Pre-flight / preview

Transaction Preview is an extension of our Transaction Simulation Platform for on-demand simulations of unsigned transactions against the current block height.

Overview

Transaction Preview lets you preview the outputs of any custom transactions that you submit to it. By harnessing the power of our network of Ethereum nodes, with our existing decoding and transaction lifecycle knowledge, we hope to give users of Transaction Preview even greater comfort when interacting in the Web3 space. Previewing a transaction is particularly useful for:

- Wallets:
- Help users preview net-balance changes & identify malicious contract behavior or buggy smart contracts before interacting with them.
- DEXs & Swaps:
- Report accurate slippage, Accurate tokens received, and accurate failing calls
- Traders:
- Preview many iterations of the same trade and execute only the most profitable one
- Lending Protocols:
- Tell whether a Borrow / Repay / Claim transaction will go through
- Auctions:
- Tell whether your bid or listing will go through and its accuracy & effects
- NFTs:
- Tell whether your NFT mint, purchase, or transfer will go through
-

& much more without spending a single wei in gas!

How it works

Transaction Preview comes in the form of both REST and Websocket endpoints, to which you can outline the basics of a single or multiple transactions, and you will receive a payload detailing the execution of the transactions in the latest block height. We simulate the proposed transactions by executing it on one of our dedicated simulator nodes on the current block height. The returned payload includes:

1. Any EVM execution errors
2. A list of Decoded Internal Transactions
3. A collection of all balance changes of tokens across all involved addresses
4. End to end performance metrics of the simulation
5.

Authentication

Include the headercredentials with yourapiKey:secretKey . Information on secret keys can be foundhere andhere .

Quick Start: Example single sim cURL call

cURL Request !! Make sure to enter your own api key and secret!!

