1556450176000,
    "expiresIn": 1556456176000
  }
}
```

2.3.6.1.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M99992",
  "msg": "Authorization failed. Applicant is not permitted to submit the request."
}
```

2.3.6.2. Revoke Symmetric Content Encryption Key

RESTful API	https://<CorpID_domain>/api/v1/security/revokeKey
Request Type	POST
Service Version	1.0.0
Description	This API allows e-service to revoke symmetric Content Encryption Key.

2.3.6.2.1. Request Parameters

Common API Parameters described 2.4

2.3.6.2.2. Sample Request

```
{
  "clientId": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX"
}
```

2.3.6.2.3. Response Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	

2.3.6.2.4. Sample Success Response

```
{
  "code": "M00000",
  "msg": ""
}
```

2.3.6.2.5. Sample Failed Response

```
{  
  "code": "M99992",  
  "msg": "Authorization failed. Applicant is not permitted to submit the request."  
}
```

2.4. COMMON PARAMETERS

2.4.1. Request Parameters

There are two kinds of parameters in each API request, the common parameters and the specific business parameters.

Common Parameters

All of the common parameters should be included in every POST request that e-service sending to the CorpID System. The parameters should put in the POST request headers (see examples in the following sections for details). For GET requests or callbacks, most of these common parameters are also required. Which of them should be used and where to put them will be described in details in every GET requests or callback.

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
clientID	M	String	e-service client identifier. The clientID will be assigned to e-service at the initial registration.	
signatureMethod	M	String	signature algorithm: HmacSHA256	
timestamp	M	Long	The timestamp is the request submit time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT. It is used to prevent replay attack. The value MUST be a positive integer and equal or greater than the timestamp used in previous requests.	
nonce	M	String	A nonce is a random string, uniquely generated for each request by e-service. It is used to prevent	

			replay attack. A nonce can be an ASCII string of any length less than or equal to 36 (UUID string length) as long as the uniqueness requirement is met.	
signature	M	Required	It is a signature of the submitted data. e-service uses clientSecret to sign the concatenated string of clientID, signatureMethod, timestamp, nonce, and encrypted_request body to get the signature. The pseudo code to generate the signature is shown at below.	

Pseudo code for signature generation

```

message = clientID + signatureMethod + timestamp + nonce + encrypted_request_body;

sha256_HMAC = Mac.getInstance("HmacSHA256");

secret_key = new SecretKeySpec(secret.getBytes(), "HmacSHA256");

sha256_HMAC.init(secret_key);

hash = Base64.encodeBase64String(sha256_HMAC.doFinal(message.getBytes()));

// If the hash needs to be URL encoded

signature = URLEncoder.encode(hash, "UTF-8");

```

Specific Business Parameters

CorpID System returns results based on the business request parameters. Those parameters are normally put in the request body as json data and submitted as POST request.

2.4.2. Response and Callback Format

CorpID System returns data in json format. The format is:

```
{
  // The txID is a unique identifier of response message.
  // The identifier will be used during troubleshooting
  "txID": "xxxxxx",
  "code": "xxxxxx",
  // return code
  "message": "xxxxxx",
  //status message
  "content": {
    // the actual business data
  }
}
```

When the code is M00000, the service execution is successful, and the content field returned by the HTTPS request contains the actual business data.

Other than M00000, it indicates that the service execution is abnormal, and the content field will be empty or only contains required parameters, e.g. businessID.

For callback, if the data passed are in the URL and error occurred before sending the callback, the URL will contain the error_code field to tell e-service the error code. If the data is in the callback body, the above json format will be used.

2.4.3. Return Code

Refer to Appendix D

2.4.4. HTTP Status Code

Refer to Appendix D

2.5. AUTHORISATION SCOPES

CorpID System makes reference to the OAuth 2.0 authentication framework to enable e-service to gain access to CorpID APIs, with CorpID user's authorisation using “iAM Smart” Mobile App and CorpID mini-program.

Determine Authorisation Scope

To access CorpID APIs, e-service should first determine the authorisation scope(s) required for the CorpID API(s) invoked for its business needs. e-service may refer the table below for the authorisation mapping:

API	Scope Value
Authentication	corpidapi_auth
Switch corporation under same user	corpidapi_corplist
Form Filling with Service Login (aka Profiles)	corpidapi_profiles
Form Filling without Service Login (Anonymous Form Filling)	corpidapi_formfill
Digital Signing with Service Login	corpidapi_sign
Digital Signing without Service Login (Anonymous Digital Signing)	corpidapi_sign
Re-authentication with Service Login	corpidapi_fr
Bulk Digital Signing with Service Login	corpidapi_bulksign
Bulk Digital Signing without Service Login (Anonymous Bulk Digital Signing)	corpidapi_bulksign
Document Wallet	corpidapi_docwallet

e-service should combine all authorisation scope(s) required into a single request to get authorisation from CorpID user through CorpID System when CorpID user requests login to e-service using “iAM Smart” Mobile App and CorpID mini program. For example, authorisation scope “corpidapi_auth”, “corpidapi_formfill” and “corpidapi_sign” are for CorpID authentication form filling and digital signing respectively.

3. API SPECIFICATION WITH IAM SMART

3.1. GET ACCESS TOKEN (CORPID-2)

RESTful API	<a href="https://<CorpID_domain>/api/v1/auth/getToken">https://<CorpID_domain>/api/v1/auth/getToken
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service uses this API to retrieve the access token and tokenised ID (openID). An authorisation code is necessary during the process. The accessToken and openID will be used to call corresponding CorpID services subsequently.

3.1.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
ticketID	M	String	The authorisation code is received from the authorisation server. It can only be used once.	
grantType	M	String	the value MUST be set to authorization_code.	
iAMSmart_AccessToken	M	String	the accessToken from iAM Smart's getToken	
iAMSmart_TokenizedID	M	String	the tokenized ID from iAM Smart's getToken	

3.1.2. Sample Request

```
{
  "clientId": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
```

// Unencrypted Request Body

```

"content":{
    "ticketID":"XXXXXX",
    "grantType": "XXXXXX",
    "iAMSmart_AccessToken": "XXXXXX",
    "iAMSmart_TokenizedID": "XXXXXX"
}
}

```

3.1.3. Response Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	
accessToken	M	String	accessToken value can only be used once.	
tokenType	M	String	Token type, support "Bearer" only	
issueAt	M	Long	The accessToken issue time is expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
expiresIn	M	Long	The lifetime in milliseconds of the token. The value may vary for different e-services.	
openID	M	String	Tokenised ID, uniquely generated for each user of each e-service website or mobile application.	
userType	M	String	iam or p12 iam: "iAM Smart+" user (digital signing capability) p12: digital certificate user	
scope	M	String	The scope of the token. Please refer to the	

			corresponding section specified in each API function.	
--	--	--	---	--

3.1.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "accessToken": "XXXXXX",
    "tokenType": "XXXXXX",
    "issueAt": XXXXX,
    "expiresIn": XXXXX,
    "openID": "XXXXXX",
    "userType": "XXXXXX",
    "scope": "XXXXXX"
  }
}
```

3.1.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M99992",
  "msg": "Authorization failed. Applicant is not permitted to submit the request."
}
```

3.2. REQUEST FORMFILLING (CORPID-3) (FOR WEB E-SERVICE)

RESTful API	https://<CorpID_domain>/api/v1/formFilling/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to request for form filling. If the following eCorpFields and corpProfileFields are not provided or are empty array, the HTTP code 200 with error code D20002 (empty parameter {eCorpFields, corpProfileFields}) will be returned. If same data field exists in both corpProfileFields and corpProfileFields, it will be shown as profile field.

3.2.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
source	M	String	Request initiator.	The supported source values refer to Appendix B
redirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	

CORPID SIMULATED API SPECIFICATION

accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
iamFormfill	M (Conditional)	Boolean	true: get both corporation and personal information false: get corporation information only	Only for “iAM Smart” user; For Sandbox, the value would only proceed as true.
clientID_iAM	M (Conditional)	String	The e-service’s client ID of “iAM Smart” system	When iamFormfill is true, there must be values here
formName	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
formNum	O	String	The name of the form should be encoded in Unicode. Maximum Length: 20s	
formDesc	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
callbackContentType	O	String	Specify the content type of the callback request that an e-service expects to receive in Section 3.4. The callbackContentType supports “application/json” and “multipart/formdata” values. The default value is “application/json”. If the e-service endpoint has a request size limit (Section 3.4), e-service may consider using “multipart/form-data” type.	
eCorpFields	M	Array	Specify the “e-Corp” fields to be requested.	Refer to Appendix A
corpUserProfileFields	M	String Array	Specify the “corpUserProfileFields” fields to be requested.	Refer to Appendix A
corpProfileFields	M	Array	Specify the profile fields to	Refer to Appendix A

			be requested.	
eMEFields	M (Conditional)	Array	Specify the “e-ME” fields to be requested. When iamFormfill is true, there must be values here Only for “iAM Smart” user	Refer to Appendix A
profileFields	M (Conditional)	Array	Specify the profile fields to be requested. When iamFormfill is true, there must be values here Only for “iAM Smart” user	Refer to Appendix A

3.2.2. Sample Request

```
{
  "clientId": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
    "accessToken": "XXXXXX",
    "openID": "XXXXXX",
    "clientId_iAM": "XXXXXX",
    "source": "Android_Chrome",
    "iamFormfill": true,
    "formName": "Example Account Registration Form",
    "formNum": "APP0001",
    "formDesc": "Example form description",
    "callbackContentType": "multipart/form-data",
    "redirectURI": "https://<CorpID_domain>/api/v1/formFilling/callback",
    "state": "XXXXXX",
    "corpUserProfileFields": ["XXXXXX", "XXXXXX"],
    "corpProfileFields": ["XXXXXX", "XXXXXX"],
  }
}
```

```

    "eCorpFields": ["XXXXXX", "XXXXXX"],
    "profileFields": ["XXXXXX", "XXXXXX"],
    "eMEFields": ["XXXXXX", "XXXXXX"]
  }
}

```

3.2.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
ticketID	M	String (Conditional)	TicketID is a unique identifier to trigger CorpID mini-program when authByQR is false.	
authByQR	M	Boolean	If user completes authentication by scanning QR code (i.e. different device), then "true" will be returned to e-service.	

