

Contract

?module=contract

Page is under construction. For a full description of RPC endpoints, visit <https://gnosis.blockscout.com/api-docs>

https://instance_base_url/api?module=contract

Get a list of contracts

listcontracts

List sorted in ascending order based on the time a contract was first indexed by the explorer. With filters `not_decompiled(4)` or `not_verified(4)` the results will not be sorted for performance reasons.

Example:

...

Copy [https://instance_base_url/api?module=contract &action=listcontracts](https://instance_base_url/api?module=contract&action=listcontracts)

...

Request Params Example Result Parameter Description page optional nonnegative integer representing the page number used for pagination. 'offset' must also be provided. offset optional nonnegative integer representing the max number of records to return when paginating. 'page' must also be provided. filter optional string `verified` | `decompiled` | `unverified` | `not_decompiled` | `empty`, or 1 | 2 | 3 | 4 | 5 respectively. Returns contracts with the requested status. `not_decompiled_with_version` optional string ensures none of the returned contracts were decompiled with the provided version. Ignored unless filtering for decompiled contracts. `verified_at_start_timestamp` optional unix timestamp Represents the starting timestamp for verified contracts. Only used with `verified` filter. `verified_at_end_timestamp` optional unix timestamp Represents the ending timestamp for verified contracts. Only used with `verified` filter. ``

Copy { "message": "OK", "result": [{ "ABI": "[{\n\"type\": \"event\", \n\"inputs\": [{\n\"name\": \"a\", \n\"type\": \"uint256\", \n\"indexed\": true}, {\n\"name\": \"b\", \n\"type\": \"bytes32\", \n\"indexed\": false}], \n\"name\": \"Event\", \n}, {\n\"type\": \"event\", \n\"inputs\": [{\n\"name\": \"a\", \n\"type\": \"uint256\", \n\"indexed\": true}, {\n\"name\": \"b\", \n\"type\": \"bytes32\", \n\"indexed\": false}], \n\"name\": \"Event2\", \n}, {\n\"type\": \"function\", \n\"inputs\": [{\n\"name\": \"a\", \n\"type\": \"uint256\"}], \n\"name\": \"foo\", \n\"outputs\": []\n}]\n", "CompilerVersion": "v0.2.1-2016-01-30-91a6b35", "ContractName": "Test", "OptimizationUsed": "1", "SourceCode": "pragma solidity >0.4.24;\n\ncontract Test {\nconstructor()\npublic { b = hex\"12345678901234567890123456789012\"; }\n\nevent Event(uint indexed a, bytes32 b);\n\nevent Event2(uint indexed a, bytes32 b);\n\nfunction foo(uint a) public { emit Event(a, b); }\n\nbytes32 b;\n}\n", "status": "1" }

...

Get ABI for a verified contract

getabi

Also available through a GraphQL `addresses` query.

Example:

...

Copy [https://instance_base_url/api?module=contract &action=getabi &address={addressHash}](https://instance_base_url/api?module=contract&action=getabi&address={addressHash})

...

Request Params Example Result Parameter Description address string containing the address hash. ``

Copy { "message": "OK", "result": "[{\n\"constant\": false, \n\"inputs\": [{\n\"name\": \"voucher_token\", \n\"type\": \"bytes32\"}], \n\"name\": \"burn\", \n\"outputs\": [{\n\"name\": \"success\", \n\"type\": \"bool\"}], \n\"payable\": false, \n\"stateMutability\": \"nonpayable\", \n\"type\": \"function\", \n}, {\n\"constant\": true, \n\"inputs\": [{\n\"name\": \"voucher_token\", \n\"type\": \"bytes32\"}], \n\"name\": \"is_expired\", \n\"outputs\": [{\n\"name\": \"\", \n\"type\": \"bool\"}], \n\"payable\": false, \n\"stateMutability\": \"view\", \n\"type\": \"function\", \n}, {\n\"constant\": false, \n\"inputs\": [{\n\"name\": \"voucher_token\", \n\"type\": \"bytes32\"}], \n\"name\": \"is_burnt\", \n\"outputs\": [{\n\"name\": \"\", \n\"type\": \"bool\"}], \n\"payable\": false, \n\"stateMutability\": \"nonpayable\", \n\"type\": \"function\", \n}, {\n\"inputs\": [{\n\"name\": \"voucher_token\", \n\"type\": \"bytes32\", \n}, {\n\"name\": \"_lifetime\", \n\"type\": \"uint256\"}], \n\"payable\": false, \n\"stateMutability\": \"nonpayable\", \n\"type\": \"constructor\", \n}], \n\"status\": \"1" }

...