This example request simulatesa single transaction : Vitalik swapping Eth for Dai on Sushiswap with a deadline of Fri Jan 15 2027

```

Copy curl --location --request POST 'https://api.blocknative.com/simulate' \ --header 'credentials::' \ --header 'Content-Type: application/json' \ --data-raw '{ "system":"ethereum", "network":"main",
"transactions":[{ "to":"0xd9e1cE17f2641f24aE83637ab66a2cca9C378B9F", "from":"0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045", "gas":10407056, "maxFeePerGas":102306635634,
"maxPriorityFeePerGas":2.14, "value":100000000000000000,
"input":"0x7ff36ab500000000000000000000000000000000000000000000000fc7fc9610eacf03cd0000000000000000000000000000000000000000000000000000000000000080000000000000000000000
} ] }'

Response [](https://docs.blocknative.com/mempool-tools/transaction-simulation/pre-flight-simulation#response)

Copy { "status": "simulated", "simulatedBlockNumber": 14674533, "transactions": [ { "from": "0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045", "to":
"0xd9e1cE17f2641f24aE83637ab66a2cca9C378B9F", "value": 100000000000000000, "gas": 10407056, "input":
"0x7ff36ab500000000000000000000000000000000000000000000000fc7fc9610eacf03cd00000000000000000000000000000000000000000000000000000000000000800000000000000000000000000d
"type": 2, "maxFeePerGas": 102306635634, "maxPriorityFeePerGas": 2.14 } ], "internalTransactions": [ [ { "type": "CALL", "from": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "to":
"0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "input": "0xd0e30db0", "gas": 10203912, "gasUsed": 23974, "value": "100000000000000000" }, { "type": "CALL", "from":
"0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "to": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "input":
"0xa9059cbb000000000000000000000000c3d03e4f041fd4cd388c549ee2a29a9e5075882f000000000000000000000000000000000000000000000000016345785d8a0000", "gas": 10178983,
"gasUsed": 8062, "value": "0", "contractCall": { "methodName": "transfer", "params": { "_to": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "_value": "100000000000000000" }, "contractAddress":
"0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "contractType": "erc20", "contractAlias": "WETH", "contractDecimals": 18, "contractName": "Wrapped Ether", "decimalValue": "0.1" } }, { "type":
"CALL", "from": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "to": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "input":
"0x022c0d9f00000000000000000000000000000000000000000000000ff0aa8d89e86502cb00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
"gas": 10168823, "gasUsed": 48406, "value": "0" }, { "type": "CALL", "from": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "to": "0x6b175474e89094c44da98b954eedeac495271d0f", "input":
"0xa9059cbb000000000000000000000000d8da6bf26964af9d7eed9e03e53415d37aa9604500000000000000000000000000000000000000000000000ff0aa8d89e86502cb", "gas": 9996662,
"gasUsed": 13074, "value": "0", "contractCall": { "methodName": "transfer", "params": { "_to": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045", "_value": "294042990044235760331" },
"contractAddress": "0x6b175474e89094c44da98b954eedeac495271d0f", "contractType": "erc20", "contractAlias": "DAI", "contractDecimals": 18, "contractName": "Dai Stablecoin", "decimalValue":
"294.042990044235760331" } } ] ], "netBalanceChanges": [ [ { "address": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "balanceChanges": [ { "delta": "-100000000000000000", "asset": { "type":
"ether", "symbol": "ETH" }, "breakdown": [ { "counterparty": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "amount": "-100000000000000000" } ] }, { "delta": "-100000000000000000", "asset": {
"type": "erc20", "symbol": "WETH", "contractAddress": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2" }, "breakdown": [ { "counterparty": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f",
"amount": "-100000000000000000" } ] } ] }, { "address": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "balanceChanges": [ { "delta": "100000000000000000", "asset": { "type": "ether", "symbol":
"ETH" }, "breakdown": [ { "counterparty": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "amount": "100000000000000000" } ] } ] }, { "address":
"0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "balanceChanges": [ { "delta": "100000000000000000", "asset": { "type": "erc20", "symbol": "WETH", "contractAddress":
"0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2" }, "breakdown": [ { "counterparty": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "amount": "100000000000000000" } ] }, { "delta": "-
294042990044235760331", "asset": { "type": "erc20", "symbol": "DAI", "contractAddress": "0x6b175474e89094c44da98b954eedeac495271d0f" }, "breakdown": [ { "counterparty":
"0xd8da6bf26964af9d7eed9e03e53415d37aa96045", "amount": "-294042990044235760331" } ] } ] }, { "address": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045", "balanceChanges": [ { "delta":
"294042990044235760331", "asset": { "type": "erc20", "symbol": "DAI", "contractAddress": "0x6b175474e89094c44da98b954eedeac495271d0f" }, "breakdown": [ { "counterparty":
"0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "amount": "294042990044235760331" } ] } ] } ] ], "error": [], "simDetails": { "blockNumber": -1, "performanceProfile": { "breakdown": [ { "label":
"detected", "timeStamp": "2022-04-28T17:37:18.993Z" }, { "label": "traceStart", "timeStamp": "2022-04-28T17:37:18.994Z" }, { "label": "traceEnd", "timeStamp": "2022-04-28T17:37:19.024Z" }, { "label":
"dispatch", "timeStamp": "2022-04-28T17:37:19.025Z" } ] }, "e2eMs": 32 }, "serverVersion": "0.133.0-0.4.8", "system": "ethereum", "network": "main", "contractCall": [ { "status": "fulfilled", "value": {
"methodName": "swapExactETHForTokens", "params": { "amountOutMin": "291111719012725097421", "deadline": "1800001651166409", "path": [ "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2",
"0x6b175474e89094c44da98b954eedeac495271d0f" ], "to": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045" }, "contractAddress": "0xd9e1cE17f2641f24aE83637ab66a2cca9C378B9F" } } ] }

```

How-to: Simulate Multiple Transactions

Endpoint Spec

POST https://api.blocknative.com/simulate

Use this endpoint to simulate multiple transactions before you submit it, allowing you to preview its effects! Include the appropriate blockchain and network, for which supported networks are listed below. Include your API key and it's associated secret key in a "credentials" header via the formatting "apiKey:secretKey". Your secret key can be set via your Accounts page where your keys are listed.

