

Class: MetadataApi

Constructors

constructor

•new MetadataApi (): [MetadataApi](#)

Returns

[MetadataApi](#)

Properties

appDataHexToCid

•appDataHexToCid : (appDataHex :string) =>Promise <string
=appDataHexToCid

Type declaration

► (appDataHex):Promise <string

Parameters

Name Type appDataHex string

Returns

Promise <string

Defined in

api/index.ts:20

appDataHexToCidLegacy

•appDataHexToCidLegacy : (appDataHex :string) =>Promise <string
=appDataHexToCidLegacy

Type declaration

► (appDataHex):Promise <string

Parameters

Name Type appDataHex string

Returns

Promise <string

Defined in

api/index.ts:31

appDataToCid

•appDataToCid : (appData :[AnyAppDataDocVersion](#)) =>Promise <[dpfsHashInfo](#)
(fullAppData :string) =>Promise <[dpfsHashInfo](#) |void =appDataToCid

Type declaration

► (appData):Promise <[IpfsHashInfo](#)

Calculates appDataHex without publishing file to IPFS

This method is intended to quickly generate the appDataHex independent of IPFS upload/pinning

Parameters

Name	Type	Description
appData	AnyAppDataDocVersion	JSON document which will be stringified in a deterministic way to calculate the IPFS hash

Returns

Promise <[IpfsHashInfo](#)

► (fullAppData):Promise <[IpfsHashInfo](#) |void

Calculates appDataHex without publishing file to IPFS

This method is intended to quickly generate the appDataHex independent of IPFS upload/pinning

Parameters

Name	Type	Description
fullAppData	string	JSON string with the full appData document

Returns

Promise <[IpfsHashInfo](#) |void

Defined in

api/index.ts:19

appDataToCidLegacy

•appDataToCidLegacy : (appData :[AnyAppDataDocVersion](#)) =>Promise <[IpfsHashInfo](#) |void

(fullAppData :string) =>Promise <[IpfsHashInfo](#) |void =appDataToCidLegacy

Type declaration

► (appData):Promise <[IpfsHashInfo](#) |void

Calculates appDataHex without publishing file to IPFS

This method is intended to quickly generate the appDataHex independent of IPFS upload/pinning

Parameters

Name	Type	Description
appData	AnyAppDataDocVersion	JSON document which will be stringified in a deterministic way to calculate the IPFS hash

Returns

Promise <[IpfsHashInfo](#) |void

Deprecated

Old way of deriving th hash

► (fullAppData):Promise <[IpfsHashInfo](#) |void

Calculates appDataHex without publishing file to IPFS

This method is intended to quickly generate the appDataHex independent of IPFS upload/pinning

Parameters

Name	Type	Description
fullAppData	string	JSON string with the full appData document

Returns

Promise <[ipfsHashInfo](#) | void

Deprecated

Old way of deriving th hash

Defined in

api/index.ts:30

cidToAppDataHex

•cidToAppDataHex : (cid :string) =>Promise <string

=cidToAppDataHex

Type declaration

► (cid):Promise <string

Parameters

Name	Type	cid	string
------	------	-----	--------

Returns

Promise <string

Defined in

api/index.ts:21

fetchDocFromAppDataHex

•fetchDocFromAppDataHex : (appDataHex :string ,ipfsUri? :string) =>Promise <void [AnyAppDataDocVersion](#)

=fetchDocFromAppDataHex

Type declaration

► (appDataHex ,ipfsUri?):Promise <void [AnyAppDataDocVersion](#)

Parameters

Name	Type	Description
appDataHex	string	Derives the CID from the appData hex, and fetches and parses the document from IPFS
ipfsUri?	string	URL of the IPFS gateway to use for the fetch

Returns

Promise <void |[AnyAppDataDocVersion](#)

a parsed AppData document

Defined in

api/index.ts:25

fetchDocFromAppDataHexLegacy

•fetchDocFromAppDataHexLegacy : (appDataHex :string ,ipfsUri? :string) =>Promise <void [AnyAppDataDocVersion](#)

=fetchDocFromAppDataHexLegacy

Type declaration

► (appDataHex ,ipfsUri?):Promise <void [AnyAppDataDocVersion](#)

Fetches the document from IPFS using the appData hex

Parameters

Name Type appDataHex string ipfsUri? string

Returns

Promise <void [AnyAppDataDocVersion](#)

Deprecated

Defined in

api/index.ts:32

fetchDocFromCid

•fetchDocFromCid : (cid :string ,ipfsUri :string) =>Promise <[AnyAppDataDocVersion](#)
=fetchDocFromCid

Type declaration

► (cid ,ipfsUri?):Promise <[AnyAppDataDocVersion](#)

Parameters

Name Type Default value cid string undefined ipfsUri string DEFAULT_IPFS_READ_URI

Returns

Promise <[AnyAppDataDocVersion](#)

Defined in

api/index.ts:24

generateAppDataDoc

•generateAppDataDoc : (params? :Partial <Omit <[AppDataRootSchema](#) ,"version"
) =>Promise <[LatestAppDataDocVersion](#) =generateAppDataDoc

Type declaration

► (params?):Promise <[LatestAppDataDocVersion](#)

Creates an appData document using the latest specification of the format

Without params creates a default minimum appData doc Optionally creates metadata docs

Example of result: { "appCode": "CoW Swap", "environment": "local", "metadata": { "quote": { "slippageBips": "50", "version": "0.2.0" }, "orderClass": { "orderClass": "market", "version": "0.1.0" } }, "version": "0.5.0" }

Parameters

Name Type params? Partial <Omit <[AppDataRootSchema](#) ,"version"

Returns

Promise <[LatestAppDataDocVersion](#)

Defined in

api/index.ts:15

getAppDataSchema

•getAppDataSchema : (version :string) =>Promise <[AnyAppDataDocVersion](#)
=getAppDataSchema

Type declaration

► (version):Promise <[AnyAppDataDocVersion](#)

Wrapper around @cowprotocol/app-data getAppDataSchema

Returns the appData schema for given version, if any Throws CowError when version doesn't exist

Parameters

Name Type version string

Returns

Promise <[AnyAppDataDocVersion](#)

Defined in

api/index.ts:14

uploadMetadataDocToIpfsLegacy

•uploadMetadataDocToIpfsLegacy : (appDataDoc :[AnyAppDataDocVersion](#) ,ipfsConfig :[Ipfs](#)) =>Promise <IpfsUploadResult
|void
=uploadMetadataDocToIpfsLegacy

Type declaration

► (appDataDoc ,ipfsConfig):Promise <IpfsUploadResult |void

Uploads a appDocument to IPFS

Parameters

Name Type Description appDataDoc [AnyAppDataDocVersion](#) Document to upload ipfsConfig [Ipfs](#) keys to access the IPFS API

Returns

Promise <IpfsUploadResult |void

the IPFS CID v0 of the content

Deprecated

Pinata IPFS automatically pins the uploaded document using some implicitly encoding and hashing algorithm. This method is not used anymore to make it more explicit these parameters and therefore less dependent on the default implementation of Pinata

Defined in

api/index.ts:29

validateAppDataDoc

•validateAppDataDoc : (appDataDoc :[AnyAppDataDocVersion](#)) =>Promise <[ValidationResult](#)

=validateAppDataDoc

Type declaration

► (appDataDoc):Promise <[ValidationResult](#)

Parameters

Name Type appDataDoc [AnyAppDataDocVersion](#)

Returns

Promise <[ValidationResult](#)

Defined in

api/index.ts:16 [Previous v1_0_0](#) [Next Ipfs](#)