3.2.4. Sample Success Response

```

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}

```

3.2.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M99992",  
  "msg": "Authorization failed. Applicant is not permitted to submit the request."  
}
```

3.3. REQUEST FORMFILLING (CORPID-3) (FOR APP E-SERVICE)

RESTful API	https://<CorpID_domain>/api/appv1/formFilling/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	False
Description	e-service (App) can use this API to request for form filling. If the following eCorpFields and corpProfileFields are not provided or are empty array, the HTTP code 200 with error code D20002 (empty parameter {eCorpFields, corpProfileFields}) will be returned. If same data field exists in both corpProfileFields and corpProfileFields, it will be shown as profile field.

3.3.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
source	M	String	App_Link (iOS) or App_Package (Android)	
clientRedirectURI	M (iOS only)	String	callback redirect URI. The value should be URL encoded and registered in the self-service portal.	
packageName	M (Android only)	String	If the state parameter is presented in the request message, the same state value will be returned to e-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by e-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers,	

			underscore and hyphens are accepted.	
activityClass	M (Android only)	String	Activity class name of the e-service App for callback use.	
activityParams	M (Android only)	String	Optional params used during callback.	
serverRedirectURI	M	String	callback URI.	
callbackContentType	O	String	Specify the content type of the callback request that an e-service expects to receive in Section 3.4. The callbackContentType supports “application/json” and “multipart/formdata” values. The default value is “application/json”. If the e-service endpoint has a request size limit (Section 3.4), e-service may consider using “multipart/form-data” type.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
iamFormfill	M (Conditional)	Boolean	true: get both corporation and personal information	Only for “iAM Smart” user;

			false: get corporation information only	For Sandbox, the value would only proceed as true.
clientID_iAM	M (Conditional)	String	The e-service's client ID of "iAM Smart" system	When iamFormfill is true, there must be values here
formName	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
formNum	O	String	The name of the form should be encoded in Unicode. Maximum Length: 20s	
formDesc	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
corpUserProfileFields	M	String Array	Specify the "corpUserProfileFields" fields to be requested.	Refer to Appendix A
eCorpFields	M	String Array	Specify the "e-Corp" fields to be requested.	Refer to Appendix A
corpProfileFields	M	String Array	Specify the profile fields to be requested.	Refer to Appendix A
eMEFields	M (Conditional)	Array	Specify the "e-ME" fields to be requested. Only for "iAM Smart" user	Refer to Appendix A
profileFields	M (Conditional)	Array	Specify the profile fields to be requested. Only for "iAM Smart" user	Refer to Appendix A

3.3.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
```

```
// Clear text before encryption

"content": {
  "businessID": "XXXXXX",
  "accessToken": "XXXXXX",
  "openID": "XXXXXX",
  "clientID_iAM": "XXXXXX",
  "source": "App_Package",
  "packageName": "com.onlineservice.myapp",
  "activityClass": "callback_activity",
  "activityParams": "callback_param",
  "redirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>",
  "state": "XXXXXX",
  "corpUserProfileFields": ["XXXXXX", "XXXXXX"],
  "corpProfileFields": ["XXXXXX", "XXXXXX"],
  "eCorpFields": ["XXXXXX", "XXXXXX"],
  "profileFields": ["XXXXXX", "XXXXXX"],
  "eMEFields": ["XXXXXX", "XXXXXX"]
}
}
```

3.3.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
ticketID	M	String (Conditional)	TicketID is a unique identifier to trigger CorpID mini-program when authByQR is false.	
authByQR	M	Boolean	If user completes authentication by scanning QR code (i.e. different device), then	

			"true" will be returned to e-service.	
--	--	--	---------------------------------------	--

3.3.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}
```

3.3.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M99992",
  "msg": "Authorization failed. Applicant is not permitted to submit the request."
}
```

3.4. FORMFILLING CALLBACK (CORPID-4) (FOR E-SERVICE)

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Provide Simulation in Sandbox	True
Description	This callback allow CorpID System to pass Form Filling information with “application/json” content type to e-service upon consent.

3.4.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
corp	M	Object	Information of CorpID account.	
iAM	O	Object	Information of “iAM Smart” user	
Details of “Corp”				
corpID	O	String	The unique ID from "CorpID" system	
brn	O	String	Business Registration Number	
corpNameEN	O	String	Corporation Name (EN)	
corpNameTC	O	String	Corporation Name (TC)	
corpNameSC	O	String	Corporation Name (SC)	
corpAddr	O	String	Corporation Business Address	
corpTypeEN	O	String	Corporation Type (EN)	
corpTypeTC	O	String	Corporation Type (TC)	
corpTypeSC	O	String	Corporation Type (SC)	
corpTel	O	String	Corporation Telephone No.	

CORPID SIMULATED API SPECIFICATION

corpStatusEN	O	String	Corporation Active Status (EN)	
corpStatusTC	O	String	Corporation Active Status (TC)	
corpStatusSC	O	String	Corporation Active Status (SC)	
placeOfIncorp	O	String	Place of incorporation	
dateOfReg	O	String	Date of registration	
corpJoinTs	O	String	Corporation join datetime	
id_cty_issue	O	String	ISO 3166 3-letter country code	
id_type	O	String	1. Identity card 2. Passport 3. Other identity type	
Details of “iAM”				
idNo	O	JSON Object	HKIC number	
prefix	O	String	prefix	
enName	O	JSON Object	English name	
chName	O	JSON Object	Chinese name	
birthDate	O	String	Date of birth	
gender	O	String	gender	
maritalStatus	O	String	marital status	
homeTelNumber	O	JSON Object	home telephone number	
officeTelNumber	O	JSON Object	office telephone number	
mobileNumber	O	JSON Object	mobile number	
emailAddress	O	String	email address	
residentialAddress	O	JSON Object	residential address	
postalAddress	O	JSON Object	postal address	
educationLevel	O	String	education level	
chNameVerified	O	String	ImmD characters code point will be returned to e-service if the Chinese name is verified.	
addressDocInfo	O	JSON Object	the data related to the obtained e-bill, including the provider name, retrieval date, owner name, service address and postal address. The sub-field lastRetrievalDate refers to the date and time at which “iAM Smart” obtained the e-bill.	
addressDocFile	O	JSON Object	the e-bill obtained from an address data provider. Sub-fields include the PDF e-bill file (docFile), the SHA256 hash (docHash) and the bill date of the e-bill (billDate).	

3.4.2. Example Callback

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "businessID": "XXXXXX",
    "state": "XXXXXX",
    "corp": {
      "corpID": "XXXXXX",
      "brn": "XXXXXX",
      "corpNameEN": "XXXXXX",
      "corpNameTC": "XXXXXX",
      "corpNameSC": "XXXXXX",
      "corpAddr": "XXXXXX",
      "corpTypeEN": "XXXXXX",
      "corpTypeTC": "XXXXXX",
      "corpTypeSC": "XXXXXX",
      "corpTel": "XXXXXX",
      "corpStatusEN": "XXXXXX",
      "corpStatusTC": "XXXXXX",
      "corpStatusSC": "XXXXXX",
      "placeOfIncorp": "XXXXXX",
      "dateOfReg": "XXXXXX",
      "corpJoinTs": "XXXXXX"
    },
    "iAM": {
      "idNo": {
        "Identification": "A123456",
        "CheckDigit": "A"
      },
      "prefix": "Mr",
      "enName": {
        "UnstructuredName": "SAN, Chi Nan"
      },
      "chName": {
        "ChineseName": "申 智能"
      },
      "chNameVerified": "申智能",

```

```
"birthDate": "19960128",
"gender": "M",
"maritalStatus": "S",
"homeTelNumber": {
  "CountryCode": "852",
  "SubscriberNumber": "98765432"
},
"officeTelNumber": {
  "CountryCode": "1",
  "SubscriberNumber": "123456"
},
"mobileNumber": {
  "CountryCode": "1",
  "SubscriberNumber": "98765432"
},
"emailAddress": "scn@digitalpolicy.gov.hk",
// Three different tags exist in residentialAddress
// and the returned value can be either one of the following
// ChiPremisesAddress, EngPremisesAddress, FreeFormatAddress
// Both ChiPremisesAddress and EngPremisesAddress contain either
// 1): Standard/Village Address
// 2): Lot Address
// Example of ChiPremisesAddress with Standard/Village Address
"residentialAddress": {
  "ChiPremisesAddress": {
    "Region": "香港",
    "ChiDistrict": {
      "DcDistrict": "WC",
      "Sub-district": "灣仔"
    },
    "BuildingName": "灣仔政府大樓",
    "ChiEstate": {
      "EstateName": "華富",
      "ChiPhase": {
        "PhaseName": "華清"
      }
    },
    "ChiStreet": {
      "StreetName": "港灣道",
```

```
        "BuildingNoFrom": "12"
      },
      "ChiBlock": {
        "BlockDescriptor": "座",
        "BlockNo": "東"
      },
      "Chi3dAddress": {
        "ChiFloor": {
          "FloorNum": "15"
        },
        "ChiUnit": {
          "UnitDescriptor": "室",
          "UnitNo": "A1"
        }
      }
    },
    // Example of EngPremisesAddress with Lot address
    "residentialAddress": {
      "EngPremisesAddress": {
        "EngLot": {
          "EngStructuredLot": {
            "DdType": "DD",
            "DdNo": "110",
            "LotType": "LOT",
            "LotNo": "157",
            "LotSection1": "A",
            "LotSubsection1": "1",
            "LotSection2": "B",
            "LotSubsection2": "1",
            "LotSection3": "AA",
            "LotSubsection3": "1",
            "LotExtendPortionCode": "3"
          }
        }
      }
    },
    // Example of FreeFormatAddress
    "residentialAddress": {
```

```
"FreeFormatAddress": {
  "LanguageCode": "en",
  "AddressLine1": "Unit 2000, 200/F",
  "AddressLine2": "5033 Yitian Road, Futian CBD",
  "AddressLine3": "Futian district, Shenzhen"
},
// Example of postalAddress with EngPostBox
"postalAddress": {
  "PostBoxAddress": {
    "EngPostBox": {
      "PoBoxNo": 24700,
      "PostOffice": "ABERDEEN POST OFFICE",
      "PostOfficeRegion": "HONG KONG"
    }
  }
},
"educationLevel": "T",
// all the sub-fields come from the address data provider
"addressDocInfo": {
  // name of the address data provider, possible values are:
  // The Hong Kong and China Gas Company Limited
  // CLP Power Hong Kong Ltd.
  // HK Electric
  // Water Supplies Department
  "enProviderName": "HK Electric",
  "tcProviderName": "港燈",
  "scProviderName": "港灯",
  // unstructured name of the e-bill owner
  "enUserName": "SAN, Chi Nan",
  "tcUserName": "申智能",
  // service address of the e-bill, at most 4 lines
  "enServiceAddress": {
    "addressLine1": "19/F, 213 Queen's Road East",
    "addressLine2": "Wanchai, Hong Kong"
  },
  "tcServiceAddress": {
    "addressLine1": "香港灣仔皇后大道東 231 號",
    "addressLine2": "胡忠大廈 19 樓"
```

