1 DID

The creation of DID is done offline, so the following API is used to upload the DID to the chain and query the DID Document information on the chain.

1.1 Verify DID Document

Interface Address	/did/verifyDoc			
Description	Verify the content format and signature value of the offline generated DID Document.			
Interface request parameter				
No.	Parameter	Type	Required	Description
1		VerifyDocumentReq	Y	Wrapper class
VerifyDocumentReq				
No.	Parameter	Type	Required	Description
1	didDoc	DidDocument	Y	DID Document
DidDocument				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	version	String	Y	Version
3	created	String	Y	Created date
4	updated	String	Y	Updated date
5	authentication	PublicKey	Y	Primary public key
6	recovery	PublicKey	Y	Recovery public key
7	proof	Proof	Y	Signature
PublicKey				
No.	Parameter	Type	Required	Description
1	publicKey	String	Y	Public key
2	type	String	Y	Algorithm type
Proof				
No.	Parameter	Type	Required	Description
1	creator	String	Y	DIDs involved in the calculation of signature values
2	type	String	Y	Algorithm type
3	signatureValue	String	Y	Signature value
Interface response parameter				
No.	Parameter	Type	Description	
1		Boolean	Return true if success, return false if failed	

1.2 Add DID Document to the chain

Interface Address	/did/putDoc
Description	The DID Document is stored in the chain. The verification will be executed internally, so if you want to upload the DID Document to the chain, you can directly call this interface.

Interface request parameter				
No.	Parameter	Type	Required	Description
1		DidDocSotreReq	Y	Wrapper class
DidDocSotreReq				
1	didDoc	Document	Y	DID Document
Document				
1	did	String	Y	DID
2	version	String	Y	Version
3	created	String	Y	Created date
4	updated	String	Y	Updated date
5	authentication	PublicKey	Y	Primary public key
6	recovery	PublicKey	Y	Recovery public key
7	proof	Proof	Y	Signature
PublicKey				
1	publicKey	String	Y	Public key
2	type	String	Y	Algorithm type
Proof				
1	creator	String	Y	DID involved in the calculation of the Signature value
2	type	String	Y	Algorithm type
3	signatureValue	String	Y	Signature value

Interface response parameter			
No.	Parameter	Type	Description
1		Boolean	Return true if success, return false if failed

1.3 Get DID Document

Interface Address	/did/getDoc
Description	The information in the DID Document is a record and description of the DID, so anyone can query the corresponding DID Document on the chain by the DID. It can be used to verify the DID and obtain the DID public key.

Interface request parameter				
No.	Parameter	Type	Required	Description
1		DidDocumentReq	Y	Wrapper class
DidDocumentReq				
No.	Parameter	Type	Required	Description

1	did	String	Y	DID
Interface response parameter				
No.	Parameter	Type	Description	
1		DidDocument	DID Document	
DidDocument				
No.	Parameter	Type	Description	
1	did	String	DID	
2	version	String	Version	
3	created	String	Created date	
4	updated	String	Updated date	
5	authentication	PublicKey	Primary public key	
6	recovery	PublicKey	Recovery public key	
7	proof	Proof	Signature	
PublicKey				
No.	Parameter	Type	Description	
1	publicKey	String	Public key	
2	type	String	Algorithm type	
Proof				
No.	Parameter	Type	Description	
1	creator	String	DID involved in the calculation of the Signature value	
2	type	String	Algorithm type	
3	signatureValue	String	Signature value	

1.4 Verify DID Signature

Interface Address	/did/verifyDidSign			
Description	Verify the signature value of the DID to ensure the authenticity and validity of the current DID.			
Interface request parameter				
No.	Parameter	Type	Required	Description
1		VerifyDidReq	Y	Wrapper class
VerifyDidReq				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	didSign	String	Y	DID signature value
Interface response parameter				
No.	Parameter	Type	Description	
1		Boolean	Return true if success, return false if failed	

1.5 Update Key

Interface Address	/did/resetDidAuth
Description	The generation of the new authentication public-private key pair from the recovery public-private key information is done by the DID SDK. The interface receives new DID Document content from the user for on-chain update.

