Sponsor UserOperation End Point

pm_sponsorUserOperation

All paymaster URLs allow you to use both Sponsorship and Token Paymasters. To switch between paymasters you will simply change the Mode of a specific request. We will highlight both types of requests below

This endpoint is responsible for calculating thepaymasterAndData field, if applicable, for the given request. The API accepts a partial UserOp and an optional token address. By examining the mode parameter in the request, it identifies the type of paymaster associated with the request and handles the processing accordingly, as described in the following.

Sponsorship Paymaster

To determine whether a request can be sponsored by the Sponsorship Paymaster, only consider the partial UserOp object and the sponsorshipInfo. If the request cannot be sponsored, thepaymasterAndData will return "0x".

Token Paymaster

Consider both the partialUserOp andtokenAddress parameters in this case. If thetokenAddress is absent, an error will be thrown. If both are present, check for theTokenPaymaster flow and send thepaymasterAndData response accordingly. If thetokenAddress is not supported, an error will also be thrown. Please see the error response at the end of this document.

Parameters

Body

Param Type Description Required method string Name of method in this case: pm_getFeeQuoteOrData Required params array An array consisting of the Useroperation object and Paymaster mode information Required id string id for request determined by client for JSON RPC requests Required jsonrpc string JSON RPC version in this case 2.0.0 Required

Params Array for Sponsorship requests

Index Type Description Required 0 object A partial userOperation object for the userOp that needs to be sponsored Required 1 object Mode specified as "SPONSORSHIP" as well as sposorship information which includes any webhook data and smart account information: name and version Required

Params Array for ERC20 paymaster requests

Index Type Description Required 0 object A partial userOperation object for the userOp that needs to be sponsored Required 1 object Mode specified as "ERC20" as well as tokenInfo: a preferred token address and list of tokens to include Required

note "MODE" is mandatory forpm_sponsorUserOperation API,

- If"MODE"
- isSPONSORED
- , we check for request sponsorship. If a request cannot be sponsored, we return 0x. This would mean users will pay for their gas fees.

 If "MODE"
- is ERC20, we returnpaymasterAndData
- for TokenPaymaster. This would mean that users will pay in their preferred ERC20 Tokens

1. Mode isSPONSORED

POST Request { "id" :
1 , "jsonrpc" :
"2.0" , "method" :
"pm_sponsorUserOperation" , "params" :
[{ sender ,
// address nonce ,
// uint256 initCode ,
// string callData ,
// string callGasLimit ,
// string verificationGasLimit ,
// string preVerificationGas ,
// string maxFeePerGas ,
// string maxPriorityFeePerGas ,
// string signature // dummy signature string } , { "mode" :
"SPONSORED", "calculateGasLimits":
true , "expiryDuration" :
300
// duration (secs) for which the generate paymasterAndData will be valid. Default duration is 300 secs. "sponsorshipInfo" :
{ "webhookData" :
<pre>{}, "smartAccountInfo" :</pre>
{ "name" :
"BICONOMY", "version":
"2.0.0"}}}]} Response Success Response
{ "jsonrpc" :
"2.0" , "id" :
1, "result":
{ "MODE" :
"SPONSORED", "paymasterAndData":
"0xdc91ffb7c4b800d70410a79a5b503ae4391f67e4000000000000000000000000000000000000
"75388" , "verificationGasLimit" :
57121 , "callGasLimit" :
108848 } }

2. Mode isERC20

POST Request: { "jsonrpc" :
"2.0" , "id" :
14 , "method" :
"pm_sponsorUserOperation" , "params" :
[{sender,
// address nonce ,
// uint256 initCode ,
// string callData ,
// string maxFeePerGas ,
// string maxPriorityFeePerGas ,
// string //optional fields callGasLimit ,
// string verificationGasLimit ,
// string preVerificationGas ,
// string } , { "calculateGasLimits" :
true , "mode" :
"ERC20" , "tokenInfo" :
{ "feeTokenAddress" :
"0xfd086bc7cd5c481dcc9c85ebe478a1c0b69fcbb9" }}]}Response: { "jsonrpc" :
"2.0" , "id" :
1 , "result" :
{ "MODE" :
"ERC20" , "paymasterAndData" :
"0xdc91ffb7c4b800d70410a79a5b503ae4391f67e4000000000000000000000000000000000000
"75388" , "verificationGasLimit" :
57121 , "callGasLimit" :
108848 } In response, the result field will containpaymasterAndData to be used in UserOpPrevious Get Fee Quotes Next Webhook API