```
    },
    // postal address of the e-bill, at most 4 lines
    "enPostalAddress": {
      "addressLine1": "19/F, 213 Queen's Road East",
      "addressLine2": "Wanchai, Hong Kong"
    },
    "tcPostalAddress": {
      "addressLine1": "香港灣仔皇后大道東 231 號",
      "addressLine2": "胡忠大廈 19 樓"
    },
    // last retrieval date of the e-bill
    // expressed in the number of milliseconds since 1970/1/1 GMT
    "lastRetrievalDate": 1560853718006
  },
  "addressDocFile": {
    // Base64 encoded string of the PDF e-bill file
    "docFile": "JVBERi0xLjUKJcK1wrXCtcK1CjEgMCBv...iag==",
    // SHA256 hash of PDF e-bill file
    "docHash": "69e3...c4e5",
    // bill date of the e-bill
    // expressed in the number of milliseconds since 1970/1/1 GMT
    "billDate": 1560849218006
  }
}
}
```

3.5. REQUEST ANONYMOUS FORMFILLING (CORPID-6) (FOR E-SERVICE)

RESTful API	https://<CorpID_domain>/api/v1/formFilling/anonymous/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to request anonymous form filling. If the following eCorpFields and corpProfileFields are not provided or are empty array, the HTTP code 200 with error code D20002 (empty parameter {eCorpFields, corpProfileFields}) will be returned. If same data field exists in both corpProfileFields and corpProfileFields, it will be shown as profile field.

3.5.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
iamFormfill	M (Conditional)	Boolean	true: get both corporation and personal information	Only for “iAM Smart” user; For Sandbox, the value would only proceed as true.
formName	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
formNum	O	String	The name of the form should be encoded in Unicode. Maximum Length: 20s	
formDesc	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	

corpUserProfileFields	M	String Array	Specify the “corpUserProfileFields” fields to be requested.	Refer to Appendix A
eCorpFields	M	Array	Specify the “e-Corp” fields to be requested.	Refer to Appendix A
corpProfileFields	M	Array	Specify the profile fields to be requested.	Refer to Appendix A
eMEFields	M (Conditional)	Array	Specify the “e-ME” fields to be requested. Only for “iAM Smart” user	Refer to Appendix A
profileFields	M (Conditional)	Array	Specify the profile fields to be requested. Only for “iAM Smart” user	Refer to Appendix A

3.5.2. Sample Request

```
{
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
    "formName": "XXXXXX",
    "formNum": "XXXXXX",
    "formDesc": "XXXXXX",
    "corpUserProfileFields": ["XXXXXX", "XXXXXX"],
    "corpProfileFields": ["XXXXXX", "XXXXXX"],
    "eCorpFields": ["XXXXXX", "XXXXXX"],
    "profileFields": ["XXXXXX", "XXXXXX"],
    "eMEFields": ["XXXXXX", "XXXXXX"]
  }
}
```

3.5.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
ticketID	M	String (Conditional)	TicketID is a unique identifier to trigger CorpID mini-program when authByQR is false.	
authByQR	M	Boolean	If user completes authentication by scanning QR code (i.e. different device), then "true" will be returned to e-service.	

3.5.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}
```

3.5.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M99992",
  "msg": "Authorization failed. Applicant is not permitted to submit the request."
}
```

3.6. ANONYMOUS FORMFILLING CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	GET
Provide Simulation in Sandbox	True
Description	This callback is used to pass authCode to e-service Server. The URI must be registered in the self-service portal.

3.6.1. Callback Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
ticketID	M (Conditional)	String	The authorisation code is generated by the “iAM Smart” System. The authorisation code will be expired 1 minute after issuance. e-service MUST NOT use the authorisation code more than once. An error message will be returned if an authorisation code is expired or re-used.	
error_code	M (Conditional)	String	Return error code when exception.	

3.6.2. Example Callback

Allow

// Line breaks are for legibility only.

GET

https://<callback_endpoint>

?ticketID=0ad186353c424c64897fcc00445c9ba1

&state=eddd527b6

Deny

// Line breaks are for legibility only.

GET

https://<callback_endpoint>

?error_code=D20001

&state=eddd527b6

3.7. GET ANONYMOUS FORMFILLING RESULT

RESTful API	https://<CorpID_domain>/api/v1/formFilling/anonymous/getResult
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to retrieve anonymous form filling result.

3.7.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
clientID_iAM	M (Conditional)	String	Client ID value from “iAM Smart” system Only for “iAM Smart” user	

3.7.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
```

```

        "businessID":"XXXXXX",
        "accessToken":"XXXXXX",
        "openID": "XXXXXX",
        "clientID_iAM": "XXXXXX"
    }
}

```

3.7.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
corp	M	Object	Information of CorpID user	
iAM	O	Object	Information of “iAM Smart” user	
Details of “corp”				
corpID	O	String	The unique ID from "CorpID" system	
brn	O	String	Business Registration Number	
corpNameEN	O	String	Corporation Name (EN)	
corpNameTC	O	String	Corporation Name (TC)	
corpNameSC	O	String	Corporation Name (SC)	
corpAddr	O	String	Corporation Business Address	
corpTypeEN	O	String	Corporation Type (EN)	
corpTypeTC	O	String	Corporation Type (TC)	
corpTypeSC	O	String	Corporation Type (SC)	
corpTel	O	String	Corporation Telephone No.	
corpStatusEN	O	String	Corporation Active Status (EN)	
corpStatusTC	O	String	Corporation Active Status (TC)	
corpStatusSC	O	String	Corporation Active Status (SC)	
placeOfIncorp	O	String	Place of incorporation	
dateOfReg	O	String	Date of registration	
corpJoinTs	O	String	Corporation join datetime	
id_cty_issue	O	String	ISO 3166 3-letter country code	

id_type	O	String	1. Identity card 2. Passport 3. Other identity type	
Details of “iAM”				
idNo	O	JSON Object	HKIC number	
prefix	O	String	prefix	
enName	O	JSON Object	English name	
chName	O	JSON Object	Chinese name	
birthDate	O	String	Date of birth	
gender	O	String	gender	
maritalStatus	O	String	marital status	
homeTelNumber	O	JSON Object	home telephone number	
officeTelNumber	O	JSON Object	office telephone number	
mobileNumber	O	JSON Object	mobile number	
emailAddress	O	String	email address	
residentialAddress	O	JSON Object	residential address	
postalAddress	O	JSON Object	postal address	
educationLevel	O	String	education level	
chNameVerified	O	String	ImmD characters code point will be returned to e-service if the Chinese name is verified.	
addressDocInfo	O	JSON Object	the data related to the obtained e-bill, including the provider name, retrieval date, owner name, service address and postal address. The sub-field lastRetrievalDate refers to the date and time at which “iAM Smart” obtained the e-bill.	
addressDocFile	O	JSON Object	the e-bill obtained from an address data provider. Sub-fields include the PDF e-bill file (docFile), the SHA256 hash (docHash) and the bill date of the e-bill (billDate).	

3.7.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```
"code": "M00000",
"msg": "",
"content": {
  "businessID": "XXXXXX",
  "corp": {
    "corpID": "XXXXXX",
    "brn": "XXXXXX",
    "corpNameEN": "XXXXXX",
    "corpNameTC": "XXXXXX",
    "corpNameSC": "XXXXXX",
    "corpAddr": "XXXXXX",
    "corpTypeEN": "XXXXXX",
    "corpTypeTC": "XXXXXX",
    "corpTypeSC": "XXXXXX",
    "corpTel": "XXXXXX",
    "corpStatusEN": "XXXXXX",
    "corpStatusTC": "XXXXXX",
    "corpStatusSC": "XXXXXX",
    "placeOfIncorp": "XXXXXX",
    "dateOfReg": "XXXXXX",
    "corpJoinTs": "XXXXXX"
  },
  "iAM": {
    "idNo": {
      "Identification": "A123456",
      "CheckDigit": "A"
    },
    "prefix": "Mr",
    "enName": {
      "UnstructuredName": "SAN, Chi Nan"
    },
    "chName": {
      "ChineseName": "申智能"
    },
    "chNameVerified": "申智能",
    "birthDate": "19960128",
    "gender": "M",
    "maritalStatus": "S",
    "homeTelNumber": {
```

```
"CountryCode": "852",
"SubscriberNumber": "98765432"
},
"officeTelNumber": {
  "CountryCode": "1",
  "SubscriberNumber": "123456"
},
"mobileNumber": {
  "CountryCode": "1",
  "SubscriberNumber": "98765432"
},
"emailAddress": "scn@digitalpolicy.gov.hk",
// Three different tags exist in residentialAddress
// and the returned value can be either one of the following
// ChiPremisesAddress, EngPremisesAddress, FreeFormatAddress
// Both ChiPremisesAddress and EngPremisesAddress contain either
// 1): Standard/Village Address
// 2): Lot Address
// Example of ChiPremisesAddress with Standard/Village Address
"residentialAddress": {
  "ChiPremisesAddress": {
    "Region": "香港",
    "ChiDistrict": {
      "DeDistrict": "WC",
      "Sub-district": "灣仔"
    },
    "BuildingName": "灣仔政府大樓",
    "ChiEstate": {
      "EstateName": "華富",
      "ChiPhase": {
        "PhaseName": "華清"
      }
    },
    "ChiStreet": {
      "StreetName": "港灣道",
      "BuildingNoFrom": "12"
    },
    "ChiBlock": {
      "BlockDescriptor": "座",
```

```
        "BlockNo": "東"
      },
      "Chi3dAddress": {
        "ChiFloor": {
          "FloorNum": "15"
        },
        "ChiUnit": {
          "UnitDescriptor": "室",
          "UnitNo": "A1"
        }
      }
    }
  },
  // Example of EngPremisesAddress with Lot address
```

```
  "residentialAddress": {
    "EngPremisesAddress": {
      "EngLot": {
        "EngStructuredLot": {
          "DdType": "DD",
          "DdNo": "110",
          "LotType": "LOT",
          "LotNo": "157",
          "LotSection1": "A",
          "LotSubsection1": "1",
          "LotSection2": "B",
          "LotSubsection2": "1",
          "LotSection3": "AA",
          "LotSubsection3": "1",
          "LotExtendPortionCode": "3"
        }
      }
    }
  },
```

```
  // Example of FreeFormatAddress
```

```
  "residentialAddress": {
    "FreeFormatAddress": {
      "LanguageCode": "en",
      "AddressLine1": "Unit 2000, 200/F",
      "AddressLine2": "5033 Yitian Road, Futian CBD",
```

```
        "AddressLine3": "Futian district, Shenzhen"
    },
    // Example of postalAddress with EngPostBox
    "postalAddress": {
        "PostBoxAddress": {
            "EngPostBox": {
                "PoBoxNo": 24700,
                "PostOffice": "ABERDEEN POST OFFICE",
                "PostOfficeRegion": "HONG KONG"
            }
        }
    },
    "educationLevel": "T",
    // all the sub-fields come from the address data provider
    "addressDocInfo": {
        // name of the address data provider, possible values are:
        // The Hong Kong and China Gas Company Limited
        // CLP Power Hong Kong Ltd.
        // HK Electric
        // Water Supplies Department
        "enProviderName": "HK Electric",
        "tcProviderName": "港燈",
        "scProviderName": "港灯",
        // unstructured name of the e-bill owner
        "enUserName": "SAN, Chi Nan",
        "tcUserName": "申智能",
        // service address of the e-bill, at most 4 lines
        "enServiceAddress": {
            "addressLine1": "19/F, 213 Queen's Road East",
            "addressLine2": "Wanchai, Hong Kong"
        },
        "tcServiceAddress": {
            "addressLine1": "香港灣仔皇后大道東 231 號",
            "addressLine2": "胡忠大廈 19 樓"
        },
        // postal address of the e-bill, at most 4 lines
        "enPostalAddress": {
            "addressLine1": "19/F, 213 Queen's Road East",
```

