@cowprotocol/cow-sdk

Namespaces

- OnchainOrderData
- OrderCancellationError
- OrderPostError
- PriceEstimationError

Enumerations

- BuyTokenDestination
- DurationType
- EcdsaSigningScheme
- OrderClass
- OrderKind
- OrderQuoteSideKindBuy
- OrderQuoteSideKindSell
- OrderStatus
- PollResultCode
- PriceQuality
- ProofLocation
- SellTokenSource
- SigningScheme
- StartTimeValue
- SupportedChainId

Classes

- ConditionalOrder
- ConditionalOrderFactory
- CowError
- Multiplexer
- OrderBookApi
- OrderBookApiError
- OrderSigningUtils
- SubgraphApi
- Twap

Interfaces

- ApiContext
- ConditionalOrderArguments
- EnrichedOrder
- FetchParams
- IpfsConfig
- IsNotValid
- IsValid
- PollResultDontTryAgain
- PollResultSuccess
- PollResultTryAtEpoch
- PollResultTryNextBlock
- PollResultTryOnBlock
- PollResultUnexpectedError
- RequestOptions
- SignOrderCancellationParams
- SignOrderCancellationsParams
- SignOrderParams
- TwapData
- TwapStruct

Type Aliases

Address

TAddress :string

20 byte Ethereum address encoded as a hex with0x prefix.

Defined in

external/cow-sdk/src/order-book/generated/models/Address.ts:8

ApiBaseUrls

TApiBaseUrls :Record <SupportedChainId ,string

Defined in

external/cow-sdk/src/common/configs.ts:47

AppData

TAppData :string

The string encoding of a JSON object representing someappData . The format of the JSON expected in theappData field is definedhere .

Defined in

external/cow-sdk/src/order-book/generated/models/AppData.ts:11

AppDataHash

TAppDataHash :string

32 bytes encoded as hex with0x prefix. It's expected to be the hash of the stringified JSON object representing theappData .

Defined in

external/cow-sdk/src/order-book/generated/models/AppDataHash.ts:10

AppDataObject

TAppDataObject :Object

AnappData document that is registered with the API.

Type declaration

Name Type fullAppData? AppData

Defined in

external/cow-sdk/src/order-book/generated/models/AppDataObject.ts:10

Auction

TAuction :Object

A batch auction for solving.

Type declaration

Name Type Description block? number The block number for the auction. Orders and prices are guaranteed to be valid on this block. Proposed settlements should be valid for this block as well. id? number The unique identifier of the auction. Increment whenever the backend creates a new auction. latestSettlementBlock? number The latest block on which a settlement has been processed.NOTE : Under certain conditions it is possible for a settlement to have been mined as part ofblock but not have yet been processed. orders? AuctionOrder [] The solvable orders included in the auction. prices? AuctionPrices -

Defined in

external/cow-sdk/src/order-book/generated/models/Auction.ts:12

AuctionOrder

TAuctionOrder :Object

A solvable order included in the current batch auction. Contains the data forwarded to solvers for solving.

Type declaration

Name Type Description appData AppDataHash - buyAmount TokenAmount seeOrderParameters::buyAmount buyToken Address seeOrderParameters::buyToken buyTokenBalance BuyTokenDestination seeOrderParameters::buyTokenBalance class OrderClass - executed TokenAmount Currently executed amount of sell/buy token, depending on the order kind. kind OrderKind seeOrderParameters::kind owner Address - partiallyFillable boolean seeOrderParameters::partiallyFillable postInteractions InteractionData [] The post-interactions that need to be executed after the execution of the order. preInteractions InteractionData [] The pre-interactions that need to be executed before the first execution of the order. protocolFees FeePolicy [] The fee policies that are used to compute the protocol fees for this order. receiverAddress |null seeOrderParameters::receiver sellAmount TokenAmount seeOrderParameters::sellAmount sellToken Address seeOrderParameters::sellToken sellTokenBalance SellTokenSource seeOrderParameters::sellTokenBalance signature Signature - uid UID - userFee TokenAmount seeOrderParameters::feeAmount validTo number seeOrderParameters::validTo

Defined in

external/cow-sdk/src/order-book/generated/models/AuctionOrder.ts:21

AuctionPrices

TAuctionPrices :Record <string ,BigUint

The reference prices for all traded tokens in the auction as a mapping from token addresses to a price denominated in native token (i.e. 1e18 represents a token that trades one to one with the native token). These prices are used for solution competition for computing surplus and converting fees to native token.

Defined in

external/cow-sdk/src/order-book/generated/models/AuctionPrices.ts:14

BigUint

TBigUint :string

A big unsigned integer encoded in decimal.