Interface request parameter

No.	Parameter	Type	Required	Description
1		RestDocAuth	Y	Wrapper class

RestDocAuth

No.	Parameter	Type	Required	Description
1	didDoc	Document	Y	DID Document
2	authPubKeySign	String	Y	The recovery private key performs k1 signature on the recovery public key

Document

No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	version	String	Y	Version
3	created	String	Y	Created date
4	updated	String	Y	Updated date
5	authentication	PublicKey	N	Primary public key
6	recovery	PublicKey	Y	Recovery public key
7	proof	Proof	Y	Signature

PublicKey

No.	Parameter	Type	Required	Description
1	publicKey	String	Y	Public key
2	type	String	Y	Algorithm type

Proof

No.	Parameter	Type	Required	Description
1	creator	String	Y	DID involved in the calculation of the Signature value
2	type	String	Y	Algorithm type
3	signatureValue	String	Y	Signature value

Interface response parameter

No.	Parameter	Type	Description
1		KeyInfo	New authentication public key information

PublicKey

No.	Parameter	Type	Description
1	publicKey	String	Public key
2	type	String	Algorithm type

2 Issuer

The issuer and user are two roles, and the following APIs are the pre-constraints for issuing credentials. The process of changing the DID user to the issuer does not change the DID Identifier or DID Document, but only the status.

To issue a credential, a DID user needs to register as an issuer and then define a template for registering the credential they want to issue. The credential template will be stored on the chain and everyone can query.

2.1 Issuer Registration

Interface Address	/did/registerAuthIssuer
Description	The DID user becomes the issuer and the issuer information is uploaded if the registration is successful.

Interface request parameter

No.	Parameter	Type	Required	Description
1		RegisterAuthorityIssuerWrapper	Y	Wrapper class

RegisterAuthorityIssuerWrapper

No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	name	String	Y	Issuer's name
3	publicKeySign	String	Y	Public key

Interface response parameter

No.	Parameter	Type	Description
1		Boolean	Return true if success, return false if failed

2.2 Query Issuer

Interface Address	/did/queryAuthIssuerList
Description	Check whether you are the issuer by DID and identify which type of credentials can be issued by name.

Interface request parameter

No.	Parameter	Type	Required	Description
1		AuthIssuerListWrapper	Y	Wrapper class

AuthIssuerListWrapper

No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	page	Integer	Y	Page
3	size	Integer	Y	Number of entries

Interface response parameter

No.	Parameter	Type	Description
1		AuthorityIssuer	Issuer Information
AuthorityIssuer			
No.	Parameter	Type	Description
1	did	String	DID
2	name	String	Issuer's name

2.3 Register Credential Template

Interface Address	/did/registerCpt
Description	The issuer customizes the credential template and can agree on which attribute values must be provided by the applicant. For example, in the template of college diploma, you can agree that "name" and "student number" are mandatory information.

Interface request parameter				
No.	Parameter	Type	Required	Description
1		RegisterCptWrapper	Y	Wrapper class
RegisterCptWrapper				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	cptJsonSchema	Map<String, JsonSchema>	Y	JsonSchema information for MapType
3	title	String	Y	Credential template title
4	description	String	Y	Credential template description
5	type	String	Y	Credential type, fill in proof
6	proof	Proof	Y	Signature
7	create	String	Y	Created date
8	update	String	Y	Updated date
Proof				
No.	Parameter	Type	Required	Description
1	creator	String	Y	DID involved in the calculation of the Signature value
2	type	String	Y	Algorithm type
3	signatureValue	String	Y	Signature value
JsonSchema				
No.	Parameter	Type	Required	Description
1	type	String	Y	Field type
2	description	String	Y	Field description
3	required	boolean	Y	Whether is required to fill

Interface response parameter			
No.	Parameter	Type	Description
1		CptBaseInfo	Credential template information

CptBaseInfo			
No.	Parameter	Type	Description
1	cptId	Long	Credential template ID
2	cptVersion	Integer	Version

2.4 Query Credential Template List

Interface Address	/did/queryCptList
Description	Anyone can check all their credential templates by DID. It is possible for the same individual/organization to register multiple credential templates. For example, a university may have a degree template, an incomplete template, etc. in addition to a diploma template.