```
        "addressLine2": "Wanchai, Hong Kong"
      },
      "tcPostalAddress": {
        "addressLine1": "香港灣仔皇后大道東 231 號",
        "addressLine2": "胡忠大廈 19 樓"
      },
      // last retrieval date of the e-bill
      // expressed in the number of milliseconds since 1970/1/1 GMT
      "lastRetrievalDate": 1560853718006
    },
    "addressDocFile": {
      // Base64 encoded string of the PDF e-bill file
      "docFile": "JVBERi0xLjUKJcK1wrXCtcK1CjEgMCBv...iag==",
      // SHA256 hash of PDF e-bill file
      "docHash": "69e3...c4e5",
      // bill date of the e-bill
      // expressed in the number of milliseconds since 1970/1/1 GMT
      "billDate": 1560849218006
    }
  }
}
```

3.7.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M99992",
  "msg": "Authorization failed. Applicant is not permitted to submit the request."
}
```

3.8. REQUEST DIGITAL SIGNING (CORPID-21)

RESTful API	https://<CorpID_domain>/api/v1/signing/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API making request for digital signing by sending accessToken and openID to CorpID system.

3.8.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart” system	For Sandbox, the value would only proceed as true. When iAMSign is true, there must be values here.
iAMSign	O	Boolean	true: integrated signing with “iAM Smart” is required false: only CorpID sign	Only for “iAM Smart” user; Default value is false.
hashCode	M	String	the document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	

CORPID SIMULATED API SPECIFICATION

sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
HKICHash	M (Conditional)	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing.	When iAMSign is true, there must be values here.
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign.	

			Maximum Length: 255	
stepupAuthMethod	O	String (Json)	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-up authentication method(s) in the e-service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field only for “iAM Smart” signing.	
stepupAuthPeriod	O	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field only for “iAM Smart” signing.	

3.8.2. Sample Request

```
{
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
```

```

    "accessToken": "XXXXXX",
    "openID": "XXXXXX",
    "clientID_iAM": "XXXXXX",
    "iAMSign": true,
    "hashCode": "XXXXXX",
    "HKICHash": "XXXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "documentName": "XXXXXX",
    "stepupAuthMethod": "{ \"unary\": [\"FR\"], \"unary\": [\"NFC\"], \"and\": [\"FR\", \"NFC\"] }",
    "stepupAuthPeriod": 720
  }
}

```

3.8.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

3.8.4. Sample Success Response

```

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

```

```
"code": "M00000",  
"msg": "",  
"content": {  
    "ticketID": "XXXXXX",  
    "authByQR": false  
}  
}
```

3.8.5. Sample Failed Response

```
{  
    "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
    "code": "M99992",  
    "msg": "Authorization failed. Applicant is not permitted to submit the request."  
}
```

3.9. REQUEST DIGITAL SIGNING (CORPID-21) (FOR APP E-SERVICE)

RESTful API	https://<CorpID_domain>/api/appv1/signing/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	False
Description	e-service can use this API making request for digital signing by sending accessToken and openID to CorpID system.

3.9.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart” system	For Sandbox, the value would only proceed as true. When iAMSign is true, there must be values here.
iAMSign	O	Boolean	true: integrated signing with “iAM Smart” is required false: only CorpID sign	Only for “iAM Smart” user; Default value is false.
clientRedirectURI	M (iOS only)	String	callback redirect URI. The value should be URL encoded and registered in the self-service portal.	
packageName	M (Android	String	If the state parameter is presented in the request message, the same state value	

	only)		will be returned to e-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by e-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
activityClass	M (Android only)	String	Activity class name of the e-service App for callback use.	
activityParams	M (Android only)	String	Optional params used during callback.	
serverRedirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
hashCode	M	String	the document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is	

			SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
HKICHash	M (Conditional)	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	

3.9.2. Sample Request

```
{
  "clientId": "XXXXXX",
```

```
"signatureMethod":"HmacSHA256",
"signature":"XXXXXX",
"timestamp": XXXXX,
"nonce":"XXXXXX",
// Clear text before encryption
"content":{
    "businessID": "XXXXXX",
    "accessToken":"XXXXXX",
    "openID": "XXXXXX",
    "clientId_iAM": "XXXXXX",
    "iAMSign": true,
    "source": "App_Package",
    "packageName": "com.onlineservice.myapp",
    "activityClass": "callback_activity",
    "activityParams": "callback_param",
    "serverRedirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>",
    "state": "XXXXXX",
    "hashCode": " XXXXX",
    "HKICHash": " XXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "documentName": "XXXXXX",
    "stepupAuthMethod": "{ \"unary\": [\"FR\"], { \"unary\": [\"NFC\"], { \"and\": [\"FR\", \"NFC\"] } }",
    "stepupAuthPeriod": 720
}
```

3.9.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default	

			empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

3.9.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}
```

3.9.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {hashCode}"
}
```

3.10. DIGITAL SIGNING CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Provide Simulation in Sandbox	True
Description	This callback will provide the digital signing result to e-service. e-service could treat the request as failed if it doesn't get callback within 18 minutes.

3.10.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
hashCode	O	String	The document hash submitted by e-service in the API request.	
timestamp	O	Long	Timestamp in milliseconds since January 1, 1970 00:00:00 GMT. CorpID System will provide this to e-service only when digital signing is successful.	
signature	O	String	Base64-encoded signature result string. CorpID System will provide this to e-service only when digital signing is successful.	
cert	O	String	Base64-encoded DER format certificate for the CorpID user.	

			CorpID System will provide this to e-service only when the digital signing is successful.	
accessToken	M (Conditional)	String	accessToken value for 2 nd signing	

3.10.2. Example Callback

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "businessID": "XXXXXX",
    "state": "XXXXXX",
    "hashCode": "XXXXXX",
    "timestamp": 1556450176000,
    "signature": "XXXXXX",
    "cert": "XXXXXX",
    "accessToken": "XXXXXX....."
  }
}
```

3.11. REQUEST PDF DIGITAL SIGNING (CORPID-21)

RESTful API	https://<CorpID_domain>/api/v1/pdfsinging/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API making request for digital signing by sending accessToken and openID to CorpID System.

3.11.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart” system	When corpIDSign is false, there must be values here.
source	M	String	Request initiator.	The supported source values refer to Appendix B
redirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen	

			are accepted.	
corpIDSign	M	Boolean	true: CorpID sign false: “iAM Smart” sign	
docDigest	M	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	
stepupAuthMethod	O	String (Json)	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-	

			up authentication method(s) in the e-service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field only for “iAM Smart” signing.	
stepupAuthPeriod	O	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field only for “iAM Smart” signing.	

3.11.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "businessID": "XXXXXX",
    "accessToken":"XXXXXX",
    "openID": "XXXXXX",
    "clientID_iAM": "XXXXXX",
    "source": "Android_Chrome",
```

```

    "redirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>",
    "state": "XXXXXX",
    "corpIDSign": false,
    "docDigest": "XXXXXX",
    "HKICHash": "XXXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "documentName": "XXXXXX",
    "stepupAuthMethod": "{\\"unary\\":[\"FR\\\"],\\"unary\\":
[\"NFC\\\"],\\"and\\":[\"FR\\\", \"NFC\\\"]}",
    "stepupAuthPeriod": 720
  }
}

```

3.11.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

3.11.4. Sample Success Response

```

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",

```

```
"msg": "",  
"content": {  
  "ticketID": "XXXXXX",  
  "authByQR": false  
}  
}
```

3.11.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M20002",  
  "msg": "empty parameter {docDigest}"  
}
```

3.12. REQUEST PDF DIGITAL SIGNING (CORPID-21) (FOR APP E-SERVICE)

RESTful API	https://<CorpID_domain>/api/appv1/pdfsigning/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	False
Description	e-service can use this API making request for PDF digital signing by sending accessToken and openID to CorpID system.

3.12.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart” system	For Sandbox, the value would only proceed as true. When iAMSign is true, there must be values here.
source	M	String	App_Link (iOS) or App_Package (Android)	
iAMSign	O	Boolean	true: integrated signing with “iAM Smart” is required false: only CorpID sign	Only for “iAM Smart” user; Default value is false.
clientRedirectURI	M (iOS only)	String	callback redirect URI. The value should be URL encoded and registered in the self-service portal.	

packageName	M (Android only)	String	If the state parameter is presented in the request message, the same state value will be returned to e-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by e-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
activityClass	M (Android only)	String	Activity class name of the e-service App for callback use.	
activityParams	M (Android only)	String	Optional params used during callback.	
serverRedirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
docDigest	M	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-	

			service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	

3.12.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "businessID": "XXXXXX",
    "accessToken":"XXXXXX",
    "openID": "XXXXXX",
    "clientID_iAM": "XXXXXX",
```

```

    "source": "App_Package",
    "serverRedirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>",
    "state": "XXXXXX",
    "iAMSign": true,
    "docDigest": "XXXXXX",
    "HKICHash": "XXXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "documentName": "XXXXXX",
    "stepupAuthMethod": "{ \"unary\": [\"FR\"], \"unary\": [\"NFC\"], \"and\": [\"FR\", \"NFC\"] }",
    "stepupAuthPeriod": 720
  }
}

```

3.12.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

3.12.4. Sample Success Response

```

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

```

```
"code": "M00000",  
"msg": "",  
"content": {  
  "ticketID": "XXXXXX",  
  "authByQR": false  
}  
}
```

3.12.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M20002",  
  "msg": "empty parameter {hashCode}"  
}
```

3.13. PDF DIGITAL SIGNING CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Provide Simulation in Sandbox	True
Description	This callback will provide the pdf digital signing result to e-service. e-service could treat the request as failed if it doesn't get callback within 18 minutes.

3.13.1. Callback Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
docDigest	O	String	The pdf document digest submitted by e-service in the API request.	
pdfSignature	O	String	Base64-encoded PKCS#7 object that is the actual PDF signature value. It contains signer's certificate, signed hash value, and the digital signing timestamp information. e-service can embed this value to the PDF document for future verification. "iAM Smart" System will provide this to e-service only when the digital signing is successful.	
accessToken	M	String	accessToken value for 2 nd signing	

	(Conditional)			
--	---------------	--	--	--

3.13.2. Example Callback

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M00000",  
  "msg": "",  
  "content": {  
    "businessID": "XXXXXX",  
    "state": "XXXXXX",  
    "docDigest": "XXXXXX",  
    "pdfSignature": "XXXXXX",  
    "accessToken": "XXXXXX....."  
  }  
}
```

3.14. ACKNOWLEDGES DIGITAL SIGNING RESULT

RESTful API	https://<CorpID_domain>/api/v1/signing/ackResult
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service calls this API to acknowledge CorpID System if the result of the digital signature is accepted or not.