Defined in

external/cow-sdk/src/order-book/generated/models/BigUint.ts:8

BlockInfo

TBlockInfo :Object

Type declaration

Name Type blockNumber number blockTimestamp number

Defined in

external/cow-sdk/src/composable/types.ts:102

CallData

TCallData :string

Somecalldata sent to a contract in a transaction encoded as a hex with0x prefix.

Defined in

external/cow-sdk/src/order-book/generated/models/CallData.ts:8

CompetitionAuction

TCompetitionAuction :Object

The components that describe a batch auction for the solver competition.

Type declaration

Name Type Description orders? UID [] The UIDs of the orders included in the auction. prices?AuctionPrices -

Defined in

external/cow-sdk/src/order-book/generated/models/CompetitionAuction.ts:12

ConditionalOrderParams

TConditionalOrderParams :Object

Type declaration

Name Type handler string salt string staticInput string

Defined in

external/cow-sdk/src/composable/types.ts:13

ConditionalOrderRegistry

TConditionalOrderRegistry :Record <string ,FromParams <unknown ,unknown

Defined in

external/cow-sdk/src/composable/ConditionalOrderFactory.ts:5

ContextFactory

TContextFactory :Object

A factory and it's arguments that are called at transaction mining time to generate the context for a conditional order(s).

This allows to support the case where conditional orders may want tocommence validity at the time of transaction mining, like in the case of aTWAP executed by a DAO orSafe that takes a reasonable amount of time to aggregate signatures or collect votes.

Remarks

This is used in conjunction withsetRootWithContext orcreateWithContext .

Type declaration

Name Type address string factoryArgs? {args :unknown [] ;argsType :string [] } factoryArgs.args unknown [] factoryArgs.argsType string []

Defined in

external/cow-sdk/src/composable/types.ts:44

CowEnv

TCowEnv :"prod" |"staging"

The environment to use for the Cow API.

Defined in

external/cow-sdk/src/common/configs.ts:35

DurationOfPart

TDurationOfPart : {durationType :AUTO } | {duration :BigNumber ;durationType LIMIT_DURATION }

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:138

EcdsaSignature

TEcdsaSignature :string

65 bytes encoded as hex with0x prefix.r || s || v from the spec.

Defined in

external/cow-sdk/src/order-book/generated/models/EcdsaSignature.ts:8

EthflowData

TEthflowData :Object

Provides the additional data for ethflow orders.

Type declaration

Name Type Description refundTxHash TransactionHash |null Specifies in which transaction the order was refunded. If this field is null the order was not yet refunded. userValidTo number Describes thevalidTo of an order ethflow order.NOTE : For ethflow orders, thevalidTo encoded in the smart contract istype(uint256).max .

Defined in

external/cow-sdk/src/order-book/generated/models/EthflowData.ts:10

FeePolicy

TFeePolicy :Surplus |Volume

Defines the ways to calculate the protocol fee.

Defined in

external/cow-sdk/src/order-book/generated/models/FeePolicy.ts:11

FromParams

TFromParams <D ,S

: (params :ConditionalOrderParams) =>ConditionalOrder <D ,S

Type parameters

Name D S

Type declaration

▸ (params):ConditionalOrder <D ,S

Parameters

Name Type params ConditionalOrderParams

Returns

ConditionalOrder <D ,S

Defined in

external/cow-sdk/src/composable/ConditionalOrderFactory.ts:4

GetOrdersRequest

TGetOrdersRequest :Object

The parameters for thegetOrders request.

Type declaration

Name Type limit? number offset? number owner Address

Defined in

external/cow-sdk/src/order-book/api.ts:64

InteractionData

TInteractionData :Object

Type declaration

Name Type Description call_data? CallData [] The call data to be used for the interaction. target?Address - value? TokenAmount -

Defined in

external/cow-sdk/src/order-book/generated/models/InteractionData.ts:9

IsValidResult

TIsValidResult :IsValid |IsNotValid

Defined in

external/cow-sdk/src/composable/types.ts:164

NativePriceResponse

TNativePriceResponse :Object

The estimated native price for the token

Type declaration

Name Type Description price? number Estimated price of the token.

Defined in

external/cow-sdk/src/order-book/generated/models/NativePriceResponse.ts:9

OnchainOrderData

TOnchainOrderData :Object

Type declaration

Name Type Description placementError? placementError Describes the error, if the order placement was not successful. This could happen, for example, if thevalidTo is too high, or no valid quote was found or generated. sender Address If orders are placed as on-chain orders, the owner of the order might be a smart contract, but not the user placing the order. The actual user will be provided in this field.

