How to send backrun bundles to MEV Blocker RPC

Bids can be submitted by any client with the relay urlhttps://rpc.mevblocker.io .

However, you need to make sure that the first transaction of the bundle you send is the hash of the target transaction.

Within the bundle itself, you need to make sure that the backrun transaction makes an ETH transfer to the fee recipient (which can either betx.origin or a custom address; learn more about how to set up a custom address here), and the value of the transfer has to be the bid amount.

To recap, a Merkle bid is a bundle with 2 transactions:

- The hash of the target transaction
- Your backrun - the backrun must pay the fee_recipient the bid with an Ether transfer

Construct a back-run bundle just like you would for a target transaction from the mempool, but make the first element of the txs array in params ofeth_sendBundle the hash of the pending target, instead of the fully-encoded transaction.

note The pending target transaction must be the first transaction in txs, and only one target transaction can be included per bundle, otherwise the request will be rejected. It is still possible to submit multiple bundles for the same block, containing different target transactions. Send the back-run bundle to the same websocket connection using theeth_sendBundle method

tip replacementUuid is supported in this version ofeth_sendBundle { "jsonrpc" :

"2.0" , "id" :

1 , "method" :

"eth_sendBundle" , "params" : [{

"txs" :

[

// this is the 32-byte transaction hash of the target transaction "0xfec1700ef24c9ff6fd2e07584a16bbb2fec1700ef24c9ff6fd2e07584a16bbb2", // this is your signed encoded transaction "0xabc123.."], "blockNumber" :

"0xb63dcd" , "minTimestamp" :

0 , "maxTimestamp" :

1615920932 }] } note Historical submitted bundles, including those that did not land on-chain, will not only be shared with builders but also archived and presented to the public for transparency. Connect to the websocket server located at searchers.mevblocker.io

websocat wss://searchers.mevblocker.io Use theeth_subscribe method to subscribe to unsigned pending transactions - mevblocker_partialPendingTransactions :

{ "method" : "eth_subscribe" , "params" :

["mevblocker_partialPendingTransactions"] } Response:

{ "jsonrpc" :

"2.0" , "id" :

1 , "result" :

"0xd58bbbc0f5190962eff01b6f0ec17724" } You'll start receiving unsigned pending transactions (missingv ,r , ands):

{ "jsonrpc" :

"2.0" , "method" :

"eth_subscription" , "params" : { "subscription" :

"0xd58bbbc0f5190962eff01b6f0ec17724" , "result" : { "chainId" :

"0x1" , "to" :

"0x6215589d293fdf52886484f46f0d6a11c76b4a7e" , "value" :

"0x4fefa17b724000" , "data" :

"0x" , "accessList" :

[ ] , "nonce" :

"0x10" , "maxPriorityFeePerGas" :

"0x0" , "maxFeePerGas" :

"0x7e1c65b04" , "gas" : " 0x5208" , "type" :

"0x2" , "hash" :

"0x5f08dd372fce1a44dda27bed60ca036acb4979fad6ca37b9c388e351a870fe4c" , "from" :

"0xcb1588f3f7e92a1278c68a6aed4bdcbc68534b29" } } }

Edit this page Previous Searchers Next How to receive backrun bundles from MEV Blocker RPC