3.14.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
signingResult	M	String	"SR001": digital signature is accepted "SR002": digital signature is rejected "SR003": no digital signature was received	

3.14.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "businessID":"XXXXXX",
```

```
"signingResult":"SR001"
}
}
```

3.14.3. Response Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	

3.14.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": ""
}
```

3.14.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {signingResult}"
}
```

3.15. REQUEST ANONYMOUS DIGITAL SIGNING (CORPID-12) (FOR E-SERVICE)

RESTful API	https://<CorpID_domain>/api/v1/signing/anonymous/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to initiate anonymous digital signing request by sending hashCode and HKICHash (for “iAM Smart” signing) to CorpID System.

3.15.1. Request Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
clientID_iAM	O (Con ditio nal)	String	Client ID value for “iAM Smart” system	For Sandbox, the value would only proceed as true. When iAMSign is true, there must be values here.
iAMSign	O	Boolean	true: integrated signing with “iAM Smart” is required false: only CorpID sign	Default value is false.
hashCode	M	String	the document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or	

			NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
HKICHash	M	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	
stepupAuthMethod	O	String (Json)	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-	

			up authentication method(s) in the e-service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field only for "iAM Smart" signing.	
stepupAuthPeriod	O	Interger	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field only for "iAM Smart" signing.	

3.15.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "businessID": "XXXXXX",
    "clientID_iAM": " XXXXX",
    "iAMSign": true,
    "hashCode": " XXXXX",
    "HKICHash": " XXXXX",
```

```

    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "documentName": "XXXXXX",
    "stepupAuthMethod": "{ \"unary\": [ \"FR\" ], \"unary\": [ \"NFC\" ] }, { \"and\": [ \"FR\", \"NFC\" ] }",
    "stepupAuthPeriod": 720
  }
}

```

3.15.3. Response Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
ticketID	M	String	ticketID is a unique identifier that will be used to identify this request in the following steps of the digital signing workflow. The ticketID will be expired 12 minutes after issuance.	

3.15.4. Sample Success Response

```

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",

```

```
"content": {  
  "businessID": "XXXXXX",  
  "ticketID": "XXXXXX"  
}  
}
```

3.15.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "D20102",  
  "msg": "empty parameter {hashCode}"  
}
```

3.16. REQUEST ANONYMOUS PDF DIGITAL SIGNING – (CORPID-13) (FOR E-SERVICE)

RESTful API	https://<CorpID_domain>/api/v1/pdfsigning/anonymous/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API making request for digital signing by sending accessToken and openID to CorpID System.

3.16.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
clientID_iAM	O (Conditional)	String	Client ID value for “iAM Smart” system	For Sandbox, the value would only proceed as true. When iAMSign is true, there must be values here.
iAMSign	O	Boolean	true: integrated signing with “iAM Smart” is required false: only CorpID sign	Only for “iAM Smart” user; Default value is false. For Sandbox, the value would only proceed as true.
docDigest	M	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64	

			encoded.	
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	
stepupAuthMethod	O	String (Json)	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-up authentication method(s) in the e-service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field is only for “iAM	

			Smart” signing.	
stepupAuthPeriod	O	Interger	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field is only for “iAM Smart” signing.	

3.16.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "businessID": "XXXXXX",
    "clientID_iAM": "XXXXXX",
    "iAMSign": true,
    "docDigest": "XXXXXX",
    "HKICHash": "XXXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "documentName": "XXXXXX",
    "stepupAuthMethod": "{ \"unary\": [\"FR\"], { \"unary\": [\"NFC\"], { \"and\": [\"FR\", \"NFC\"] } }",
    "stepupAuthPeriod": 720
  }
}
```

3.16.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
businessID	M	String	businessID is a unique identifier for e-service to differentiate different requests. e-service can use the businessID to relate the callback message with the original request.	
ticketID	M (Conditional)	String	ticketID is a unique identifier that will be used to identify this request in the following steps of the digital signing workflow. The ticketID will be expired 12 minutes after issuance.	

3.16.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "businessID": "XXXXXX",
    "ticketID": "XXXXXX"
  }
}
```

3.16.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
```

```
"msg": "empty parameter {hashCode}"  
}
```

3.17. ANONYMOUS DIGITAL SIGNING CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	GET
Provide Simulation in Sandbox	True
Description	This callback is used to pass authCode to e-service Server. The URI must be registered in the self-service portal.

3.17.1. Callback Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID will only be returned in anonymous workflows. businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
ticketID	M (Conditional)	String	The authorisation code is generated by the “iAM Smart” System. The authorisation code will be expired 1 minute after issuance. e-service MUST NOT use the authorisation code more than once. An error message will be returned if an authorisation code is expired or re-used.	
error_code	M (Conditional)	String	Return error code when exception.	

3.17.2. Example Callback

Allow

// Line breaks are for legibility only.

GET

https://<callback_endpoint>

?businessID=XXXXXX

?ticketID=0ad186353c424c64897fcc00445c9ba1

&state=eddd527b6

Deny

// Line breaks are for legibility only.

GET

https://<callback_endpoint>

?error_code=D20001

&state=eddd527b6

3.18. GET ANONYMOUS DIGITAL SIGNING RESULT

RESTful API	https://<CorpID_domain>/api/v1/signing/anonymous/getResult
Request Type	POST
Service Version	1.0.0
Description	e-service can use this API to retrieve anonymous digital signing result.

3.18.1. Request Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different requests. e-service can use the businessID to relate the callback message with the original request.	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	

3.18.2. Sample Request

```
{
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
    "accessToken": "XXXXXX",
    "openID": "XXXXXX"
  }
}
```

3.18.3. Response Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
hashCode	O	String	The document hash submitted by e-service in the API request.	
timestamp	O	Long	Timestamp in milliseconds since January 1, 1970 00:00:00 GMT. CorpID System will provide this to e-service only when digital signing is successful.	
signature	O	String	Base64-encoded signature result string. CorpID System will provide this to e-service only when digital signing is	

			successful.	
cert	O	String	Base64-encoded DER format certificate for the CorpID user. CorpID System will provide this to e-service only when the digital signing is successful.	
accessToken	M (Conditional)	String	accessToken value for 2 nd signing	

3.18.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "businessID": "XXXXXX",
    "hashCode": "XXXXXX",
    "timestamp": 1556450176000,
    "signature": "XXXXXX",
    "cert": "XXXXXX",
    "accessToken": "XXXXXX....."
  }
}
```

3.18.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M99992",
  "msg": "Authorization failed. Applicant is not permitted to submit the request."
}
```

3.19. GET ANONYMOUS PDF DIGITAL SIGNING RESULT

RESTful API	https://<CorpID_domain>/api/v1/pdfsigning/anonymous/getResult
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to retrieve anonymous PDF digital signing result.

3.19.1. Request Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different requests. e-service can use the businessID to relate the callback message with the original request.	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	

3.19.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "businessID":"XXXXXX",
    "accessToken":"XXXXXX",
    "openID": "XXXXXX"
```

```

    }
}

```

3.19.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
businessID	M	String	businessID is a unique identifier for e-service to differentiate different requests. e-service can use the businessID to relate the callback message with the original request.	
docDigest	O	String	The pdf document digest submitted by e-service in the API request.	
pdfSignature	O	String	Base64-encoded PKCS#7 object that is the actual PDF signature value. It contains the signer's certificate, signed hash value, and the digital signing timestamp information. e-service can embed this value to the PDF document for future verification. CorpID System will provide this to e-service only when the digital signing is successful.	
accessToken	M (Conditional)	String	accessToken value for 2 nd signing	

3.19.4. Sample Success Response

```

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",

```

```
"msg": "",
"content": {
  "businessID": "XXXXXX",
  "docDigest": "tGzv3JOkF0XG5Qx2TlKWIA.....",
  "pdfSignature": "nnoadisauflanehykdjf.....",
  "accessToken": "XXXXXX....."
}
}
```

3.19.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {accessToken}"
}
```

3.20. REQUEST BULK DIGITAL SIGNING

RESTful API	https://<CorpID_domain>/api/v1/bdss/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to make request for bulk digital signing by sending accessToken and openID to CorpID system.

3.20.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart”	When corpIDSign is false, there must be values here
redirectURI	M	String	Callback URI for the endpoint that e-service uses to receive BSQC Token	
callbackResultURI	M	String	Callback URI for the endpoint that e-service uses to receive bulk digital signing result	
corpIDSign	M	Boolean	true: CorpID sign false: “iAM Smart” sign	
source	M	String	Request initiator. The supported source values can be found in Appendix B of this specification.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-	

			service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
requestName	M	String	The bulk digital signing request name that will be displayed on the consent page of the CorpID mini-program. Maximum Length: 255	
maxCallbackSigs	O	Integer	The maximal number of signatures that the e-service endpoint can accept in one callback for receiving the bulk digital signing result. The default value is 1000.	

stepupAuthMethod	O	String (Json)	{ "unary": ["FR"] }, { "unary": ["NFC"] }, { "and": ["FR", "NFC"] }. e-service should have been granted the requested step-up authentication method(s) in the e-service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before.	
stepupAuthPeriod	O	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720	
documents	M	Array of JSON	The document hashes and/or pdf digests that are to be signed by CorpID. The number of array items should be greater than 0 and less than 1001. The array items are in one of the following JSON dictionary formats: For document hash { "id": "egh...cva", "documentName": "Document", "hashCode": "965...56d", "sigAlgo": "SHA256withRSA" } For PDF document digest { "id": "egh...cva", "documentName": "PDF Document", "docDigest": "965...56d" } The following rows show the detail of each dictionary key	
Details for “documents”				
id	M	String	The document unique identifier. It should be ASCII string with length less than or equal to 36	

			chars.	
documentName	M	String	The document name that will be displayed in the CorpID mini-program when asking user's consent for the bulk digital signing request. Maximum Length: 255	
hashCode	M (Conditional)	String	The document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
docDigest	M (Conditional)	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	

3.20.2. Sample Request

```
{
  "clientId":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
```

```
"content":{
    "businessID": "XXXXXX",
    "accessToken": "XXXXXX",
    "openID": "XXXXXX",
    "clientID_iAM": "XXXXXX",
    // For e-service
    "redirectURI": "https://<rp_domain>/<rp_context>/<call_back_endpoint>",
    "callbackResultURI": "https://<rp_domain>/<rp_context>/<endpoint>",
    // For iAM Smart
    "redirectURI": "https://<CorpID_domain>/api/v1/bdss/callback",
    "callbackResultURI": "https://<CorpID_domain>/api/v1/bdss/signingResult",
    "source": "Android_Chrome",
    "state": "XXXXXX",
    "HKICHash": "XXXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "requestName": "XXXXXX",
    "maxCallbackSigs": 200,
    "stepupAuthMethod": "{ \"unary\": [ \"FR\" ], \"unary\": [ \"NFC\" ], \"and\": [ \"FR\", \"NFC\" ] }",
    "stepupAuthPeriod": 720,
    "documents": [
        {
            "id": "aef...cth",
            "documentName": "Document",
            "hashCode": "965e...356d",
            "sigAlgo": " SHA256withRSA"
        },
        {
            "id": "trx...hnm",
            "documentName": "PDF Document",
            "docDigest": "7rtg...y67d"
        },
        .....
    ]
}
```

```
}

```

3.20.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

3.20.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}
```

3.20.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
```

```
"msg": "empty parameter {documents}"  
}
```

3.21. REQUEST BULK DIGITAL SIGNING (FOR APP E-SERVICE)

RESTful API	https://<CorpID_domain>/api/appv1/bdss/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	False
Description	e-service can use this API to make request for bulk digital signing by sending accessToken and openID to CorpID system.

