SDK support for custom gas token Orbit chains

Arbitrum SDK is a TypeScript library for client-side interactions with Arbitrum. It provides common helper functionality as well as access to the underlying smart contract interfaces.

We've developed different versions of the SDK for different use cases. Orbit functionality can be found under theorbit andorbit-custom-fee-token tags of theArbitrum SDK package .

PUBLIC PREVIEW, MAINNET READY Orbit chains are nowMainnet ready ! Note that Orbit is still apublic preview capability - the Orbit product and its supporting documentation may change significantly as we capture feedback from readers like you.

To provide feedback, click theRequest an update button at the top of this document,join the Arbitrum Discord , or reach out to our team directly by completingthis form .

Custom gas token SDK

Theorbit-custom-fee-token SDK feature introduces a suite of APIs designed for the specific purpose of facilitatingbridging operations. These APIs are tailored for use cases where there is a need to transfer a native token or an ERC20 token from the parent chain to an orbit chain utilizing acustom gas token . The process involves an initial step of authorizing the native token on the parent chain. To streamline this, our APIs provide functionalities for token approval and offer a mechanism to verify the current status of this approval. Detailed below is a guide to how each of these APIs can be effectively utilized for distinct purposes:

1. EthBridger Context:
2.
    - APIs:
3.
    - getApproveFeeTokenRequest
4.
    - andapproveFeeToken
5.
    - .
6.
    - Purpose:
7.
    - These APIs are essential for the bridging of native tokens to the Orbit chain. They facilitate the necessary approval for native tokens, allowing contracts to manage fund movements. This process includes escrowing a specified amount of the native token on the parent chain and subsequently bridging it to the Orbit chain.
8. Erc20Bridger Context:
9.
    - APIs:
10.
    - getApproveFeeTokenRequest
11.
    - andapproveFeeToken
12.
    - .
13.
    - Purpose:
14.
    - In the scenario of bridging ERC20 assets to an Orbit chain, these APIs play a crucial role. Token Bridging on Arbitrum Nitro stack uses Retryable tickets and needs specific amount of fees to be paid for creation and redemption of the ticket. For more information about retryable tickets please take a look atthis
15.
    - part of our docs. The Orbit chain operates as a custom gas token network, necessitating the payment of fees in native tokens for the creation of retryable tickets and their redemption on the Orbit chain. To cover the submission and execution fees associated with retryable tickets on the Orbit chain, an adequate amount of native tokens must be approved and allocated on the parent chain to cover the fees.

Note that these APIs are just needed forcustom gas token orbit chains and for ETH-powered rollup and anytrust orbit chains, you don't need to use them.

Note that when native tokens are transferred to the custom gas token orbit chain, they function equivalently to ETH on EVM chains. This means that these tokens will exhibit behavior identical to that of ETH, which is the native currency on EVM chains. This similarity in functionality is a key feature to consider in transactions and operations within the orbit chain.

Note that everything else is under the hood, and the custom gas token code paths will be executed just if theL2Network object config has anativeToken field.

options: Orbit nodes