Get contract source code for a verified contract

getsourcecode

Also available through a GraphQLAddresses query.

Example:

...

Copy https://instance_base_url/api?module=contract&action=getsourcecode&address={addressHash}

...

Request Params Example Result Parameter Description address string containing the address hash. ``

```
Copy { "message": "OK", "result": { "ABI": "{\n  \"type\": \"event\", \"inputs\": [\n    {\n      \"name\": \"a\", \"type\": \"uint256\", \"indexed\": true,\n      \"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false\n    },\n    {\n      \"name\": \"Event\", \"type\": \"event\", \"inputs\": [\n        {\n          \"name\": \"a\", \"type\": \"uint256\", \"indexed\": true,\n          \"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false\n        },\n        {\n          \"name\": \"Event2\", \"type\": \"function\", \"inputs\": [\n            {\n              \"name\": \"a\", \"type\": \"uint256\", \"indexed\": true\n            },\n            {\n              \"name\": \"foo\", \"type\": \"uint256\", \"indexed\": false\n            }\n          ]\n        },\n        {\n          \"name\": \"CompilerVersion\", \"type\": \"string\", \"indexed\": false\n        },\n        {\n          \"name\": \"ContractName\", \"type\": \"string\", \"indexed\": false\n        },\n        {\n          \"name\": \"FileName\", \"type\": \"string\", \"indexed\": false\n        },\n        {\n          \"name\": \"ImplementationAddress\", \"type\": \"string\", \"indexed\": false\n        },\n        {\n          \"name\": \"IsProxy\", \"type\": \"bool\", \"indexed\": false\n        },\n        {\n          \"name\": \"OptimizationUsed\", \"type\": \"bool\", \"indexed\": false\n        },\n        {\n          \"name\": \"SourceCode\", \"type\": \"string\", \"indexed\": false\n        }\n      ]\n    },\n    {\n      \"name\": \"Event\", \"type\": \"event\", \"inputs\": [\n        {\n          \"name\": \"a\", \"type\": \"uint256\", \"indexed\": true\n        },\n        {\n          \"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false\n        }\n      ]\n    },\n    {\n      \"name\": \"Event2\", \"type\": \"event\", \"inputs\": [\n        {\n          \"name\": \"a\", \"type\": \"uint256\", \"indexed\": true\n        },\n        {\n          \"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false\n        }\n      ]\n    },\n    {\n      \"name\": \"foo\", \"type\": \"function\", \"inputs\": [\n        {\n          \"name\": \"a\", \"type\": \"uint256\", \"indexed\": true\n        }\n      ]\n    },\n    {\n      \"name\": \"CompilerVersion\", \"type\": \"string\", \"indexed\": false\n    },\n    {\n      \"name\": \"ContractName\", \"type\": \"string\", \"indexed\": false\n    },\n    {\n      \"name\": \"FileName\", \"type\": \"string\", \"indexed\": false\n    },\n    {\n      \"name\": \"ImplementationAddress\", \"type\": \"string\", \"indexed\": false\n    },\n    {\n      \"name\": \"IsProxy\", \"type\": \"bool\", \"indexed\": false\n    },\n    {\n      \"name\": \"OptimizationUsed\", \"type\": \"bool\", \"indexed\": false\n    },\n    {\n      \"name\": \"SourceCode\", \"type\": \"string\", \"indexed\": false\n    }\n  ]\n}, \"status\": \"1\" }
```

...

Get contract creator address hash and creation transaction hash

getcontractcreation

Returns contract creator and transaction hash. Up to 10 contracts at the one request

Example:

...

Copy https://instance_base_url/api?module=contract&action=getcontractcreation&contractaddresses={addressHash},{addressHash}

...