3.21.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart” When corpIDSign is false, there must be values here	
clientRedirectURI	M (iOS only)	String	callback redirect URI. The value should be URL encoded and registered in the self-service portal.	
packageName	M (Android only)	String	If the state parameter is presented in the request message, the same state value will be returned to e-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by e-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens	

			are accepted.	
activityClass	M (Android only)	String	Activity class name of the e-service App for callback use.	
activityParams	M (Android only)	String	Optional params used during callback.	
serverRedirectURI	M	String	callback URI.	
callbackResultURI	M	String	Callback URI for the endpoint that e-service uses to receive bulk digital signing result	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	

serviceName	M	String	The e-service name. Maximum Length: 255	
requestName	M	String	The bulk digital signing request name that will be displayed on the consent page of the CorpID mini-program. Maximum Length: 255	
maxCallbackSigs	O	Integer	The maximal number of signatures that the e-service endpoint can accept in one callback for receiving the bulk digital signing result. The default value is 1000.	
stepupAuthMethod	O	String (Json)	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-up authentication method(s) in the e-service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before.	
stepupAuthPeriod	O	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720	
documents	M	String (Json)	The document hashes and/or pdf digests that are to be signed by CorpID. The number of array items should be greater than 0 and	

			less than 1001. The array items are in one of the following JSON dictionary formats: For document hash { "id": "egh...cva", "documentName": "Document", "hashCode": "965...56d", "sigAlgo": "SHA256withRSA" } For PDF document digest { "id": "egh...cva", "documentName": "PDF Document", "docDigest": "965...56d" } The following rows show the detail of each dictionary key	
Details for “documents”				
id	M	String	The document unique identifier. It should be ASCII string with length less than or equal to 36 chars.	
documentName	M	String	The document name that will be displayed in the CorpID mini-program when asking user’s consent for the bulk digital signing request. Maximum Length: 255	
hashCode	M (Conditional)	String	The document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
docDigest	M	String	The PDF document digest to be signed. e-service should compute	

	(Conditional)		the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	
--	---------------	--	--	--

3.21.2. Sample Request

```
{
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
    "accessToken": "XXXXXX",
    "openID": "XXXXXX",
    "clientID_iAM": "XXXXXX",
    // For e-service
    "serverRedirectURI": "https://<rp_domain>/<rp_context>/<call_back_endpoint>",
    "callbackResultURI": "https://<rp_domain>/<rp_context>/<endpoint>",
    // For iAM Smart
    "serverRedirectURI": "https://<CorpID_domain>/api/v1/bdss/callback",
    "callbackResultURI": "https://<CorpID_domain>/api/v1/bdss/signingResult",
    "source": "App_Package",
    "packageName": "com.onlineservice.myapp",
    "activityClass": "callback_activity",
    "activityParams": "callback_param",
    "state": "XXXXXX",
    "HKICHash": "XXXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "requestName": "XXXXXX",
    "maxCallbackSigs": 200,
    "stepupAuthMethod": "{ \"unary\": [\"FR\"], \"unary\": [\"NFC\"], \"and\": [\"FR\", \"NFC\"] }",

```

```

    "stepupAuthPeriod": 720,
    "documents": [
      {
        "id": "aef...cth",
        "documentName": "Document",
        "hashCode": "965e...356d",
        "sigAlgo": " SHA256withRSA"
      },
      {
        "id": "trx...hnm",
        "documentName": "PDF Document",
        "docDigest": "7rtg...y67d"
      },
      .....
    ]
  }
}

```

3.21.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

3.21.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}
```

3.21.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {documents}"
}
```

3.22. CALLBACK TO RECEIVE BSQC TOKEN

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	This callback will provide the BSQC Token to e-service.

3.22.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
BSQCToken	M	String	The BSQC Token that can be used by the corresponding e-service to enquire the status of the submitted bulk digital signing request or cancel the corresponding request	

3.22.2. Example Callback

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "clientID": "XXXXXX",
```

```
"signatureMethod":"HmacSHA256",
"signature":"XXXXXX",
"timestamp": XXXXX,
"nonce":"XXXXXX",
// Clear text before encryption
"content":{
    "businessID":"XXXXXX",
    "state":"XXXXXX",
    "BSQCToken": "XXXXXX"
}
}
```

3.23. BULK DIGITAL SIGNING RESULT CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	This callback will provide the bulk digital signing result to e-service.

3.23.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
cert	O	String	Base64-encoded DER format certificate for the “iAM Smart” user. “iAM Smart” System will provide this to e-service only when the digital signing is successful.	
totalCallbacks	O (Conditional)	Integer	If maxCallbackSigs (m) is less than the number of hashes and digests (n) when submitting request], the digital signing results of these documents will be called back in totalCallbacks = $\text{int}((n-1) / m + 1)$ times. m signatures will be returned in each of the first totalCallbacks-1 callbacks. n-	

			(totalCallbacks-1)*m signatures will be returned in the last callback. If totalCallbacks is greater than 1, it must be specified. The default value is 1.	
callbackSeq	O (Conditional)	Integer	If totalCallbacks is greater than 1, this callbackSeq should be 1, 2, ..., totalCallbacks. If totalCallbacks is greater than 1, this callbackSeq must be specified. The default value is 1.	
totalSignNum	O (Conditional)	Integer	The Number of signed documents. It should be equal to the number of documents submitted by the user.	
failSignNum	O (Conditional)	Integer	The Number of signed failed documents. It should be less than the number of documents submitted by users.	
accessToken	M (Conditional)	String	accessToken value for 2 nd signing	
sigFile (For Hash Signature)				
id	M	String	The document unique identifier. It should be ASCII string with length less than or equal to 36 chars.	
hash	M	String	The document hash submitted by e-service in the API request	
timestamp	M (Conditional)	Long	Timestamp in milliseconds since January 1, 1970 00:00:00 GMT. CorpID System will provide this to e-service only when digital signing is successful.	
signature	M (Conditional)	String	Base64-encoded signature result string. “iAM Smart” System will provide this to e-service only when digital signing is successful.	
sigFile (For PDF Signature)				
id	M	String	The document unique identifier. It should be ASCII string with length less than or equal to 36	

			chars.	
docDigest	M	String	The pdf document digest submitted by e-service in the API request	
pdfSignature	M (Conditional)	String	Base64-encoded PKCS#7 object that is the actual PDF signature value. It contains signer's certificate, signed hash value, and the digital signing timestamp information. e-service can embed this value to the PDF document for future verification. "iAM Smart" System will provide this to e-service only when the digital signing is successful.	

3.23.2. Example Callback (Signing Success)

// Callback Header

// The \$<boundary> looks like --ZJpuW5l0YYKRmTVwkt76oSIv9-pw8Cm0nW9

// It is automatically generated and is different in every callback

Content-Type: multipart/form-data;boundary=\$<boundary>

// Callback Body

\$<boundary>

Content-Disposition: form-data; name="sigFile"; filename="signatures.txt"

Content-Type: text/plain

Content-Length: 62892

// The content of the text file looks like the following.

// Line breaks are for legibility only.

```
"signatures": [
  {
    "id": "egh...cva",
    "hash": "tGzv3JOkF0XG5Qx2TIKWIA",
    "timestamp": 1556450176000,
    "signature": "nnoadisauflanehykdjf"
  },
  .....
  {
    "id": "hrf...bnd",
```

```
"docDigest": "tGzv3JOkF0XG5Qx2TIKWIA",
"pdfSignature": "sdfGSDGsdfaGDEHfjsgQG.....GSGjljlkwjmh"
}
]
$<boundary>
Content-Disposition: form-data; name="sigMetada"
Content-Type: application/json;charset=UTF-8
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "D00000",
  "message": "SUCCESS",
  "content": {
    "businessID": "2YotnFZFEjrlzCsicMWpAA",
    "state": "unesidkd",
    "cert": "sdfGSDGsdfaGDEHfjsgQG.....GSGjljlkwjmh",
    "totalCallbacks": 10,
    "callbackSeq": 2,
    "totalSignNum": 100,
    "failSignNum": 20,
    "accessToken": "XXXXX....."
  }
}
$<boundary>
```

3.23.3. Example Callback (Signing Failure)

```
// Callback Header
// The $<boundary> looks like --ZJpuW5l0YYKRmTVwkt76oSIv9-pw8Cm0nW9
// It is automatically generated and is different in every callback
Content-Type: multipart/form-data;boundary=$<boundary>
// Callback Body
$<boundary>
Content-Disposition: form-data; name="sigMetada"
Content-Type: application/json;charset=UTF-8
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```
"code": "D71002",  
"message": "failed to request signing",  
"content": {  
  "businessID": "2YotnFZFEjrlzCsicMWpAA",  
  "state": "unesidkd",  
}  
}  
$<boundary>
```

3.24. ACKNOWLEDGES BULK DIGITAL SIGNING RESULT

RESTful API	https://<CorpID_domain>/api/v1/bdss/ackResult
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service calls this API to acknowledge CorpID System if the result of the bulk digital signature is accepted or not. e-service should send the acknowledgement regardless of whether the digital signing result (Section 3.24) is success or failure.

3.24.1. Request Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart”	
signingResult	M	String	"SR001": digital signature is accepted "SR002": digital signature is rejected "SR003": no digital signature was received	
signatures	M (Conditional)	ListArray	When signingResult has no value, there must be values here	
Details for “signatures”				
id	M	String	The document unique identifier. It should be ASCII string with length less than or equal to 36 chars.	
signingResult	M	String	"SR001": digital signature is accepted "SR002": digital	

			signature is rejected "SR003": no digital signature was received	
--	--	--	--	--

3.24.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "businessID": "XXXXXX",
    "ClientID_iAM": "XXXXXX",
    "signingResult": "SR001"
  }
}
or
"content":{
  "businessID": "XXXXXX",
  "ClientID_iAM": "XXXXXX",

  "signingResult": "",
  "signatures": [
    {
      "id": "egh...cva",
      "signingResult": "SR001"
    },
    {
      "id": "egh...cva",
      "signingResult": "SR002"
    },
    {
      "id": "egh...cva",
      "signingResult": "SR003"
    }
  ]
}
```

```
]
}
}
```

3.24.3. Response Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	

3.24.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": ""
}
```

3.24.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {signingResult}"
}
```

3.25. ENQUIRE BULK DIGITAL SIGNING STATUS

RESTful API	https://<CorpID_domain>/api/v1/bdss/enquireStatus
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service calls this API to enquire the digital signing status of the submitted bulk digital signing request.