Defined in

external/cow-sdk/src/order-book/generated/models/OnchainOrderData.ts:7

external/cow-sdk/src/order-book/generated/models/OnchainOrderData.ts:24

Order

TOrder :OrderCreation &OrderMetaData

Defined in

external/cow-sdk/src/order-book/generated/models/Order.ts:8

OrderCancellation

TOrderCancellation :Object

EIP-712 signature of structOrderCancellation(bytes orderUid) from the order's owner.

Type declaration

Name Type Description signature EcdsaSignature OrderCancellation signed by owner signingScheme EcdsaSigningScheme -

Defined in

external/cow-sdk/src/order-book/generated/models/OrderCancellation.ts:13

OrderCancellationError

TOrderCancellationError :Object

Type declaration

Name Type description string errorType errorType

Defined in

external/cow-sdk/src/order-book/generated/models/OrderCancellationError.ts:5

external/cow-sdk/src/order-book/generated/models/OrderCancellationError.ts:10

OrderCancellations

TOrderCancellations :Object

EIP-712 signature of struct OrderCancellations { orderUid: bytes[] } from the order's owner.

Type declaration

Name Type Description orderUids? UID [] UIDs of orders to cancel. signatureEcdsaSignature OrderCancellation signed by the owner. signingScheme EcdsaSigningScheme -

Defined in

external/cow-sdk/src/order-book/generated/models/OrderCancellations.ts:13

OrderCreation

TOrderCreation :Object

Data a user provides when creating a new order.

Type declaration

Name Type Description appData AppData |AppDataHash This field comes in two forms for backward compatibility. The hash form will eventually stop being accepted. appDataHash? AppDataHash |null May be set for debugging purposes. If set, this field is compared to what the backend internally calculates as the app data hash based on the contents ofappData . If

the hash does not match, an error is returned. If this field is set, thenappData MUST be a string encoding of a JSON object. buyAmount TokenAmount seeOrderParameters::buyAmount buyToken Address seeOrderParameters::buyToken buyTokenBalance? BuyTokenDestination seeOrderParameters::buyTokenBalance feeAmount TokenAmount seeOrderParameters::feeAmount from? Address |null If set, the backend enforces that this address matches what is decoded as thesigner of the signature. This helps catch errors with invalid signature encodings as the backend might otherwise silently work with an unexpected address that for example does not have any balance. kind OrderKind seeOrderParameters::kind partiallyFillable boolean seeOrderParameters::partiallyFillable quoteId? number |null Orders can optionally include a quote ID. This way the order can be linked to a quote and enable providing more metadata when analysing order slippage. receiver? Address |null seeOrderParameters::receiver sellAmount TokenAmount seeOrderParameters::sellAmount sellToken Address seeOrderParameters::sellToken sellTokenBalance? SellTokenSource seeOrderParameters::sellTokenBalance signature Signature - signingScheme SigningScheme - validTo number seeOrderParameters::validTo

Defined in

external/cow-sdk/src/order-book/generated/models/OrderCreation.ts:18

OrderMetaData

TOrderMetaData :Object

Extra order data that is returned to users when querying orders but not provided by users when creating orders.

Type declaration

Name Type Description availableBalance? TokenAmount |null Unused field that is currently always set tonull and will be removed in the future.Deprecated class OrderClass - creationDate string Creation time of the order. Encoded as ISO 8601 UTC. ethflowData? EthflowData - executedBuyAmount BigUint The total amount ofbuyToken that has been executed for this order. executedFeeAmount BigUint The total amount of fees that have been executed for this order. executedSellAmount BigUint The total amount ofsellToken that has been executed for this order including fees. executedSellAmountBeforeFees BigUint The total amount ofsellToken that has been executed for this order without fees. executedSurplusFee?BigUint |null Surplus fee that the limit order was executed with. fullAppData? string |null FullappData , which the contract-levelappData is a hash of. SeeOrderCreation for more information. fullFeeAmount? TokenAmount Amount that the signed fee would be without subsidies. invalidated boolean Has this order been invalidated? isLiquidityOrder? boolean Liquidity orders are functionally the same as normal smart contract orders but are not placed with the intent of actively getting traded. Instead they facilitate the trade of normal orders by allowing them to be matched against liquidity orders which uses less gas and can have better prices than external liquidity. As such liquidity orders will only be used in order to improve settlement of normal orders. They should not be expected to be traded otherwise and should not expect to get surplus. onchainOrderData? OnchainOrderData There is some data only available for orders that are placed on-chain. This data can be found in this object. onchainUser? Address This represents the actual trader of an on-chain order. ### ethflow orders In this case, theowner would be theEthFlow contract andnot the actual trader. owner Address - status OrderStatus Order status. uid UID -

Defined in

external/cow-sdk/src/order-book/generated/models/OrderMetaData.ts:19

OrderParameters

TOrderParameters :Object

Order parameters.