Request Params Example Result Parameter Description contractaddresses string containing address hashes, separated by,

...

```
Copy { "message": "OK", "result": [ { "contractAddress": "0xdc2082945d55596bf39f362d9ec0f7f65edbb9dd", "contractCreator": "0xbb36c792b9b45aaf8b848a1392b0d6559202729e", "txHash": "0xe79426c9a5560cfbd78d0ef058e455c94483224ec265475583ff01ebdedceaf2" }, { "contractAddress": "0x1b02da8cb0d097eb8d57a175b88c7d8b47997506", "contractCreator": "0xf87bc5535602077d340806d71f805ea9907a843d", "txHash": "0xe54bf38dfad760b8f96a76d8136d6540d74b02dafb5781746bf705bf7e647a3b" } ], "status": "1" }
```

...

Verify a contract with its source code and contract creation information

verify

Example:

...

Copy https://instance_base_url/api?module=contract&action=verify&addressHash={addressHash}&name={name}&compilerVersion={compilerVersion}&optimization={false}&contractSourceCode={contractSourceCode}

...

Curl Post Example

...

Copy curl -d

```
'{"addressHash":"0xc63BB6555C90846afACaC08A0F0Aa5caFCB382a1","compilerVersion":"v0.5.4+commit.9549d8ff",
"contractSourceCode":"pragma solidity ^0.5.4; contract Test { }","name":"Test","optimization":false}' -H "Content-Type:
application/json" -X POST "https://blockscout.com/poa/sokol/api?module=contract&action=verify"
```

...

On successful submission you will receive a guid as a receipt. Use this with [checkverifystatus](#) to view verification status.

Params Example

Result	Parameter	Description
addressHash	string	containing the address hash of the contract.
name	string	containing the name of the contract.
compilerVersion	string	containing the compiler version for the contract.
optimization	enum	whether or not compiler optimizations were enabled 0 =false, 1 =true
contractSourceCode	string	containing the source code of the contract.
constructorArguments	optional string	constructor argument data provided.
autodetectConstructorArguments	optional boolean	whether or not automatically detect constructor argument.
evmVersion	optional	EVM version for the contract.
optimizationRuns	optional	number of optimization runs used during compilation
library1Name	optional string	name of the first library used.
library1Address	optional string	address of the first library used.
library2Name	optional string	name of the second library used.
library2Address	optional string	address of the second library used.
library3Name	optional string	name of the third library used.
library3Address	optional string	address of the third library used.
library4Name	optional string	name of the fourth library used.
library4Address	optional string	address of the fourth library used.
library5Name	optional string	name of the fifth library used.
library5Address	optional string	address of the fifth library used.

...

Copy { "message": "OK", "result": { "ABI": "{\n\"type\": \"event\", \n\"inputs\":

```
[{"name": \"a\", \"type\": \"uint256\", \"indexed\": true},
{ \"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false}], \n\"name\": \"Event\", \n}, { \"type\": \"event\", \n\"inputs\":
[{"name\": \"a\", \"type\": \"uint256\", \"indexed\": true},
{ \"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false}], \n\"name\": \"Event2\", \n}, { \"type\": \"function\", \n\"inputs\":
[{"name\": \"a\", \"type\": \"uint256\"}], \n\"name\": \"foo\", \n\"outputs\": [ ]\n} ]\n\", \"CompilerVersion\": \"v0.2.1-2016-01-30-
91a6b35\", \"ContractName\": \"Test\", \"ImplementationAddress\": \"0x0000000000000000000000000000000000000000e\",
\"IsProxy\": \"true\", \"OptimizationUsed\": \"1\", \"SourceCode\": \"pragma solidity >0.4.24;\n\ncontract Test {\n\nconstructor() public {
b = hex\"12345678901234567890123456789012\"; }\n\nevent Event(uint indexed a, bytes32 b);\n\nevent Event2(uint indexed a,
bytes32 b);\n\nfunction foo(uint a) public { emit Event(a, b); }\n\nbytes32 b;\n}\n\", \"status\": \"1\" }
```

...

Verify a contract through [Sourcify](#)

verify_via_sourcify

1. if a smart contract is already verified on Sourcify, it will automatically fetch the data from the [repo](#)
2. otherwise you need to upload source files and JSON metadata file(s).
- 3.

