1 DID

1.1 Generate Private and Public Keys by Mnemonics

The user can customize mnemonics and call this function to generate a pair of public-private keys for the k1 algorithm offline. As long as mnemonics are the same, the generated public and private keys must be the same for each call.

Function name	createKeyPair(List<String> mnemList)			
Description	The user can generate the private and public keys by mnimonics			
Request Parameters				
No.	Parameter	Type	Required	Description
1	mnemList	List<String>	Y	Mnemonics
Response Parameters				
No.	Parameter	Type	Required	Description
1		DidDataWrapper	Y	Private key
KeyPair				
No.	Parameter	Type	Required	Description
1	privateKey	String	Y	Private key
2	publicKey	String	Y	Public key
3	type	String	Y	Algorithm Type

1.2 Create DID

Function name	createDid(Boolean isStorageOnChain)			
Description	Call this function to create a DID. isStorageOnChain indicates whether the DID Document is stored on-chain or not.			
Request Parameters				
No.	Parameter	Type	Required	Description
1	isStorageOnChain	Boolean	Y	On-chain marker. true means DID Document is stored on-chain; false means DID Document is not stored on-chain.
Response Parameters				
No.	Parameter	Type	Required	Description
1		DidDataWrapper	Y	
DidDataWrapper				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	authPublicKey	KeyPair	Y	Primary public/private key information
3	recyPublicKey	KeyPair	Y	Recovery public/private key information

4	document	DocumentInfo	N	DID Document
5	didSign	String	Y	DID signature
6	address	String	Y	Account address
No.	Parameter	Type	Required	Description
DocumentInfo				
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
KeyPair				
No.	Parameter	Type	Required	Description
1	privateKey	String	Y	Private key
2	publicKey	String	Y	Public key
3	type	String	Y	Algorithm type
PublicKey				
No.	Parameter	Type	Required	Description
1	type	String	Y	Algorithm type
2	publicKey	String	Y	Public key
Proof				
No.	Parameter	Type	Required	Description
1	type	String	Y	Algorithm type
2	creator	String	Y	DID
3	signatureValue	String	Y	Signature value

1.3 Verify DID Document

Function name	verifyDidDocument(DidDocument didDocument)			
Description	Verify the content format and signature value of the offline generated DID Document.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		DidDocument	Y	
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	type	String	Y	Algorithm type
2	publicKey	String	Y	Public key

Proof

No.	Parameter	Type	Required	Description
1	type	String	Y	Algorithm type
2	creator	String	Y	DID
3	signatureValue	String	Y	Signature value

Response Parameters

No.	Parameter	Type	Required	Description
1		Boolean	Y	Return true if success, return false if failure

1.4 Upload DID Document

Function name	storeDidDocumentOnChain(DidDocument didDocument)
Description	Store the DID document on-chain. Firstly to execute the verification, so that you can call this function if you want to store the DID Document on chain.

Request Parameters

No.	Parameter	Type	Required	Description
1		DidDocument	Y	

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	type	String	Y	Algorithm type
2	publicKey	String	Y	Public key

Proof

No.	Parameter	Type	Required	Description

1	type	String	Y	Algorithm type
2	creator	String	Y	DID
3	signatureValue	String	Y	Signature value

Response Parameters

No.	Parameter	Type	Required	Description
1		Boolean	Y	Storage result

1.5 Get DID Document

Function name	getDidDocument(String did)
Description	The information in the DID Document is a record and description of the DID, and anyone can query the corresponding DID Document from the chain by the DID. It can be used to verify the DID and obtain the DID public key.

