# inbox

## Classes

### InboxTools

Tools for interacting with the inbox and bridge contracts

#### Methods

##### estimateArbitrumGas()

private

estimateArbitrumGas ( transactionl2Request : RequiredTransactionRequestType , l2Provider : Provider ) :

Promise < GasComponentsWithL2Part

> We should use nodeInterface to get the gas estimate is because we are making a delayed inbox message which doesn't need l1 calldata gas fee part.

###### Parameters

Parameter Type transactionl2Request RequiredTransactionRequestType l2Provider Provider

###### Returns

Promise <GasComponentsWithL2Part

###### Source

inbox/inbox.ts:120

##### findFirstBlockBelow()

private

findFirstBlockBelow ( blockNumber :

number , blockTimestamp :

number ) :

Promise < Block

> Find the first (or close to first) block whose number is below the provided number, and whose timestamp is below the provided timestamp

###### Parameters

Parameter Type Description blockNumber number blockTimestamp number

###### Returns

Promise <Block

###### Source

inbox/inbox.ts:83

##### forceInclude()

forceInclude < T

> ( messageDeliveredEvent ? :

T , overrides ? : Overrides ) :

Promise < T

extends

ForceInclusionParams

? ContractTransaction :

null

| ContractTransaction

> Force includes all eligible messages in the delayed inbox. The inbox contract doesnt allow a message to be force-included until after a delay period has been completed.

###### Type parameters

Type parameter T extends undefined |ForceInclusionParams

###### Parameters

Parameter Type Description messageDeliveredEvent ? T Provide this to include all messages up to this one. Responsibility is on the caller to check the eligibility of this event. overrides ? Overrides -

###### Returns

Promise <T extends ForceInclusionParams ?ContractTransaction :null |ContractTransaction

The force include transaction, or null if no eligible message were found for inclusion

###### Source

inbox/inbox.ts:304

#### getEventsAndIncreaseRange()

private

getEventsAndIncreaseRange ( bridge : Bridge , searchRangeBlocks :

number , maxSearchRangeBlocks :

number , rangeMultiplier :

number ) :

Promise < FetchedEvent < MessageDeliveredEvent

> [ ]
>
> Look for force includable events in the search range blocks, if no events are found the search range is increased incrementally up to the max search range blocks.

###### Parameters

Parameter Type Description bridge Bridge searchRangeBlocks number maxSearchRangeBlocks number rangeMultiplier number -

###### Returns

Promise <FetchedEvent <MessageDeliveredEvent

> []>

###### Source

inbox/inbox.ts:204

**getForceIncludableBlockRange()**

private

getForceIncludableBlockRange ( blockNumberRangeSize :

number ) :

Promise < object

> Get a range of blocks within messages eligible for force inclusion emitted events

###### Parameters

Parameter Type Description blockNumberRangeSize number

###### Returns

Promise

Member Type Value endBlock number firstEligibleBlock.number startBlock number ... ###### Source[](https://docs.arbitrum.io/welcome/get-started#source-4) [inbox/inbox.ts:150](https://github.com/OffchainLabs/arbitrum-sdk/blob/c8aa61ee10c7bb2f622c2603f3d2a81287390d4b/src/lib/inbox/inbox.ts#L150) ##### getForceIncludableEvent()[](https://docs.arbitrum.io/welcome/get-started#getforceincludableevent) getForceIncludableEvent ( maxSearchRangeBlocks : number , startSearchRangeBlocks : number , rangeMultipler : number ) : Promise < null | ForceInclusionParams > Find the event of the latest message that can be force include ###### Parameters[](https://docs.arbitrum.io/welcome/get-started#parameters-5) Parameter Type Default value Description maxSearchRangeBlocks number undefined The max range of blocks to search in. Defaults to 3 * 6545 ( = ~3 days) prior to the first eligble block startSearchRangeBlocks number 100 The start range of block to search in. Moves incrementally up to the maxSearchRangeBlocks. Defaults to 100; rangeMultipler number 2 - ###### Returns[](https://docs.arbitrum.io/welcome/get-started#returns-5) Promise Null if non can be found. ###### Source[](https://docs.arbitrum.io/welcome/get-started#source-5) [inbox/inbox.ts:255](https://github.com/OffchainLabs/arbitrum-sdk/blob/c8aa61ee10c7bb2f622c2603f3d2a81287390d4b/src/lib/inbox/inbox.ts#L255) ##### sendL2SignedTx()[](https://docs.arbitrum.io/welcome/get-started#sendl2signedtx) sendL2SignedTx ( signedTx : string ) : Promise < null | ContractTransaction > Send l2 signed tx using delayed inox, which won't alias the sender's adddress It will be automatically included by the sequencer on l2, if it isn't included within 24 hours, you can force include it ###### Parameters[](https://docs.arbitrum.io/welcome/get-started#parameters-6) Parameter Type Description signedTx string A signed transaction which can be sent directly to network, you can call inboxTools.signL2Message to get. ###### Returns[](https://docs.arbitrum.io/welcome/get-started#returns-6) Promise The l1 delayed inbox's transaction itself. ###### Source[](https://docs.arbitrum.io/welcome/get-started#source-6) [inbox/inbox.ts:349](https://github.com/OffchainLabs/arbitrum-sdk/blob/c8aa61ee10c7bb2f622c2603f3d2a81287390d4b/src/lib/inbox/inbox.ts#L349) ##### signL2Tx()[](https://docs.arbitrum.io/welcome/get-started#signl2tx) signL2Tx ( txRequest : RequiredTransactionRequestType , l2Signer : Signer ) : Promise < string > Sign a transaction with msg.to, msg.value and msg.data. You can use this as a helper to call inboxTools.sendL2SignedMessage above. ###### Parameters[](https://docs.arbitrum.io/welcome/get-started#parameters-7) Parameter Type Description txRequest RequiredTransactionRequestType - l2Signer Signer ethers Signer type, used to sign l2 transaction ###### Returns[](https://docs.arbitrum.io/welcome/get-started#returns-7) Promise The l1 delayed inbox's transaction signed data. ###### Source[](https://docs.arbitrum.io/welcome/get-started#source-7) [inbox/inbox.ts:377](https://github.com/OffchainLabs/arbitrum-sdk/blob/c8aa61ee10c7bb2f622c2603f3d2a81287390d4b/src/lib/inbox/inbox.ts#L377) [Edit this page](https://github.com/OffchainLabs/arbitrum-docs/edit/master/arbitrum-docs/sdk-docs/inbox/inbox.md) [Previous TransactionRequest](https://docs.arbitrum.io/sdk-docs/dataEntities/transactionRequest) [Next L1ToL2Message](https://docs.arbitrum.io/sdk-docs/message/L1ToL2Message)