Type declaration

Name Type Description appData AppDataHash - buyAmount TokenAmount Amount ofbuyToken to be bought in atoms. buyToken Address ERC-20 token to be bought. buyTokenBalance?BuyTokenDestination - feeAmount TokenAmount feeRatio * sellAmount + minimal_fee in atoms. kind OrderKind The kind is either a buy or sell order. partiallyFillable boolean Is the order fill-or-kill or partially fillable? receiver? Address |null An optional Ethereum address to receive the proceeds of the trade instead of the owner (i.e. the order signer). sellAmount TokenAmount Amount ofsellToken to be sold in atoms. sellToken Address ERC-20 token to be sold. sellTokenBalance?SellTokenSource - signingScheme? SigningScheme - validTo number Unix timestamp (uint32) until which the order is valid.

Defined in

external/cow-sdk/src/order-book/generated/models/OrderParameters.ts:16

OrderPostError

TOrderPostError :Object

Type declaration

Name Type description string errorType errorType

Defined in

external/cow-sdk/src/order-book/generated/models/OrderPostError.ts:5

external/cow-sdk/src/order-book/generated/models/OrderPostError.ts:10

OrderQuoteRequest

TOrderQuoteRequest :OrderQuoteSide &OrderQuoteValidity & {appData? :AppData |AppDataHash ;appDataHash? :AppDataHash ;buyToken :Address ;buyTokenBalance? :BuyTokenDestination ;from :Address ;onchainOrder? :any ;priceQuality? :PriceQuality ;receiver? :Address |null ;sellToken :Address ;sellTokenBalance? :SellTokenSource ;signingScheme? :SigningScheme }

Request fee and price quote.

Defined in

external/cow-sdk/src/order-book/generated/models/OrderQuoteRequest.ts:18

OrderQuoteResponse

TOrderQuoteResponse :Object

An order quoted by the backend that can be directly signed and submitted to the order creation backend.

Type declaration

Name Type Description expiration string Expiration date of the offered fee. Order service might not accept the fee after this expiration date. Encoded as ISO 8601 UTC. from? Address - id? number Quote ID linked to a quote to enable providing more metadata when analysing order slippage. quote OrderParameters - verified boolean Whether it was possible to verify that the quoted amounts are accurate using a simulation.

Defined in

external/cow-sdk/src/order-book/generated/models/OrderQuoteResponse.ts:13

OrderQuoteSide

TOrderQuoteSide : {kind :OrderQuoteSideKindSell ;sellAmountBeforeFee :TokenAmount } | {kind :OrderQuoteSideKindSell ;sellAmountAfterFee :TokenAmount } | {buyAmountAfterFee :TokenAmount ;kind :OrderQuoteSideKindBuy }

The buy or sell side when quoting an order.

Defined in

external/cow-sdk/src/order-book/generated/models/OrderQuoteSide.ts:12

OrderQuoteValidity

TOrderQuoteValidity : {validTo? :number } | {validFor? :number }

The validity for the order.

Defined in

external/cow-sdk/src/order-book/generated/models/OrderQuoteValidity.ts:8

Orders

TOrders :Record <string ,ConditionalOrder <unknown ,unknown

Defined in

external/cow-sdk/src/composable/Multiplexer.ts:15

OwnerContext

TOwnerContext :Object

Type declaration

Name Type chainId SupportedChainId owner string provider providers.Provider

Defined in

external/cow-sdk/src/composable/types.ts:85

PartialApiContext

TPartialApiContext :Partial <ApiContext

Override some properties of theApiContext .

Defined in

external/cow-sdk/src/common/configs.ts:40

PayloadLocationEmitted

TPayloadLocationEmitted :Object

Payload for emitting a merkle root to a ComposableCoW-enabled Safe.

If settingProofLocation.EMITTED , this type should be used as thedata in theProof struct.