Example:

...

Copy `https://instance_base_url/api ?module=contract &action=verify_via_sourcify &addressHash={addressHash}`

...

POST body example

...

Copy `--6e1e4c11657c62dc1e4349d024de9e28 Content-Disposition: form-data; name="addressHash"`

`0xb77b7443e0f32f1fEBf0BE0fBd7124D135d0a525`

`--6e1e4c11657c62dc1e4349d024de9e28 Content-Disposition: form-data; name="files[0]"; filename="contract.sol" Content-Type: application/json`

...Source code...

`--6e1e4c11657c62dc1e4349d024de9e28 Content-Disposition: form-data; name="files[1]"; filename="metadata.json" Content-Type: application/json`

...JSON metadata...

--6e1e4c11657c62dc1e4349d024de9e28--

...

Params Example Parameter Description addressHash string containing the address hash. files array with sources and metadata files ```

```
Copy { "message": "OK", "result": { "ABI": "[{\n\"type\": \"event\", \n\"inputs\": [\n\"name\": \"a\", \"type\": \"uint256\", \"indexed\": true}, {\n\"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false}], \n\"name\": \"Event\", \n}, {\n\"type\": \"event\", \n\"inputs\": [\n\"name\": \"a\", \"type\": \"uint256\", \"indexed\": true}, {\n\"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false}], \n\"name\": \"Event2\", \n}, {\n\"type\": \"function\", \n\"inputs\": [\n\"name\": \"a\", \"type\": \"uint256\"], \n\"name\": \"foo\", \n\"outputs\": []\n}]\n", "CompilerVersion": "v0.2.1-2016-01-30-91a6b35", "ContractName": "Test", "ImplementationAddress": "0x0000000000000000000000000000000000000000e", "IsProxy": "true", "OptimizationUsed": "1", "SourceCode": "pragma solidity >0.4.24;\n\ncontract Test {\n  constructor() public {\n    b = hex\"12345678901234567890123456789012\";\n  }\n  event Event(uint indexed a, bytes32 b);\n  event Event2(uint indexed a, bytes32 b);\n  function foo(uint a) public {\n    emit Event(a, b);\n    bytes32 b;\n  }\n}, "status": "1" }
```

...

Verify a vyper contract with its source code and contract creation information

verify_vyper_contract

Example

...

Copy https://instance_base_url/api?module=contract&action=verify_vyper_contract&addressHash={addressHash}&name={name}&compilerVersion={compilerVersion}&contractSourceCode={contractSourceCode}

...

curl POST example

...

Copy `curl --location --request POST 'http://localhost:4000/api?module=contract&action=verify_vyper_contract' --form 'contractSourceCode="SOURCE_CODE"' --form 'name="Vyper_contract"' --form 'addressHash="0xE60B1B8bD493569a3E945be50A6c89d29a560Fa1"' --form 'compilerVersion="v0.2.12"'`

...

First Tab Example Parameter Description addressHash string containing the address hash of the contract. name string containing the name of the contract. compilerVersion string containing the compiler version for the contract. contractSourceCode string containing the source code of the contract. constructorArguments string constructor argument data provided. ```

```
Copy { "message": "OK", "result": { "ABI": "[{\n\"type\": \"event\", \n\"inputs\": [\n\"name\": \"a\", \"type\": \"uint256\", \"indexed\": true}, {\n\"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false}], \n\"name\": \"Event\", \n}, {\n\"type\": \"event\", \n\"inputs\": [\n\"name\": \"a\", \"type\": \"uint256\", \"indexed\": true}, {\n\"name\": \"b\", \"type\": \"bytes32\", \"indexed\": false}], \n\"name\": \"Event2\", \n}, {\n\"type\": \"function\", \n\"inputs\": [\n\"name\": \"a\", \"type\": \"uint256\"], \n\"name\": \"foo\", \n\"outputs\": []\n}]\n", "CompilerVersion": "v0.2.1-2016-01-30-91a6b35", "ContractName": "Test", "ImplementationAddress": "0x0000000000000000000000000000000000000000e", "IsProxy": "true", "OptimizationUsed": "1", "SourceCode": "pragma solidity >0.4.24;\n\ncontract Test {\n  constructor() public {\n    b = hex\"12345678901234567890123456789012\";\n  }\n  event Event(uint indexed a, bytes32 b);\n  event Event2(uint indexed a, bytes32 b);\n  function foo(uint a) public {\n    emit Event(a, b);\n    bytes32 b;\n  }\n}, "status": "1" }
```

...