3.25.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
BSQCToken	M	String	The BSQC Token which is returned in section 3.23. Maximum Length: 32	
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
openID	M	String	Tokenised ID value Maximum Length: 64	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart”	

3.25.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
```

```

    "content":{
        "BSQCToken":"XXXXXX",
        "businessID": "XXXXXX",
        "openID": "XXXXXX",
        "clentID_iAM": "XXXXXX"
    }
}

```

3.25.3. Response Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	
status	M	String	The digital signing status of the submitted bulk digital signing request.	Its value should be: “SS001”: Pending for Signing “SS002”: Signing in Progress “SS003”: Signing Completed “SS004”: Signing Cancelled “SS005”: Signing Failed “SS006”: No Record Found

3.25.4. Sample Success Response

```

{
    "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
    "code": "M00000",
    "msg": "",
    "content": {

```

```
      "status": "SS003"
    }
  }
```

3.25.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "D20102",
  "msg": "empty parameter {BSQCToken}"
}
```

3.26. CANCEL BULK DIGITAL SIGNING REQUEST

RESTful API	https://<CorpID_domain>/api/v1/bdss/cancelRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service calls this API to cancel the submitted bulk digital signing request. Only the request of which status is “Pending for Signing” can be cancelled.

3.26.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
BSQCToken	M	String	The BSQC Token which is returned in section 3.23 Maximum Length: 32	
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
openID	M	String	Tokenised ID value Maximum Length: 64	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart”	

3.26.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
```

```
"content":{
    "BSQCToken":"XXXXXX",
    "businessID": "XXXXXX",
    "openID": "XXXXXX",
    "clentID_iAM": "XXXXXX"
}
```

3.26.3. Response Parameters

Field Name	Mandatory/O ptional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	

3.26.4. Sample Success Response

```
{
    "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
    "code": "M00000",
    "msg": ""
}
```

3.26.5. Sample Failed Response

```
{
    "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
    "code": "D21002",
    "msg": "the request to be cancelled is not Pending for Signing"
}
```

3.27. REQUEST ANONYMOUS BULK DIGITAL SIGNING

RESTful API	https://<CorpID_domain>/api/v1/bdss/anonymous/initiateRequest
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to submit anonymous bulk digital signing request by sending documents and HKICHash to “iAM Smart” System.

3.27.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
callbackResultURI	M	String	Callback URI for the endpoint that e-service uses to receive bulk digital signing result	
corpIDSign	M	Boolean	true: CorpID sign false: “iAM Smart” sign	
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded.	

CORPID SIMULATED API SPECIFICATION

department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
requestName	M	String	The bulk digital signing request name that will be displayed on the consent page of the CorpID mini-program. Maximum Length: 255	
maxCallbackSignatures	O	Integer	The maximal number of signatures that the e-service endpoint can accept in one callback for receiving the bulk digital signing result. The default value is 1000.	
stepupAuthMethod	O	String (Json)	<pre>{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}</pre> . e-service should have been granted the requested step-up authentication method(s) in the e-service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before.	
stepupAuthPeriod	O	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720	

documents	M	String (Json)	The document hashes and/or pdf digests that are to be signed by CorpID. The number of array items should be greater than 0 and less than 1001. The array items are in one of the following JSON dictionary formats: For document hash { "id": "egh...cva", "documentName": "Document", "hashCode": "965...56d", "sigAlgo": "SHA256withRSA" } For PDF document digest { "id": "egh...cva", "documentName": "PDF Document", "docDigest": "965...56d" } The following rows show the detail of each dictionary key	
Details for “documents”				
id	M	String	The document unique identifier. It should be ASCII string with length less than or equal to 36 chars.	
documentName	M	String	The document name that will be displayed in the CorpID mini-program when asking user’s consent for the bulk digital signing request. Maximum Length: 255	
hashCode	M (Conditional)	String	The document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2,	

			hashCode provided must be hashed with SM3.	
docDigest	M (Conditional)	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	

3.27.2. Sample Request

```
{
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": "XXXXXX",
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
    // For e-service
    "callbackResultURI": "https://<rp_domain>/<rp_context>/<endpoint>",
    // For iAM Smart
    "callbackResultURI": "https://<CorpID_domain>/api/v1/bdss/signingResult",
    "state": "XXXXXX",
    "HKICHash": "XXXXXX",
    "department": "XXXXXX",
    "serviceName": "XXXXXX",
    "requestName": "XXXXXX",
    "maxCallbackSigs": 200,
    "stepupAuthMethod": "{ \"unary\": [\"FR\"], \"unary\": [\"NFC\"], \"and\": [\"FR\", \"NFC\"] }",
    "stepupAuthPeriod": 720,
    "documents": [
      {
        "id": "aef...cth",
        "documentName": "Document",
        "hashCode": "965e...356d",
```

```

        "sigAlgo": " SHA256withRSA"
    },
    {
        "id": "trx...hnm",
        "documentName": "PDF Document",
        "docDigest": "7rtg...y67d"
    },
    .....
]
}
}

```

3.27.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

3.27.4. Sample Success Response

```

{
    "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
    "code": "M00000",
    "msg": "",
    "content": {

```

```
    "ticketID": "XXXXXX",  
    "authByQR": false  
  }  
}
```

3.27.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M20002",  
  "msg": "empty parameter {documents}"  
}
```

Callback please refer to 3.25.

3.28. REQUEST BSQC TOKEN

RESTful API	https://<CorpID_domain>/api/v1/bdss/getBSQCToken
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service uses this API to get BSQC Token for the submitted bulk digital signing request.

3.28.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value Maximum Length: 64	
clientID_iAM	M (Conditional)	String	Client ID value for “iAM Smart”	

3.28.2. Sample Request

```
{
  "clientID":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "accessToken ":"XXXXXX",
    "openID": "XXXXXX",
    "clentID_iAM": "XXXXXX"
  }
}
```

3.28.3. Response Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	
BSQCToken	M	String	The BSQC Token that can be used by the corresponding e-service to enquire or cancel the submitted bulk digital signing request.	

3.28.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "BSQCToken": "XXXXXX"
  }
}
```

3.28.5. Sample Failed Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "D20102",
  "msg": " empty parameter {accessToken}"
}
```

4. API SPECIFICATION OF CORPID

4.1. AUTHENTICATION BY REQUESTING QR PAGE/ BROKER PAGE

RESTful API	https://<CorpID_domain>/api/v1/auth/getQR
Request Type	GET
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service calls this API to get the QR/App broker page or QR page. After the user authorises login or anonymous request on the CorpID mini-program, the page will be redirected to the redirectURI with authCode and state parameters. If the user denies it, only the state parameter will be redirected.

4.1.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
Query Param				
clientID	M	String	e-service client identifier. The clientID will be assigned to e-service at the initial registration.	
clientID_iAM	M (Conditional)	String	The e-service's client ID of "iAM Smart" system When iamFormfill is true, there must be values here	
responseType	M	String	The value MUST be set to code.	
redirectURI	M	String	callback URI.	
source	M	String	Request initiator.	The supported source values refer to Appendix B
ticketID	M (Conditional)	String	Required only in anonymous form filling or anonymous digital signing workflows	

accessToken	M (Conditional)	String	Required only in select corporation from token exchange workflows	
lang	O	String	Language to display: en-US, zh-HK, or zh-CN. If this parameter is not specified, zh-HK will be shown.	
state	O	String	If the state parameter is presented in the request message, the same state value will be returned to e-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by e-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
scope	M	String	The value should be URL encoded.	
brokerPage	O	Boolean	If brokerPage is set to true, DeepLink will be leveraged to open the CorpID mini-program. This feature is useful for e-service supporting mobile web versions while triggering the CorpID mini-program or showing a QR page automatically. The default value is false.	

4.1.2. Sample Request

For mini-program

```
<Deeplink_URL>://<CorpID_domain>/path
?clientId=XXXXXX
```

?clientID_iAM=XXXXXX
&responseType=code
&source=Android_Chrome
&redirectURI=XXXXXX
&scope=XXXXXX
&lang=en-US
&state=XXXXXX

https://<CorpID_domain>/api/v1/auth/getQR
?clientID=XXXXXX
&responseType=code
&source=Android_Chrome
&redirectURI=XXXXXX
&scope=XXXXXX
&lang=en-US
&state=XXXXXX

4.2. REQUEST DOCUMENT LIST IN DOCUMENT WALLET

RESTful API	https://<CorpID_domain>/api/v1/corp/requestDocList
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API making request for document list from Document Wallet by sending accessToken and openID to CorpID System.

4.2.1. Request Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
redirectURI	M	String	callback URI.	

4.2.2. Sample Request

```
{
  "clientId": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "accessToken": "XXXXXX",
    "openID": "XXXXXX",
    "redirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>"
  }
}
```

4.2.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

4.2.4. Sample Success Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M00000",  
  "msg": "",  
  "content": {  
    "ticketID": "XXXXXX",  
    "authByQR": false  
  }  
}
```

4.2.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M99992",  
  "msg": " Authorization failed. Applicant is not permitted to submit the request."
```

}

4.3. CALLBACK TO RECEIVE DOCUMENT LIST

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	This callback will provide the document list to e-service.