Type declaration

Name Type proofs ProofWithParams []

Defined in

external/cow-sdk/src/composable/types.ts:70

PollParams

TPollParams :OwnerContext & {blockInfo? :BlockInfo ;offchainInput? :string ;orderBookApi :OrderBookApi ;proof? :string [] }

Defined in

external/cow-sdk/src/composable/types.ts:91

PollResult

TPollResult :PollResultSuccess |PollResultErrors

Defined in

external/cow-sdk/src/composable/types.ts:107

PollResultErrors

TPollResultErrors :PollResultTryNextBlock |PollResultTryOnBlock |PollResultTryAtEpoch |PollResultUnexpectedError |PollResultDontTryAgain

Defined in

external/cow-sdk/src/composable/types.ts:109

PreSignature

TPreSignature :string

Empty signature bytes. Used for "presign" signatures.

Defined in

external/cow-sdk/src/order-book/generated/models/PreSignature.ts:8

PriceEstimationError

TPriceEstimationError :Object

Type declaration

Name Type description string errorType errorType

Defined in

external/cow-sdk/src/order-book/generated/models/PriceEstimationError.ts:5

external/cow-sdk/src/order-book/generated/models/PriceEstimationError.ts:10

ProofStruct

TProofStruct :Object

A struct for a proof that can be used withsetRoot andsetRootWithContext on a ComposableCoW-enabled Safe.

Type declaration

Name Type data string |"0x" location ProofLocation

Defined in

external/cow-sdk/src/composable/types.ts:58

ProofWithParams

TProofWithParams :Object

A proof for a conditional order and it's parameters.

Type declaration

Name Type params ConditionalOrderParams proof string []

Defined in

external/cow-sdk/src/composable/types.ts:78

ProtocolAppData

TProtocolAppData :Object

Defined in

external/cow-sdk/src/order-book/generated/models/ProtocolAppData.ts:5

Signature

TSignature :EcdsaSignature |PreSignature

A signature.

Defined in

external/cow-sdk/src/order-book/generated/models/Signature.ts:11

SigningResult

TSigningResult :Object

Encoded signature including signing scheme for the order.

Type declaration

Name Type signature string signingScheme EcdsaSigningScheme

Defined in

external/cow-sdk/src/order-signing/types.ts:13

SolverCompetitionResponse

TSolverCompetitionResponse :Object

The settlements submitted by every solver for a specific auction. TheauctionId corresponds to the id external solvers are provided with.

Type declaration

Name Type Description auction? CompetitionAuction - auctionId? number The ID of the auction the competition info is for. competitionSimulationBlock? number - gasPrice? number Gas price used for ranking solutions. liquidityCollectedBlock? number - solutions? SolverSettlement [] Maps from solver name to object describing that solver's settlement. transactionHash? TransactionHash |null The hash of the transaction that the winning solution of this info was submitted in.

Defined in

external/cow-sdk/src/order-book/generated/models/SolverCompetitionResponse.ts:15

SolverSettlement

TSolverSettlement :Object

Type declaration

Name Type Description callData? CallData Transactioncalldata that is executed on-chain if the settlement is executed. clearingPrices? Record <string ,BigUint

> The prices of tokens for settled user orders as passed to the settlement contract. objective? {cost? :number ;fees? :number ;gas? :number ;surplus? :number ;total? :number } - objective.cost? number - objective.fees? number - objective.gas? number - objective.surplus? number - objective.total? number The total objective value used for ranking solutions. orders? {executedAmount? :BigUint ;id? :UID }[] Touched orders. score? BigUint |null The score of the current auction as defined inCIP-20 . It isnull for old auctions. solver? string Name of the solver. solverAddress? string The address used by the solver to execute the settlement on-chain. This field is missing for old settlements, the zero address has been used instead. uninternalizedCallData? CallData Fullcalldata as generated from the original solver output. It can be different from the executed transaction if part of the settlements are internalised (use internal liquidity in lieu of trading against on-chain liquidity). This field is omitted in case it coincides withcallData .

Defined in

external/cow-sdk/src/order-book/generated/models/SolverSettlement.ts:9

StartTime

TStartTime : {startType :AT_MINING_TIME } | {epoch :BigNumber ;startType :AT_EPOCH }

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:147

Surplus

TSurplus :Object

The protocol fee is taken as a percent of the surplus.

Type declaration

Name Type factor number max_volume_factor number

Defined in

external/cow-sdk/src/order-book/generated/models/Surplus.ts:8

TokenAmount

TTokenAmount :string

Amount of a token.uint256 encoded in decimal.

Defined in

external/cow-sdk/src/order-book/generated/models/TokenAmount.ts:8

TotalSurplus

TTotalSurplus :Object

The total surplus.

Type declaration

Name Type Description totalSurplus? string The total surplus.

Defined in

external/cow-sdk/src/order-book/generated/models/TotalSurplus.ts:9

Trade

TTrade :Object

Trade data such as executed amounts, fees,orderUid andblock number.