Interface request parameter				
No.	Parameter	Type	Required	Description
1		QueryCptListWrapper	Y	Wrapper class

QueryCptListWrapper				
No.	Parameter	Type	Required	Description
1	page	Integer	Y	Page
2	size	Integer	Y	Number of entries
3	did	String	Y	DID

Interface response parameter			
No.	Parameter	Type	Description
1		Pages<CptInfo>	Credential template list info

Pages<CptInfo>			
No.	Parameter	Type	Description
1	page	Integer	Page
2	size	Integer	Number of entries
3	totalNum	Integer	Number of entries in total
4	totalPage	Integer	Number of pages in total
5	result	List<CptInfo>	Result list

CptInfo			
No.	Parameter	Type	Description
1	cptVersion	Integer	Credential template version
2	cptJsonSchema	Map<String, JsonSchema>	JsonSchema information for MapType
3	title	String	Credential template title
4	description	String	Credential template description
5	publisherDid	String	DID of the credential template issuer
6	proof	Proof	Signature
7	cptId	Long	Credential template ID
8	create	String	Created date
9	update	String	Updated date

JsonSchema			
1	type	String	Field type
2	description	String	Field description
3	required	Boolean	whether is required to fill
Proof			
1	creator	String	DID involved in the calculation of the Signature value
2	type	String	Algorithm type
3	signatureValue	String	Signature value

2.5 Query Credential Template

Interface Address	/did/queryCptById			
Description	Query the contents of a specific credential template by its ID.			
Interface request parameter				
No.	Parameter	Type	Required	Description
1		QueryCptByIdWrapper	Y	Wrapper class
QueryCptByIdWrapper				
No.	Parameter	Type	Required	Description
1	cptId	Long	Y	Credential template ID
Interface response parameter				
No.	Parameter	Type	Description	
1		CptInfo	Credential template information	
CptInfo				
No.	Parameter	Type	Description	
1	cptVersion	Integer	Version	
2	cptJsonSchema	Map<String, JsonSchema>	JsonSchema information for MapType	
3	title	String	Credential template title	
4	description	String	Credential template description	
5	publisherDid	String	DID of the credential template issuer	
6	proof	Proof	Signature	
7	cptId	Long	Credential template ID	
8	create	String	Created date	
9	update	String	Updated date	
JsonSchema				
No.	Parameter	Type	Description	
1	type	String	Field type	

2	description	String	Field description
3	required	boolean	WHETHER IS REQUIRED TO FILL
Proof			
No.	Parameter	Type	Description
1	creator	String	DID involved in the calculation of the Signature value
2	type	String	Algorithm type
3	signatureValue	String	Signature value

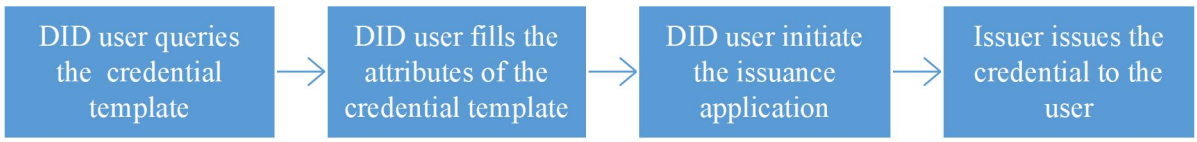
2.6 Update Credential Template

Interface Address	/did/updateCpt			
Description	The issuer updates the content of its own registered credential templates. The update of credential template does not affect credentials already issued.			
Interface request parameter				
No.	Parameter	Type	Required	Description
1		RegisterCptWrapper	Y	Wrapper class
RegisterCptWrapper				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	cptJsonSchema	Map<String, JsonSchema>	Y	JsonSchema information for MapType
3	title	String	Y	Credential template title
4	description	String	Y	Credential template description
5	type	String	Y	Credential type, fill in proof
6	proof	Proof	Y	Signature
7	cptId	Long	Y	Credential template ID
8	create	String	Y	Created date
9	update	String	Y	Updated date
Proof				
No.	Parameter	Type	Required	Description
1	creator	String	Y	DID involved in the calculation of the Signature value
2	type	String	Y	Algorithm type
3	signatureValue	String	Y	Signature value
JsonSchema				
No.	Parameter	Type	Required	Description
1	type	String	Y	Field type
2	description	String	Y	Field description
3	required	boolean	Y	WHETHER IS REQUIRED TO FILL
Interface response parameter				
No.	Parameter	Type		Description