Note: You will need to either include gasPrice or maxFeePerGas and maxPriorityFeePerGas to differentiate between a type 0 and type 2 (EIP1559) transaction, for more information check out our blog post here . We recommend using type 2 transactions.

Headers

Name Type Description credentials* String Here we define an API key and secret key together in the form "apiKey:secretKey"

Request Body

Name Type Description system* String Blockchain of this transaction, currently we only support

ethereum

. network* String Network name of this transaction, currently we support

main and goerli

. transactons* Array of JSON objects This is an array of your transaction objects, information about this structure can be found

here

, however below we describe what is required. transactions[0].to*String An address from which the transaction would be sent from. transactions[0].from*String An address to where the transaction is sent to. transactions[0].value *Integer This is the amount of ETH transferred, if there is none transferred, leave this field as "value": 0. This field is in wei units. transactions[0].input*String This is the input arguments, encoded per the contract specifications.

Here

you can find some information detailing this field. transactions[0].gas* Integer Maximum amount of gas available to the transaction. transactions[0].gasPrice Integer (Type 0) Amount, in wei, per unit of gas. transactions[0].maxFeePerGas Integer (Type 2) Max fee per gas in wei. transactions[0].maxPriorityFeePerGas Integer (Type 2) Max priority fee per gas in wei. 200: OK A successful simulation response will have the following response fields: 400: Bad Request This response typically means the body payload was malformed in some way. An error message detailing specifically what is returned. ```

Copy { "status":"simulation", "simulatedBlockNumber":, "from": "", "transactions": [{}] // list of original transaction objects "error": [""], // optional, if the EVM execution encountered an error while tracing "internalTransactions": [[]], // A list of lists of internal transactions from the EVM execution ordered by transaction "netBalanceChanges": [[]], // A list of lists of addresses with their balance changes, ETH and erc-20 tokens ordered by transaction "simDetails": {}, // Some latency metrics "serverVersion": "0", "system": "ethereum", "network": "main", "contractCall": [{}], // a list of the top level transaction contract call for each decoded }

Copy { // Response error message detailing the body payload error }

```

Example: Multiple unsigned simulations cURL call

cURL Request This payload will simulate Vitalik approving Sushiswap to transfer Balancer tokens, and then swaps Balancer for DAI on Sushiswap.

```

Copy curl --location --request POST 'https://api.blocknative.com/simulate' \ --header 'credentials::' \ --header 'Content-Type: application/json' \ --data-raw '{ "system":"ethereum", "network":"main", "transactions":[{ "to":"0xba100000625a3754423978a60c9317c58a424e3D", "from":"0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045", "gas":12407056, "maxFeePerGas":82306635634, "maxPriorityFeePerGas":2.14, "value":0, "input":"0x095ea7b3000000000000000000000000d9e1cE17f2641f24aE83637ab66a2cca9C378B9F000000000000000000000000000000000785ee10d5da46d900f436a000000000" }, { "to":"0xd9e1cE17f2641f24aE83637ab66a2cca9C378B9F", "from":"0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045", "gas":10407056, "maxFeePerGas":102306635634, "maxPriorityFeePerGas":2.14, "value":0, "input":"0x38ed1739000000000000000000000000000000000000000000000000016345785d8a0000000000000000000000000000000000000000000000000000000014257090bdd86ae20000000000000000 }] }'

Response [](https://docs.blocknative.com/mempool-tools/transaction-simulation/pre-flight-simulation#response-1)