Type declaration

Name Type Description blockNumber number Block in which trade occurred. buyAmountTokenAmount Total amount ofbuyToken received in this trade. buyToken Address Address of token bought. logIndex number Index in which transaction was included in block. orderUid UID UID of the order matched by this trade. ownerAddress Address of trader. sellAmount TokenAmount Total amount ofsellToken that has been executed for this trade (including fees). sellAmountBeforeFees BigUint The total amount ofsellToken that has been executed for this order without fees. sellTokenAddress Address of token sold. txHash TransactionHash |null Transaction hash of the corresponding settlement transaction containing the trade (if available).

Defined in

external/cow-sdk/src/order-book/generated/models/Trade.ts:15

TransactionHash

TTransactionHash :string

32 byte digest encoded as a hex with0x prefix.

Defined in

external/cow-sdk/src/order-book/generated/models/TransactionHash.ts:8

TwapDataBase

TTwapDataBase :Object

Base parameters for a TWAP order. Shared by:

- TwapStruct (modelling the contract's struct used forstaticInput
-).
- TwapData (modelling the friendly SDK interface).

Type declaration

Name Type Description appData string Meta-data associated with the order. Normally would be the keccak256 hash of the document generated inhttp://github.com/cowprotocol/app-data This hash should have been uploaded to the APIhttps://api.cow.fi/docs/#/default/put_api_v1_app_data__app_data_hash _ and potentially to other data availability protocols like IPFS. buyToken string which token to buy receiver string who to send the tokens to sellToken string which token to sell

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:41

UID

TUID :string

Unique identifier for the order: 56 bytes encoded as hex with0x prefix. Bytes 0..32 are the order digest, bytes 30..52 the owner address and bytes 52..56 the expiry (validTo) as auint32 unix epoch timestamp.

Defined in

external/cow-sdk/src/order-book/generated/models/UID.ts:11

UnsignedOrder

TUnsignedOrder :Omit <OrderParameters ,"receiver"

& {receiver :string }

Unsigned order intent to be placed.

Defined in

external/cow-sdk/src/order-signing/types.ts:8

Volume

TVolume :Object

The protocol fee is taken as a percent of the order volume.

Type declaration

Name Type factor number

Defined in

external/cow-sdk/src/order-book/generated/models/Volume.ts:8

Variables

ALL_SUPPORTED_CHAIN_IDS

•Const ALL_SUPPORTED_CHAIN_IDS :SupportedChainId []

The list of supported chains.

Defined in

external/cow-sdk/src/common/consts.ts:13

BUY_ETH_ADDRESS

•Const BUY_ETH_ADDRESS :"0xEeeeeEeeeEeEeeEeEeEeeEEEeeeeEeeeeeeeEEeE"

Defined in

external/cow-sdk/src/common/consts.ts:3

COMPOSABLE_COW

•Const COMPOSABLE_COW :"0xfdaFc9d1902f4e0b84f65F49f244b32b31013b74"

Defined in

external/cow-sdk/src/common/consts.ts:5

COMPOSABLE_COW_CONTRACT_ADDRESS

•Const COMPOSABLE_COW_CONTRACT_ADDRESS :Record <SupportedChainId ,string

An object containing the addresses of theComposableCow contracts for each supported chain.

Defined in

external/cow-sdk/src/common/consts.ts:51

CONDITIONAL_ORDER_PARAMS_ABI

•Const CONDITIONAL_ORDER_PARAMS_ABI :string []

Defined in

external/cow-sdk/src/composable/utils.ts:19

COW_PROTOCOL_SETTLEMENT_CONTRACT_ADDRESS

•Const COW_PROTOCOL_SETTLEMENT_CONTRACT_ADDRESS :Record <SupportedChainId ,string

An object containing the addresses of the CoW Protocol settlement contracts for each supported chain.

Defined in

external/cow-sdk/src/common/consts.ts:36

COW_PROTOCOL_VAULT_RELAYER_ADDRESS

•Const COW_PROTOCOL_VAULT_RELAYER_ADDRESS :Record <SupportedChainId ,string

An object containing the addresses of the CoW Protocol Vault realyer contracts for each supported chain.

Defined in

external/cow-sdk/src/common/consts.ts:41

CURRENT_BLOCK_TIMESTAMP_FACTORY_ADDRESS

•Const CURRENT_BLOCK_TIMESTAMP_FACTORY_ADDRESS :"0x52eD56Da04309Aca4c3FECC595298d80C2f16BAc"

The address of theCurrentBlockTimestampFactory contract

NOTE : This is used in the event that TWAP's have at0 of0 .