4.3.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
docList	M	Object Array		
Details of “docList”				
docID	M	Integer	The ID of documents in Document Wallet	

docName	M	String	The Name of documents in Document Wallet	
docDesc	M	String	Document description	
docCrtTs	M	String	The create datetime of the documents in Document Wallet	
docLstUpdTs	M	String	The last update datetime of the documents in Document Wallet	
docType	M	String	The type of document	

4.3.2. Example Callback

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": 1757653929,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
    "state": "XXXXXX",
    "docID": "000001",
    "docName": "XXXXXX",
    "docDesc": "XXXXXX",
    "docCrtTs": "2025-01-01 15:00:00",
    "docLstUpdTs": "2025-01-01 15:00:00",
    "docType": "XXXXXX"
  }
}
```

4.4. GET DOCUMENT DETAIL

RESTful API	https://<CorpID_domain>/api/v1/corp/getDocDetail
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to get document detail content with docID.

4.4.1. Request Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
docID	M	String	The ID of documents in Document Wallet	

4.4.2. Sample Request

```
{
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": XXXXX,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "accessToken": "XXXXXX",
    "openID": "XXXXXX",
    "docID": "XXXXXX"
  }
}
```

4.4.3. Response Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	
docID	M	Integer	The ID of documents in Document Wallet	
docName	M	String	Documents name	
docDesc	M	String	Document description	
docCrtTs	M	String	The create datetime of the documents in Document Wallet	
docLstUpdTs	M	String	The last update datetime of the documents in Document Wallet	
docType	M	String	The type of document	
docContent	M	String	The document content in Base64 format	

4.4.4. Sample Success Response

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "docID": "000001",
    "docName": "XXXXXX",
    "docDesc": "XXXXXX",
    "docCrtTs": "2025-01-01 15:00:00",
    "docLstUpdTs": "2025-01-01 15:00:00",
    "docType": "XXXXXX",
    "docContent": "XXXXXXXXXXXXXXXXXXXX...."
  }
}
```

```
}  
}
```

4.4.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M99992",  
  "msg": " Authorization failed. Applicant is not permitted to submit the request."  
}
```

4.5. UPLOAD DOCUMENT

RESTful API	https://<CorpID_domain>/api/v1/corp/uploadDoc
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API to upload document to Document Wallet

4.5.1. Request Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
docName	M	String	Document name	
docDesc	M	String	Document description	
docType	M	String	The type of document	
docContent	M	String	The document content in Base64 format	

4.5.2. Sample Request

```
{
  "clientId":"XXXXXX",
  "signatureMethod":"HmacSHA256",
  "signature":"XXXXXX",
  "timestamp": XXXXX,
  "nonce":"XXXXXX",
  // Clear text before encryption
  "content":{
    "accessToken":"XXXXXX",
    "openID": "XXXXXX",
```

```
"docName": "XXXXXX",  
"docDesc": "XXXXXX",  
"docType": "XXXXXX",  
"docContent": "XXXXXXXXXX....."  
}  
}
```

4.5.3. Response Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix
msg	O	String	Error message. (Return value if the result is failed. Default empty string)	

4.5.4. Sample Success Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M00000",  
  "msg": ""  
}
```

4.5.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M99992",  
  "msg": " Authorization failed. Applicant is not permitted to submit the request."  
}
```

4.6. REQUEST DOCUMENT SHARING

RESTful API	https://<CorpID_domain>/api/v1/corp/requestDocSharing
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	e-service can use this API making request for document(s) sharing from Document Wallet by sending accessToken and openID to CorpID System.

4.6.1. Request Parameters

Field Name	Man dator y/Opt ional	Data Type	Description	Values/Remark
JSON Object				
accessToken	M	String	accessToken value	
openID	M	String	Tokenised ID value	
docIDList	M	String Array	The list of ID of documents in Document Wallet	
redirectURI	M	String	callback URI.	

4.6.2. Sample Request

```
{
  "clientId": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": 1757653929,
  "nonce": "XXXXXX",
  "content": {
    "accessToken": "XXXXXX",
    "openID": "XXXXXX",
    "docID": [
      "XXXXXX",
      "XXXXXX"
    ]
  }
}
```

```

    ],
    "redirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>"
  }
}

```

4.6.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
code	M	String	The response code	Refer to Appendix D
msg	M	String	Error message. (Return value if the result is failed. Default empty string)	
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

4.6.4. Sample Success Response

```

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}

```

4.6.5. Sample Failed Response

```
{  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  "code": "M99992",  
  "msg": " Authorization failed. Applicant is not permitted to submit the request."  
}
```

4.7. CALLBACK TO RECEIVE DOCUMENT SHARING

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Service Version	1.0.0
Provide Simulation in Sandbox	True
Description	This callback will provide the document(s) from Document Wallet to e-service.

4.7.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
docList	M	Object Array		
Details of “docList”				
docID	M	Integer	The ID of documents in Document Wallet	

docName	M	String	Documents name	
docDesc	M	String	Document description	
docCrtTs	M	String	The create datetime of the documents in Document Wallet	
docLstUpdTs	M	String	The last update datetime of the documents in Document Wallet	
docType	M	String	The type of document	
docContent	M	String	Document content in Base64 encoded	

4.7.2. Example Callback

```
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "clientID": "XXXXXX",
  "signatureMethod": "HmacSHA256",
  "signature": "XXXXXX",
  "timestamp": 1757653929,
  "nonce": "XXXXXX",
  // Clear text before encryption
  "content": {
    "businessID": "XXXXXX",
    "state": "XXXXXX",
    "docList": [
      {
        "docID": "000001",
        "docName": "XXXXXX",
        "docDesc": "XXXXXX",
        "docCrtTs": "2025-01-01 15:00:00",
        "docLstUpdTs": "2025-01-01 15:00:00",
        "docType": "XXXXXX",
        "docContent": "XXXXX....."
      },
      .....
    ]
  }
}
```

```
]
}
}
```

5. APPENDIX A

5.1. DETAILS OF “CORPUSERPROFILEFIELDS”

corpProfileFields	Description
id_cty_issue	ISO 3166 3-letter country code
id_type	4. Identity card 5. Passport 6. 3. Other identity type

5.2. DETAILS OF “CORPPROFILEFIELDS”

corpProfileFields	Description
corpID	The unique ID from "CorpID" system
brn	Business Registration Number
corpNameEN	Corporation Name (EN)
corpNameTC	Corporation Name (TC)
corpNameSC	Corporation Name (SC)
corpAddr	Corporation Business Address

5.3. DETAILS OF “ECORPFIELDS”

eCorpFields	Description
corpID	The unique ID from "CorpID" system
brn	Business Registration Number
corpNameEN	Corporation Name (EN)
corpNameTC	Corporation Name (TC)
corpNameSC	Corporation Name (SC)
corpAddr	Corporation Business Address
corpTypeEN	Corporation Type (EN)
corpTypeTC	Corporation Type (TC)
corpTypeSC	Corporation Type (SC)
corpTel	Corporation Telephone No.
corpStatusEN	Corporation Active Status (EN)

corpStatusTC	Corporation Active Status (TC)
corpStatusSC	Corporation Active Status (SC)
placeOfIncorp	Place of incorporation
dateOfReg	Date of registration

5.4. DETAILS OF “PROFILEFIELDS” AND “EMEFIELDS”

For the details of “profileFields” and “eMEFields”, please reference to “iAM Smart”’s API specification.

6. APPENDIX B

6.1. SUPPORTED VALUE AT SOURCE PARAMETER

CorpID System supports different types of browsers and e-service App calling methods. e-service should detect its user agent value for a browser use case and pass the respective source value to the CorpID System.

Platform	Supported Browser / e-service App calling method	Source
For e-service Web/App in Different Device and For e-service Web in Same Device		
Android Browser	Chrome	Android_Chrome
	Firefox	Android_Firefox
	Edge	Android_Edge
	Samsung built-in browser	Android_Samsung
	Huawei built-in browser	Android_Huawei
	Xiaomi built-in browser	Android_Xiaomi
iOS Browser	Safari	iOS_Safari
	Chrome	iOS_Chrome
	Firefox	iOS_Firefox
	Edge	iOS_Edge
Desktop Browser	Chrome, IE, Edge, Firefox, Safari	PC_Browser

For e-service App in Same Device With CorpID App API v1
--

CORPID SIMULATED API SPECIFICATION

e-service Mobile App	To be invoked by Universal Link (iOS)	App_Link
	To be invoked by App_Package (Android)	App_Package

7. APPENDIX C

7.1. IDENTIFY DOCUMENT TYPE

ID	Name in Form	
ID	Hong Kong Identity Card	香港身份證
PSP	Passport	護照
OID	Other identity type	其他身份證明文件

8. APPENDIX D

8.1. RETURN CODE

Response Code	Message/ Description
M00000	SUCCESS
Common Error Code	
M10001	The application is suspended. System should redirect to the “Service suspended” page.
M99990	Unhandled exception. System should redirect to the “Unhandled exception” page.
M99991	Cannot send the email notification. Please try again.
M99992	Authorization failed. Applicant is not permitted to submit the request.
S00001	Database error
C00001	Connection refused by backend API server (core)
M20000	unknown exception
M20001	parameter {%s} is missing
M20002	empty parameter {%s}
M20003	invalid parameter {%s}
M20004	duplicated request
M20005	unsupported signature method
M20006	signature verification failed
M20007	unsupported source
M20008	invalid e-service URL The callback URL provided by e-service is not registered in the ESP portal
M20009	accessToken not exist or expired

M20010	openID not exist
M20011	duplicated businessID
M20012	insufficient permission for e-service / Forbidden e-service shall check the scope or API access control in the the ESP portal
M20015	scope mismatch e-service shall verify the scopes approved for the client id in ESP and the scopes configured in e-service catalogue.
Encryption/Decryption Error Code	
M30001	key encryption key not exist or expired
M30002	content encryption key not exist or expired
M30003	encryption exception
M30004	decryption exception
Authentication Error Code	
M40000	user cancelled authentication request
M40001	user rejected authentication request
M40002	failed to authenticate
M40003	authentication request timeout
M40004	authCode not exist or expired
Profile Request Error Code	
M50001	user rejected profile request
M50002	failed to request profile
M50003	profile request timeout
Formfilling Request Error Code	
M60000	user cancelled Form Filling request
M60001	user rejected Form Filling request
M60002	failed to request Form Filling
M60003	Form Filling request timeout
Signing Request Error Code	
M70000	user cancelled signing request
M70001	user rejected signing request
M70002	failed to request signing
M70003	signing request timeout

M70004	user not allowed to sign
M70005	inconsistent HKIC number
M70006	failed to process signing acknowledgement
Bulk Digital Signing Request Error Code	
M71000	user cancelled bulk signing request
M71001	user rejected bulk signing request
M71002	failed to request signing
M71003	signing request timeout
M71004	user is not allowed to sign
M71005	inconsistent HKIC number
M71006	failed to process signing acknowledgement
M71007	the request to be cancelled is not Pending for Signing
M71008	BSQC Token not found
M71009	inconsistent department or service name
Re-authentication Error Code	
M80001	user rejected re-authentication request
M80002	failed to request re-authentication
M80003	re-authentication request timeout

8.2. HTTP STATUS CODE

Status Code	Status description
200 OK	Request successful
302 Found	The requested resource resides temporarily under a different URI.
400 Bad Request	The request could not be understood by the server due to malformed syntax.
401 Unauthorised	The request has not been applied because it lacks valid authentication credentials for the target resource.
403 Forbidden	The server understood the request but is refusing to fulfil it.
404 Not Found	The server has not found anything matching the

	Request-URI.
413 Payload Too Large	The uploaded files are larger than 5MB or the request entity is larger than 10MB.
429 Too Many Requests	The user has sent too many requests in a given amount of time.
500 Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.
503 Service Unavailable	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

End of Document