Request Parameters

No.	Parameter	Type	Required	Description
1	did	String	Y	DID

Response Parameters

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

Proof

No.	Parameter	Type	Required	Description
1	type	String	Y	Algorithm type
2	creator	String	Y	DID
3	signatureValue	String	Y	Signature value

1.6 Verify DID

Function name	verifyDIdSign(String did, String didSign)

Description	Verify the digital signature value of the DID, so that it can ensure the authenticity and validity of the current DID.			
Request Parameters				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	didSign	String	Y	DID signature
Response Parameters				
No.	Parameter	Type	Required	Description
1		Boolean	Y	Return true if success, return false if failure

1.7 Key Update

Function name	resetDidAuth(ResetDidAuth restDidAuth)			
Description	If the primary private key is lost or leaked, a pair of primary public and private keys can be regenerated by the recovery private key. The user completes the primary public-private keys update with the recovery public-private keys. After the key is updated, the user's DID Document will also be updated, but the DID remains the same. If the user fills in the primary public-private keys, the primary public keys in the DID Document is updated and the signature is recalculated using the filled-in primary public key; otherwise, a new pair of primary public private keys is automatically generated and the primary public key and signature calculation of the DID Document are updated. Note: If the issuer updates the key, all the previously issued credentials will not pass the signature verification (*if the issuer records the master public key of the credential in the business system, it can transmit the old master public key information to the user, then it can also pass the credential verification*).			
Request Parameters				
No.	Parameter	Type	Required	Description
1		ResetDidAuth	Y	
ResetDidAuth				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	primaryKeyPair	KeyPair	N	Primary public and private key
3	recoveryKey	KeyPair	Y	Recovery public and private key
KeyPair				
1	privateKey	String	Y	Private Key
2	publicKey	String	Y	Public Key
3	type	String	Y	Algorithm type
Response Parameters				
No.	Parameter	Type	Required	Description
1		KeyPair	Y	New public and private key pair
KeyPair				
No.	Parameter	Type	Required	Description
1	privateKey	String	Y	Private key

2	publicKey	String	Y	Public key
3	type	String	Y	Algorithm type

2 Issuer

2.1 Register Issuer

Function name	registerAuthIssuer(RegisterAuthorityIssuer register)			
Description	The DID user becomes the issuer, and the issuer information is uploaded to the chain if the registration is successful.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		RegisterAuthorityIssuer	Y	
RegisterAuthorityIssuer				
No.	Parameter	Type	Required	Description
1	privateKey	String	Y	Private key
2	did	String	Y	DID
3	name	String	Y	Issuer's name
Response Parameters				
No.	Parameter	Type	Required	Description
1		Boolean	Y	Return true if success, return false if failure

2.2 Query Issuer

Function name	queryAuthIssuerList(AuthIssuerList query)			
Description	You can query whether it is the issuer through DID and identify the type of credential that can be issued by name.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		AuthIssuerList	Y	
AuthIssuerList				
No.	Parameter	Type	Required	Description
1	page	Integer	Y	Number of pages
2	size	Integer	Y	Number of entries per page
3	did	String	Y	DID
Response Parameters				
No.	Parameter	Type	Required	Description
1		Pages<AuthorityIssuer>	Y	Query result, the list of issuers
Pages				
No.	Parameter	Type	Required	Description
1	page	Integer	Y	Page number

2	size	Integer	Y	Paging Size
3	totalNum	Integer	Y	Total number
4	totalPage	Integer	Y	Total pages
5	result	List< AuthorityIssuer>	Y	List of issuers
AuthorityIssuer				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	name	String	Y	Issuer’s name

2.3 Register credential template

Function name	registerCpt(RegisterCpt 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.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		RegisterCpt	Y	
RegisterCpt				
No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	privateKey	String	Y	Private key
3	cptJsonSchema	Map<String, JsonSchema>	Y	JsonSchema of credential template
4	title	String	Y	Title
5	description	String	Y	Description
6	type	String	Y	Credential Type, fill in Proof
7	cptId	Long	Y	Credential template ID
JsonSchema				
No.	Parameter	Type	Required	Description
1	type	String	Y	Field type
2	description	String	Y	Field description
3	required	Boolean	Y	true: required; false: optional
Response Parameters				
No.	Parameter	Type	Required	Description
1		CptBaseInfo	Y	Registration result, basic information of credential template
CptBaseInfo				
No.	Parameter	Type	Required	Description
1	cptId	Long	Y	Credential template ID
2	cptVersion	Integer	Y	Credential template Version