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:26

DEFAULT_BACKOFF_OPTIONS

•Const DEFAULT_BACKOFF_OPTIONS :BackoffOptions

The default backoff options for CoW Protocol's API

See

Backoff configuration:https://www.npmjs.com/package/@insertish/exponential-backoff

Defined in

external/cow-sdk/src/order-book/request.ts:41

DEFAULT_CONDITIONAL_ORDER_REGISTRY

•Const DEFAULT_CONDITIONAL_ORDER_REGISTRY :ConditionalOrderRegistry

Defined in

external/cow-sdk/src/composable/orderTypes/index.ts:5

DEFAULT_COW_API_CONTEXT

•Const DEFAULT_COW_API_CONTEXT :ApiContext

The default CoW Protocol API context.

Defined in

external/cow-sdk/src/common/configs.ts:80

DEFAULT_IPFS_READ_URI

•Const DEFAULT_IPFS_READ_URI :"https://gnosis.mypinata.cloud/ipfs"

Defined in

external/cow-sdk/src/common/ipfs.ts:1

DEFAULT_IPFS_WRITE_URI

•Const DEFAULT_IPFS_WRITE_URI :"https://api.pinata.cloud"

Defined in

external/cow-sdk/src/common/ipfs.ts:2

DEFAULT_LIMITER_OPTIONS

•Const DEFAULT_LIMITER_OPTIONS :RateLimiterOpts

The default rate limiter options for CoW Protocol's API.

CAUTION : The CoW Protocol OrderBook API is limited to 5 requests per second per IP.

Defined in

external/cow-sdk/src/order-book/request.ts:59

ENVS_LIST

•Const ENVS_LIST :CowEnv []

The list of available environments.

Defined in

external/cow-sdk/src/common/configs.ts:75

EXTENSIBLE_FALLBACK_HANDLER

•Const EXTENSIBLE_FALLBACK_HANDLER :"0x2f55e8b20D0B9FEFA187AA7d00B6Cbe563605bF5"

Defined in

external/cow-sdk/src/common/consts.ts:4

EXTENSIBLE_FALLBACK_HANDLER_CONTRACT_ADDRESS

•Const EXTENSIBLE_FALLBACK_HANDLER_CONTRACT_ADDRESS :Record <SupportedChainId ,string

An object containing the addresses of theExtensibleFallbackHandler contracts for each supported chain.

Defined in

external/cow-sdk/src/common/consts.ts:46

MAX_FREQUENCY

•Const MAX_FREQUENCY :BigNumber

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:29

MAX_UINT32

•Const MAX_UINT32 :BigNumber

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:28

ORDER_BOOK_PROD_CONFIG

•Const ORDER_BOOK_PROD_CONFIG :ApiBaseUrls

An object containingproduction environment base URLs for each supportedchainId .

See

https://api.cow.fi/docs/#/

Defined in

external/cow-sdk/src/order-book/api.ts:38

ORDER_BOOK_STAGING_CONFIG

•Const ORDER_BOOK_STAGING_CONFIG :ApiBaseUrls

An object containingstaging environment base URLs for each supportedchainId .

Defined in

external/cow-sdk/src/order-book/api.ts:47

SUBGRAPH_PROD_CONFIG

•Const SUBGRAPH_PROD_CONFIG :SubgraphApiBaseUrls

CoW Protocol Production Subgraph API configuration.

See

- https://api.thegraph.com/subgraphs/name/cowprotocol/cow
- https://api.thegraph.com/subgraphs/name/cowprotocol/cow-gc

Defined in

external/cow-sdk/src/subgraph/api.ts:24

SUBGRAPH_STAGING_CONFIG

•Const SUBGRAPH_STAGING_CONFIG :SubgraphApiBaseUrls

CoW Protocol Staging Subgraph API configuration.

Deprecated

See

- https://api.thegraph.com/subgraphs/name/cowprotocol/cow-staging
- https://api.thegraph.com/subgraphs/name/cowprotocol/cow-gc-staging

Defined in

external/cow-sdk/src/subgraph/api.ts:36

TWAP_ADDRESS

•Const TWAP_ADDRESS :"0x6cF1e9cA41f7611dEf408122793c358a3d11E5a5"

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:20

logPrefix

•Const logPrefix :"cow-sdk:"

Defined in

external/cow-sdk/src/common/cow-error.ts:10

Functions

DEFAULT_TOKEN_FORMATTER

▸DEFAULT_TOKEN_FORMATTER (address ,amount):string

Parameters

Name Type address string amount BigNumber

Returns

string

Defined in

external/cow-sdk/src/composable/utils.ts:21

createSetDomainVerifierTx

▸createSetDomainVerifierTx (domain ,verifier):string

Parameters

Name Type domain string verifier string

Returns

string

Defined in

external/cow-sdk/src/composable/utils.ts:44

decodeParams

▸decodeParams (encoded):ConditionalOrderParams

Decode theConditionalOrderParams for the conditional order.