Verify a contract with Standard input JSON file

verifysourcecode

Example

...

Copy https://instance_base_url/api?module=contract&action=verifysourcecode&codeformat={solidity-standard-json-input}&contractaddress={contractaddress}&contractname={contractname}&compilerversion={compilerversion}&sourceCode={sourceCode}

...

solidity-single-file :

...

Copy curl --location 'localhost:4000/api?module=contract&action=verifysourcecode' \ --form 'contractaddress="0xDc2082945d55596bf39F362d9EC0F7F65eDBB9DD"' \ --form 'sourceCode="// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

contract Storage { uint256 number;

function store(uint256 num) public { number = num; }

function retrieve() public view returns (uint256){ return number; } }" \ --form 'contractname="Storage"' \ --form 'codeformat="solidity-single-file"' \ --form 'compilerversion="v0.8.17+commit.8df45f5"' \ --form 'optimizationUsed="1"' \ --form 'runs="199"' \ --form 'constructorArguments=""' \ --form 'evmversion="london"' \ --form 'libraryname1="qwe"' \ --form 'libraryaddress1="0xDc2082945d55596bf39F362d9EC0F7F65eDBB9DD"' --form 'licenseType=0'

...

Params Example Result Parameter Description codeformat Format of sourceCode (solidity-standard-json-input or solidity-single-file) contractaddress string containing the address hash of the contract. contractname string name of the contract. It can be an empty string(""), just the contract name("ContractName"), or a filename and contract name("contracts/contract_1.sol:ContractName") compilerversion string containing the compiler version for the contract. sourceCode string standard input json or flattened solidity code optimizationUsed boolean could be 0 ,false ,1 ,true . Should be set when codeformat=solidity-single-file runs integer is equal to optimization runs number set on compilation. Should be set when optimizationUsed is 1 or true evmversion string EVM version. Should be set when codeformat=solidity-single-file constructorArguments optional string constructor argument data provided. autodetectConstructorArguments optional boolean whether or not automatically detect constructor argument. licenseType string or number representing the license type.

License type

"See [available license types](#) ." "

Copy { "message": "OK", "result": "b080b96bd06ad1c9341c2afb7e3730311388544961acde94", "status": "1" }

...

Return status of a verification attempt

checkverifystatus

guid is received as a receipt from the verifysourcecode method. Example

...

Copy https://instance_base_url/api ?module=contract &action=checkverifystatus &guid={identifierString}

...

Params Example Result Parameter Description guid string used for identifying verification attempt "

Copy { "message": "OK", "result": "Pending in queue", "status": "1" }

...

Return Options: Pending in queue | Pass - Verified | Fail - Unable to verify | Unknown UID

Verify proxy contract

verifyproxycontract

Example

...

Copy https://instance_base_url/api ?module=contract &action=verifyproxycontract &address={addressHash}

...

Params Example Result Parameter Description address string containing the address hash of the contract ``

Copy { "message": "OK", "result": "c32d204404f33ff38fee42394f7e671fd96314b3658d466a", "status": "1" }

...

Return status of a proxy contract verification attempt

checkproxyverification

guid is received as a receipt from theverifyproxycontract method. Example

...

Copy https://instance_base_url/api ?module=contract &action=checkproxyverification &guid={identifierString}

...

Params Example Result Parameter Description guid string used for identifying verification attempt ``

Copy { "message": "OK", "result": "Implementation (0x5a3f40fd57731bbf62a38fd290add074e9cdb844) was verified and saved for proxy (0xc32d204404f33ff38fee42394f7e671fd96314b3)", "status": "1" }

...

Return Options:

- Verification in progress
- The proxy's ({addressHash}) implementation contract is found at {addressHash} and is successfully updated.
- A corresponding implementation contract was unfortunately not detected for the proxy address.
- Unknown UID *

Last updated1 month ago