1		CptBaseInfo	Credential template information
CptBaseInfo			
No.	Parameter	Type	Description
1	cptId	Long	Credential template ID
2	cptVersion	Integer	Version, plus one after each update is successful

3 Credential

The credential is generated based on the credential template. The application of the credential is made by the user, and then the issuer issues the credential. The credential issuance process is generally as follows:



Once the user has the credentials issued by the issuer, he/she can present them to the verifier for further use.

3.1 Issue Credential

Interface Address	/did/createCredential
Description	The attribute values defined in the credential template are provided by the issuer for the DID user to obtain on the front page. The issuer issues the credentials for the DID user through this interface. If there are more Claim parameters than defined in the credential template, the server side will discard them.

Interface request parameter				
No.	Parameter	Type	Required	Description
1		CreateCredentialReq	Y	Wrapper class
CreateCredentialReq				
No.	Parameter	Type	Required	Description
1	cptId	Long	Y	Credential template ID
2	issuerDid	String	Y	DID of the credential template issuer
3	userDid	String	Y	DID of the user requesting the credentials
4	expirationDate	String	Y	Credential expiration date
5	claim	Map<String,Object>	Y	Claim data
6	type	String	Y	Credential type, fill in Proof
7	shortDesc	String	N	Brief description of the credential template. If this field is null, the value of the title field in the credential template is displayed. If

				not, the input value is displayed.
8	longDesc	String	N	Detailed description of the credential template
Interface response parameter				
No.	Parameter	Type		Description
1		CredentialWrapper		Credential issuance information
CredentialWrapper				
No.	Parameter	Type		Description
1	context	String		Specification
2	id	String		Credential ID
3	type	String		Credential type, fill in proof
4	cptId	Long		Credential template ID
5	issuerDid	String		DID of credential issuer
6	userDid	String		DID of the user requesting the credentials
7	expirationDate	String		Expiration date
8	created	String		Created date
9	shortDesc	String		Brief description of the credential
10	longDesc	String		Detailed description of the credential
11	claim	Map<String, Object>		Claim data
12	proof	Map<String, Object>		Signature

3.2 Verify Credential

Interface Address	/did/verifyCredential			
Description	Generally called by the verifier. It can verify whether a particular credential is valid or not. Verify the signature of the credential, whether the credential is expired, and whether the credential is revoked, respectively.			
Interface request parameter				
No.	Parameter	Type	Required	Description
1		VerifyCredentialReq	Y	Wrapper class
VerifyCredentialReq				
No.	Parameter	Type	Required	Description
1	credentialWrapper	CredentialWrapper	Y	Credential information
2	publicKey	PublicKey	Y	Issuer's public key
CredentialWrapper				
No.	Parameter	Type	Required	Description
1	context	String	Y	Specification
2	id	String	Y	Credential ID
3	type	String	Y	Credential type, fill in Proof
4	cptId	Long	Y	Credential template ID

5	issuerDid	String	Y	DID of credential issuer
6	userDid	String	Y	DID of the user requesting the credentials
7	expirationDate	String	Y	Expiration date
8	created	String	Y	Created date
9	shortDesc	String	N	Brief description of the credential
10	longDesc	String	N	Detailed description of the credential
11	claim	Map<String, Object>	Y	Claim data
12	proof	Map<String, Object>	Y	Signature

PublicKey

No.	Parameter	Type	Required	Description
1	publicKey	String	Y	Public key
2	type	String	Y	Algorithm type

Interface response parameter

No.	Parameter	Type	Description
1		Boolean	Return true if success, return false if failed

3.3 Revoke Credential

Interface Address	/did/revokeCredential
Description	Called by the issuer to revoke or void a credential that has been issued. Since the issued credential is already in the custody of the user, the revocation of the credential is followed by the upload of its credential ID.