Parameters

Name Type Description encoded string The encoded conditional order.

Returns

ConditionalOrderParams

The decoded conditional order.

Defined in

external/cow-sdk/src/composable/utils.ts:68

encodeParams

▸encodeParams (params):string

Encode theConditionalOrderParams for the conditional order.

Parameters

Name Type Description params ConditionalOrderParams TheConditionalOrderParams struct representing the conditional order as taken from a merkle tree.

Returns

string

The ABI-encoded conditional order.

See

ConditionalOrderParams

Defined in

external/cow-sdk/src/composable/utils.ts:58

formatEpoch

▸formatEpoch (epoch):string

Parameters

Name Type epoch number

Returns

string

Defined in

external/cow-sdk/src/composable/utils.ts:97

fromStructToOrder

▸fromStructToOrder (order):Order

Parameters

Name Type order DataStruct

Returns

Order

Defined in

external/cow-sdk/src/composable/utils.ts:135

getBlockInfo

▸getBlockInfo (provider):Promise <BlockInfo

Parameters

Name Type provider Provider

Returns

Promise <BlockInfo

Defined in

external/cow-sdk/src/composable/utils.ts:88

getDomainVerifier

▸getDomainVerifier (safe ,domain ,chainId ,provider):Promise <string

Parameters

Name Type safe string domain string chainId SupportedChainId provider Provider

Returns

Promise <string

Defined in

external/cow-sdk/src/composable/utils.ts:31

isComposableCow

▸isComposableCow (handler ,chainId):boolean

Parameters

Name Type handler string chainId SupportedChainId

Returns

boolean

Defined in

external/cow-sdk/src/composable/utils.ts:27

isExtensibleFallbackHandler

▸isExtensibleFallbackHandler (handler ,chainId):boolean

Parameters

Name Type handler string chainId SupportedChainId

Returns

boolean

Defined in

external/cow-sdk/src/composable/utils.ts:23

isValidAbi

▸isValidAbi (types ,values):boolean

Helper method for validating ABI types.

Parameters

Name Type Description types readonly (string |ParamType)[] ABI types to validate against. values any [] The values to validate.

Returns

boolean

Whether the values are valid ABI for the given types.

Defined in

external/cow-sdk/src/composable/utils.ts:79

mapAddressToSupportedNetworks

▸mapAddressToSupportedNetworks (address):Record <SupportedChainId ,string

Parameters

Name Type address string

Returns

Record <SupportedChainId ,string

Defined in

external/cow-sdk/src/common/consts.ts:29

mapSupportedNetworks

▸mapSupportedNetworks <T

(value):Record <SupportedChainId ,T

Type parameters

Name T

Parameters

Name Type value (chainId :SupportedChainId) =>T

Returns

Record <SupportedChainId ,T

Defined in

external/cow-sdk/src/common/consts.ts:17

▸mapSupportedNetworks <T

(value):Record <SupportedChainId ,T

Type parameters

Name T

Parameters

Name Type value T

Returns

Record <SupportedChainId ,T

Defined in

external/cow-sdk/src/common/consts.ts:18

request

▸request <T

(baseUrl ,«destructured» ,rateLimiter ,backoffOpts):Promise <T

Helper function to make a rate-limited request to an API.

Type parameters

Name T

Parameters

Name Type Description baseUrl string The base URL of the API. «destructured»FetchParams - rateLimiter RateLimiter The rate limiter to use. backoffOpts Partial <IBackOffOptions

The backoff options to use.

Returns

Promise <T

The response of the request.

Throws

If the API returns an error or if the request fails.

Defined in

external/cow-sdk/src/order-book/request.ts:104

transformDataToStruct

▸transformDataToStruct (data):TwapStruct

Transform parameters into a native struct.

Parameters

Name Type Description data TwapData As passed by the consumer of the API.

Returns

TwapStruct

A formatted struct as expected by the smart contract.

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:538

transformStructToData

▸transformStructToData (struct):TwapData

Transform parameters into a TWAP order struct.

Parameters

Name Type struct TwapStruct

Returns

TwapData

A formatted struct as expected by the smart contract.

Defined in

external/cow-sdk/src/composable/orderTypes/Twap.ts:580 Previous Readme Next OnchainOrderData