


# API

You can use the Hub GraphQL API to create flexible queries for the data you need to integrate with Snapshot.

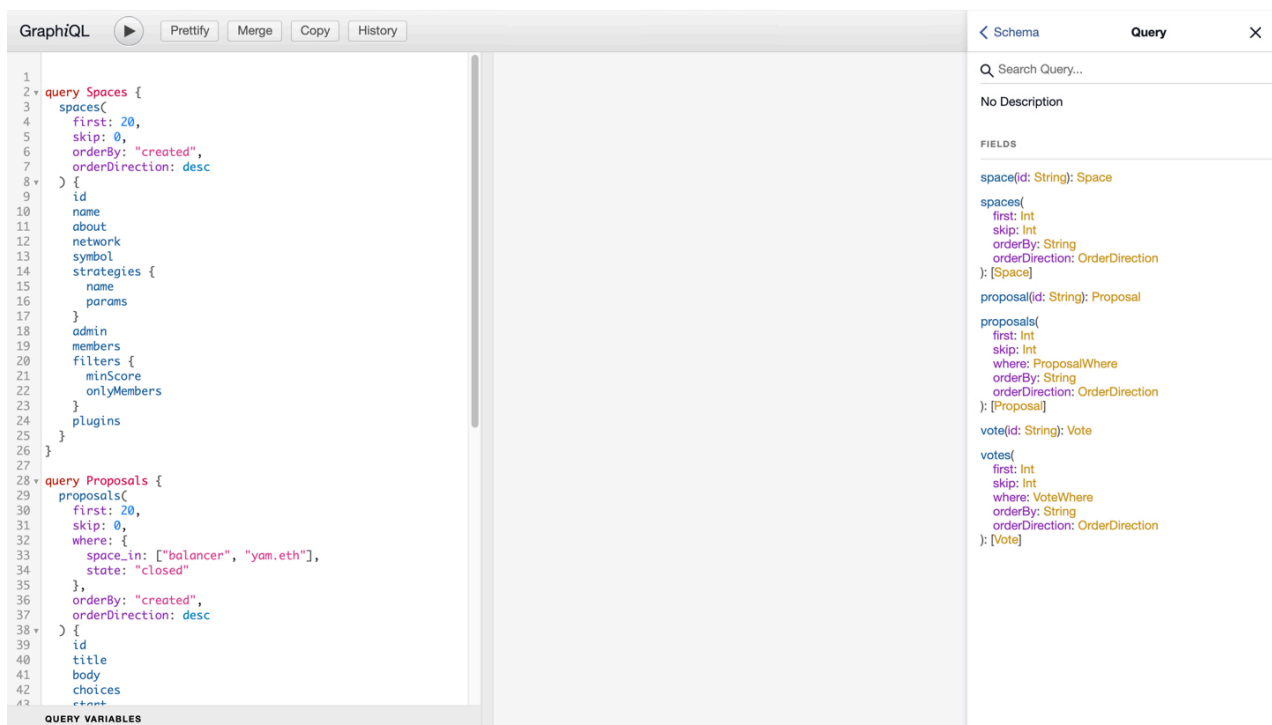
 There is a limit of 60 requests per minute with the API, to get higher limits please apply for an API Key by following this guide: [API Keys](#)

## Hub GraphQL API - Explorer

You can run queries on Snapshot data using a GraphQL Explorer.

We have exposed an integrated development environment in the browser that includes docs, syntax highlighting, and validation errors. Click the link below to access the interface.