2.4 Query Credential Template List

Function name	queryCptListByDid(QueryCptList query)			
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.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		QueryCpt	Y	
QueryCpt				
No.	Parameter	Type	Required	Description
1	page	Integer	Y	Number of pages
2	size	Integer	Y	Number of entries per page
3	did	String	Y	DID
Response Parameters				
No.	Parameter	Type	Required	Description
1		Pages<CptInfo>	Y	Query result, credential template information list
Pages				
No.	Parameter	Type	Required	Description
1	page	Integer	Y	Page number
2	size	Integer	Y	Paging Size
3	totalNum	Integer	Y	Total number
4	totalPage	Integer	Y	Total pages
5	result	List<CptInfo>	Y	List of credential templates
CptInfo				
No.	Parameter	Type	Required	Description
1	cptJsonSchema	Map<String, JsonSchema>	Y	JsonSchema for Credential template
2	title	String	Y	Title
3	description	String	Y	Description
4	publisherDid	String	Y	DID to create credential template
5	proof	Proof	Y	Signature
6	create	String	Y	Created date
7	**update**	**String**	**Y**	Updated date
8	**cptId**	**Long**	**Y**	Credential template ID
9	**cptVersion**	**Integer**	**Y**	Credential template version
Proof				
No.	Parameter	Type	Required	Description

1	type	String	Y	Algorithm type
2	creator	String	Y	DID
3	signatureValue	String	Y	Signature value
JsonSchema				
No.	Parameter	Type	Required	Description
1	type	String	Y	Type
2	description	String	Y	Description
3	required	Boolean	Y	True: required; false: optional

2.5 Query Credential Template

Function name	queryCptById(Long cptId)			
Description	Query the contents of a specific credential template by its ID.			
Request Parameters				
No.	Parameter	Type	Required	Description
1	cptId	Long	Y	Credential template ID
Response Parameters				
No.	Parameter	Type	Required	Description
1		CptInfo	Y	Query result, credential template information
CptInfo				
No.	Parameter	Type	Required	Description
1	cptJsonSchema	Map<String, JsonSchema>	Y	JsonSchema for Credential template
2	title	String	Y	Title
3	description	String	Y	Description
4	publisherDid	String	Y	DID to create the credential template
5	proof	Proof	Y	Signature
6	create	String	Y	Created date
7	update	String	Y	Updated date
8	cptId	Long	Y	Credential template ID
9	cptVersion	Integer	Y	Credential template version
JsonSchema				
No.	Parameter	Type	Required	Description
1	type	String	Y	Type
2	description	String	Y	Description
3	required	Boolean	Y	True: required ; false: optional
Proof				

No.	Parameter	Type	Required	Description
1	type	String	Y	Algorithm type
2	creator	String	Y	DID
3	signatureValue	String	Y	Signature value

2.6 Update Credential Template

Function name	updateCpt(RegisterCpt registerCpt)
Description	The issuer updates the content of its own registered credential templates. The update of the credential template ID does not affect issued credentials.

Request Parameters

No.	Parameter	Type	Required	Description
1		RegisterCpt	Y	

RegisterCpt

No.	Parameter	Type	Required	Description
1	did	String	Y	DID
2	privateKey	String	Y	Private key
3	cptJsonSchema	Map<String, JsonSchema>	Y	JsonSchema for Credential template
4	title	String	Y	Credential template title
5	description	String	Y	Credential template description
6	type	String	Y	Credential Type, fill in proof
7	cptId	Long	Y	Credential template ID

JsonSchema

No.	Parameter	Type	Required	Description
1	type	String	Y	Type
2	description	String	Y	Description
3	required	Boolean	Y	true：required；false：optional