Copy { "status": "simulated", "simulatedBlockNumber": 14674533, "transactions": [{ "from": "0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045", "to": "0xba100000625a3754423978a60c9317c58a424e3D", "value": 0, "gas": 12407056, "input": "0x095ea7b3000000000000000000000000d9e1cE17f2641f24aE83637ab66a2cca9C378B9F000000000000000000000000000000000785ee10d5da46d900f436a000000000", "type": 2, "maxFeePerGas": 82306635634, "maxPriorityFeePerGas": 2.14 }, { "from": "0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045", "to": "0xd9e1cE17f2641f24aE83637ab66a2cca9C378B9F", "value": 0, "gas": 10407056, "input": "0x38ed1739000000000000000000000000000000000000000000000000016345785d8a0000000000000000000000000000000000000000000000000000000014257090bdd86ae200000000000000000000000 "type": 2, "maxFeePerGas": 102306635634, "maxPriorityFeePerGas": 2.14 }], "internalTransactions": [[], [{ "type": "CALL", "from": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "to": "0xba100000625a3754423978a60c9317c58a424e3d", "input": "0x23b872dd000000000000000000000000d8da6bf26964af9d7eed9e03e53415d37aa960450000000000000000000000009bffa3ce3e56d0d26447a45771fec76bd4173022000000000000000000000000 "gas": 10201236, "gasUsed": 25509, "value": "0", "contractCall": { "methodName": "transferFrom", "params": { "amount": "100000000000000000", "recipient": "0x9bffa3ce3e56d0d26447a45771fec76bd4173022", "sender": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045" }, "contractAddress": "0xba100000625a3754423978a60c9317c58a424e3d", "contractType": "Balancer: BAL Token" } }, { "type": "CALL", "from": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "to": "0x9bffa3ce3e56d0d26447a45771fec76bd4173022", "input": "0x022c0d9f00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001c0c6cf1cacf1000000000000000000000000 "gas": 10172992, "gasUsed": 48134, "value": "0" }, { "type": "CALL", "from": "0x9bffa3ce3e56d0d26447a45771fec76bd4173022", "to": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "input": "0xa9059cbb000000000000000000000000c3d03e4f041fd4cd388c549ee2a29a9e5075882f0000000000000000000000000000000000000000000000000001c0c6cf1cacf1", "gas": 10000736, "gasUsed": 12862, "value": "0", "contractCall": { "methodName": "transfer", "params": { "_to": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "_value": "493435087531249" }, "contractAddress": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "contractType": "erc20", "contractAlias": "WETH", "contractDecimals": 18, "contractName": "Wrapped Ether", "decimalValue": "0.000493435087531249" } }, { "type": "CALL", "from": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f", "to": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "input": "0x022c0d9f000000000000000000000000000000000000000000000000143f3a1695d5a8dc0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 "gas": 10123578, "gasUsed": 48406, "value": "0" }, { "type": "CALL", "from": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "to": "0x6b175474e89094c44da98b954eedeac495271d0f", "input": "0xa9059cbb000000000000000000000000d8da6bf26964af9d7eed9e03e53415d37aa96045000000000000000000000000000000000000000000000000143f3a1695d5a8dc", "gas": 9952124, "gasUsed": 13074, "value": "0", "contractCall": { "methodName": "transfer", "params": { "_to": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045", "_value": "1458948672968829148" }, "contractAddress": "0x6b175474e89094c44da98b954eedeac495271d0f", "contractType": "erc20", "contractAlias": "DAI", "contractDecimals": 18, "contractName": "Dai Stablecoin", "decimalValue": "1.458948672968829148" } }]], "netBalanceChanges": [[], [{ "address": "0x9bffa3ce3e56d0d26447a45771fec76bd4173022", "balanceChanges": [{ "delta": "-493435087531249", "asset": { "type": "erc20", "symbol": "WETH", "contractAddress": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2" }, "breakdown": [{ "counterparty": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "amount": "-493435087531249" }] }] }, { "address": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "balanceChanges": [{ "delta": "493435087531249", "asset": { "type": "erc20", "symbol": "WETH", "contractAddress": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2" }, "breakdown": [{ "counterparty": "0x9bffa3ce3e56d0d26447a45771fec76bd4173022", "amount": "493435087531249" }] }, { "delta": "-1458948672968829148", "asset": { "type": "erc20", "symbol": "DAI", "contractAddress": "0x6b175474e89094c44da98b954eedeac495271d0f" }, "breakdown": [{ "counterparty": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045", "amount": "-1458948672968829148" }] }] }, { "address": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045", "balanceChanges": [{ "delta": "1458948672968829148", "asset": { "type": "erc20", "symbol": "DAI", "contractAddress": "0x6b175474e89094c44da98b954eedeac495271d0f" }, "breakdown": [{ "counterparty": "0xc3d03e4f041fd4cd388c549ee2a29a9e5075882f", "amount": "1458948672968829148" }] }] }]], "error": [], "simDetails": { "blockNumber": -1, "performanceProfile": { "breakdown": [{ "label": "detected", "timeStamp": "2022-04-28T18:18:42.893Z" }, { "label": "traceStart", "timeStamp": "2022-04-28T18:18:42.896Z" }, { "label": "traceEnd", "timeStamp": "2022-04-28T18:18:42.929Z" }, { "label": "dispatch", "timeStamp": "2022-04-28T18:18:42.929Z" }] }, "e2eMs": 36 }, "serverVersion": "0.133.0-0.4.8", "system": "ethereum", "network": "main", "contractCall": [{ "status": "fulfilled", "value": { "methodName": "approve", "params": { "amount": "1000000000000000000000000000000000000", "spender": "0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f" }, "contractAddress": "0xba100000625a3754423978a60c9317c58a424e3D", "contractType": "erc20", "contractAlias": "BAL", "contractDecimals": 18, "contractName": "Balancer" } }, { "status": "fulfilled", "value": { "methodName": "swapExactTokensForTokens", "params": { "amountIn": "100000000000000000", "amountOutMin": "1451690221859531490", "deadline": "1800001651169842", "path": ["0xba100000625a3754423978a60c9317c58a424e3d", "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2", "0x6b175474e89094c44da98b954eedeac495271d0f"], "to": "0xd8da6bf26964af9d7eed9e03e53415d37aa96045" }, "contractAddress": "0xd9e1cE17f2641f24aE83637ab66a2cca9C378B9F" } }] }

```

How-to: Websocket Endpoint

wss://api.blocknative.com/v0

Example websocket call