Interface request parameter

No.	Parameter	Type	Required	Description
1		RevokCredentialReq	Y	Wrapper class

RevokCredentialReq

No.	Parameter	Type	Required	Description
1	credId	String	Y	Credential ID to be revoked
2	cptId	Long	Y	Credential template ID
3	did	String	Y	Issuer's DID
4	revokeDate	String	Y	Revoked date
5	publicKeySign	String	Y	The primary private key performs a k1 signature on the primary public key
6	revokeSign	String	Y	After splicing the certificate ID and revocation time, use the primary private key for k1 revocation signature

Interface response parameter

No.	Parameter	Type	Description

1		Boolean	Return true if success, return false if failed

3.4 Query Revoked Credential

Interface Address	/did/getRevokedCredList			
Description	Called when verifying credentials. Find out all its revoked credential IDs by Issuer's DID.			
Interface request parameter				
No.	Parameter	Type	Required	Description
1		QueryCredentialWrapper	Y	Wrapper class
QueryCredentialWrapper				
No.	Parameter	Type	Required	Description
1	Page	Integer	Y	Page
2	Size	Integer	Y	Number of entries
3	did	String	Y	Issuer's DID
Interface response parameter				
No.	Parameter	Type	Description	
1		Pages<BaseCredential>	Revocation list information	
Pages<BaseCredential>				
No.	Parameter	Type	Description	
1	page	Integer	Page	
2	size	Integer	Number of entries	
3	totalNum	Integer	Number of entries in total	
4	totalPage	Integer	Number of pages in total	
5	result	List<BaseCredential >	Result list	
BaseCredential				
No.	Parameter	Type	Description	
1	id	String	Credential ID	
2	created	String	Revoked date	

4 Response Code

Response code	Description
0	Success
1001	{attribute} is null
1002	The format of {attribute} is invalid
1003	{attribute} contains a null attribute value
1004	{attribute} is too long
1005	Transaction timeout
1006	Transaction error
1008	Config file does not exist

1009	Node private key is empty
1010	DID contract address is empty
1011	CPT contract address is empty
1012	Auth issuer contract address is empty
1013	DID blockchain type is empty
1014	Failed to initialize the DID SDK
1020	Failed to create the key pair
1021	Public and private keys do not match
1022	Public key is empty
1023	Invalid public key format
1024	Private key is empty
1025	Invalid private key format
1027	Encryption Type is empty
1028	Invalid Encryption Type
1029	Failed to sign the data
1030	Signer and DID do not match
1031	Signature verification failed
1032	grantPublicKey with public key in database do not match
1033	Public key and document's recovery public key do not match
1040	DID already exists
1041	DID does not exist
1042	Failed to create DID
1043	Invalid DID
1044	Failed to generate the DID
1045	Failed to generate the DID Document
1046	DID Document verification success
1050	DID is registered as the issuer
1051	DID is not registered as the issuer
1052	Failed to register as the issuer
1054	Issuer does not exist
1060	CPT does not exist
1062	privateKey does not match issuerDid
1063	Issuer and publisherDid in the CPT do not match
1070	The credential has been revoked
1071	The credential has expired
1072	Failed to revoke the credential
1073	CPT and credential do not match
1074	Failed to create credential
1075	Credential verification success

1076	The credential is not in the revoke list
1077	Computed DID from the document is not the same with the DID in the document
1078	Created time is different with updated time in DID Document
1079	Public key signature verification failed
1080	DID is not the same with the proof creator in CPT
1081	The DID Document version does not match the one on-chain
1082	The DID Document created time does not match the one on-chain
1083	The DID Document recovery key does not match the one on-chain
1084	Failed to add DID Document to the chain
1085	Failed to create the key pair
1086	Failed to calculate the DID
1087	Failed to calculate the DID Document signature
1088	Failed to create the DID Document
1090	The mnemonic is empty
1337	Failed to encrypt the key
1338	Failed to sign the data
2000	credId does not match cptId
9999	Unknown exception

Note: {attribute} is a dynamic parameter.