Response Parameters

No.	Parameter	Type	Required	Description
1		CptBaseInfo	Y	Update result, basic information of credential template

CptBaseInfo

No.	Parameter	Type	Required	Description
1	cptId	Long	Y	Credential template ID
2	cptVersion	Integer	Y	Credential template version, add 1 after each successful update

3 Credential

3.1 Create Credential

Function name	createCredential(CreateCredential 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.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		CreateCredential	Y	
CreateCredential				
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 who created the credentials
4	expirationDate	String	Y	Credential expiration date. Should be greater than today. In the form of yyyy-mm-dd
5	claim	Map<String, Object>	Y	Content of the credential. The claim data needs to correspond to the format of the credential template
6	type	String	Y	Credential type, input Proof
7	privateKey	String	Y	Private key
8	shortDesc	String	N	Brief description of the credential. The default value is the credential template title.
9	longDesc	String	N	Detailed description of the credential
Response Parameters				
No.	Parameter	Type	Required	Description
1		CredentialWrapper	Y	Creation result, Credential information
CredentialWrapper				
No.	Parameter	Type	Required	Description
1	context	String	Y	Version
2	id	String	Y	Credential ID
3	type	String	Y	Credential type, Proof

4	cptId	Long	Y	Credential template Id
5	issuerDid	String	Y	DID of the credential template issuer
6	userDid	String	Y	DID of the user who created the credentials
7	expirationDate	String	Y	Credential expiration date
8	created	String	Y	Created date
9	shortDesc	String	Y	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

3.2 Verify Credential

Function name	verifyCredential(CredentialWrapper createCredential,PublicKey publicKey)			
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.			
Request Parameters				
No.	Parameter	Type	Required	Description
1	createCredential	CredentialWrapper	Y	
2	publicKey	PublicKey	Y	Public key
CredentialWrapper				
No.	Parameter	Type	Required	Description
1	context	String	Y	Version
2	id	String	Y	Credential ID
3	type	String	Y	Credential type, Proof
4	cptId	Long	Y	Credential template ID
5	issuerDid	String	Y	DID of the credential template issuer
6	userDid	String	Y	DID of the user who created the credentials
7	expirationDate	String	Y	Credential 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	type	String	Y	Algorithm type
2	publicKey	String	Y	Public key
Response Parameters				
No.	Parameter	Type	Required	Description
1		Boolean	Y	Return true if success, return false if failure

3.3 Revoke Credential

Function name	revokeCredential(RevokeCredential cred)			
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.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		RevokeCredential	Y	
RevokeCredential				
No.	Parameter	Type	Required	Description
1	credId	String	Y	Credential ID
2	cptId	Long	Y	Credential template Id
3	did	String	Y	DID
4	privateKey	String	Y	Private key
Response Parameters				
No.	Parameter	Type	Required	Description
1		Boolean	Y	Return true if success, return false if failure

3.4 Query Revoked Credential

Function name	getRevokedCredList(QueryCredentialList queryCredentialList)			
Description	Called when verifying credentials. Find out all its revoked credential IDs by Issuer's DID.			
Request Parameters				
No.	Parameter	Type	Required	Description
1		QueryCredential	Y	
QueryCredential				
No.	Parameter	Type	Required	Description
1	page	Integer	Y	Number of pages
2	size	Integer	Y	Number of entries per page

3	did	String	Y	DID
Response Parameters				
No.	Parameter	Type	Required	Description
1		Pages<BaseCredential>	Y	Query result, basic info list of the credential
Pages				
No.	Parameter	Type	Required	Description
1	page	Integer	Y	Page number
2	size	Integer	Y	Paging Size
3	totalNum	Integer	Y	Total number
4	totalPage	Integer	Y	Total pages
5	result	List<BaseCredential>	Y	List of revoked documents
BaseCredential				
No.	Parameter	Type	Required	Description
1	id	String	Y	Credential ID
2	created	String	Y	Revoked time