```

Copy ws.send(JSON.stringify({ "timeStamp": new Date().toISOString(), "dappId":, "version": "1", "blockchain": { "system": "ethereum", "network": "main" }, "categoryCode": "simulate", "eventCode": "txSimulation", "transaction": { "from": "0xAb5801a7D398351b8bE11C439e05C5B3259aeC9B", "to": "0x7a250d5630b4cf539739df2c5dacb4c659f2488d", "gas": 332160, "gasPrice": 0, "input": "0xfb3bdb410000000000000000000000000000000000000000000000003635c9adc5dea00000000000000000000000000000000000000000000000000000000000000000008000000000000000000000000 "value": 28981678599856503 } }))

```

JSON Response Fields

Name Type Description internalTransactions [JSON object] A collection of internal transactions made by the activated contract. Supported calls are decoded . If you wish for your contract to be decoded, contact us on Discord . netBalanceChanges [JSON object] Contains details of all the balance changes resulting from the internal transactions details in theinternalTransactions array. The object is an array with an entry for each address involved in the internal transactions. simDetails JSON object This object contains details about the simulation, such as blockHeight at the time of calculation, key action timestamps, and end-to-end latency value, in milliseconds. contractCall [JSON object] This is the top level contract call decoded. If we do not support decoding for this contract, this field will be null. gasUsed Number Gas used during transaction execution.

Supported Networks

Transaction Preview currently supportsEthereum Mainnet & Goerli Testnet . Please stay tuned for its availability on other networks supported by Blocknative.
```

Rate Limits

Usage rate limit for the Transaction Preview API is 100 requests per 60-second window. Transaction Preview consumes the same daily limits on your Blocknative plan as pending simulated transactions in Simulation Platform. You can view your current consumption on yourBlocknative Account Page .

Previous Transaction Simulation Next In-flight / pending Last updated1 year ago

Was this helpful?