<https://hub.snapshot.org/graphql>  
hub.snapshot.org



# Endpoints

## Production hub

```
https://hub.snapshot.org/graphql
```

## Demo hub

```
https://testnet.hub.snapshot.org/graphql
```

# Queries

## Get a single space

### Arguments

id `string`

### Example

#### Request

```
query {  
  space(id: "yam.eth") {  
    id  
    name  
    about  
    network  
    symbol  
    members  
  }  
}
```

## Response

```
{
  "data": {
    "space": {
      "id": "yam.eth",
      "name": "Yam Finance",
      "about": "",
      "network": "1",
      "symbol": "YAM",
      "members": [
        "0x683A78bA1f6b25E29fbBC9Cd1BFA29A51520De84",
        "0x9Ebc8AD4011C7f559743Eb25705CCF5A9B58D0bc",
        "0xC3edCBe0F93a6258c3933e86fFaA3bcF12F8D695",
        "0xbdac5657eDd13F47C3DD924eAa36Cf1Ec49672cc",
        "0xEC3281124d4c2FCA8A88e3076C1E7749CfEcb7F2"
      ]
    }
  }
}
```

Try on [GraphiQL](#)

## Get multiple spaces

### Arguments

first

skip

where:

- id

- id\_in

orderBy

orderDirection  or

### Example

## Request

```
query {  
  spaces(  
    first: 20,  
    skip: 0,  
    orderBy: "created",  
    orderDirection: asc  
  ) {  
    id  
    name  
    about  
    network  
    symbol  
    strategies {  
      name  
      params  
    }  
    admins  
    members  
    filters {  
      minScore  
      onlyMembers  
    }  
    plugins  
  }  
}
```

## Response

```
{
  "data": {
    "spaces": [
      {
        "id": "bonustrack.eth",
        "name": "Fabien",
        "about": "",
        "network": "1",
        "symbol": "TICKET",
        "strategies": [
          {
            "name": "erc20-balance-of",
            "params": {
              "symbol": "DAI",
              "address": "0x6B175474E89094C44Da98b954EedeAC495271d0F",
              "decimals": 18
            }
          }
        ],
        "admins": [],
        "members": [
          "0x24A12Fa313F57aF541d447c594072A992c605DCf"
        ],
        "filters": {
          "minScore": 0,
          "onlyMembers": false
        },
        "plugins": {
          "quorum": {
            "total": 500,
            "strategy": "static"
          }
        }
      }
    ]
  }
}
```

Try on [GraphiQL](#)

## Get a single proposal

## Arguments

id `string`

## Example

### Request

```
query {  
  proposal(id: "QmWbpCtwdLzxuLKnMW4Vv4MPFd2pdPX71YBKPasfZxqLUS") {  
    id  
    title  
    body  
    choices  
    start  
    end  
    snapshot  
    state  
    author  
    created  
    scores  
    scores_by_strategy  
    scores_total  
    scores_updated  
    plugins  
    network  
    strategies {  
      name  
      network  
      params  
    }  
    space {  
      id  
      name  
    }  
  }  
}
```

### Response

```
{
  "data": {
    "proposal": {
      "id": "QmWbpCtwdLzXuLKnMW4Vv4MPFd2pdPX71YBKPasfZxqLUS",
      "title": "Select Initial Umbrella Metapool",
      "body": "Eventually, we hope that anyone will be able to create",
      "choices": [
        "Option 1: BlueChips MetaPool",
        "Option 2: Hot New Projects MetaP",
        "Option 3: Integrated DegenV2 Met"
      ],
      "start": 1620676800,
      "end": 1620806400,
      "snapshot": "12408670",
      "state": "closed",
      "author": "0xEC3281124d4c2FCA8A88e3076C1E7749CfEcb7F2",
      "space": {
        "id": "yam.eth",
        "name": "Yam Finance"
      }
    }
  }
}
```

Try on [GraphiQL](#)

## Get proposals

### Arguments

first

skip

where:

- id

- id\_in

- space:

- space\_in:

- author:

- author\_in:

- network: `string`  
- network\_in: `array`  
- state: `array`  
orderBy `string`  
orderDirection `asc` or `desc`

## Example

### Request

```
query {  
  proposals (  
    first: 20,  
    skip: 0,  
    where: {  
      space_in: ["yam.eth"],  
      state: "closed"  
    },  
    orderBy: "created",  
    orderDirection: desc  
  ) {  
    id  
    title  
    body  
    choices  
    start  
    end  
    snapshot  
    state  
    scores  
    scores_by_strategy  
    scores_total  
    scores_updated  
    author  
    space {  
      id  
      name  
    }  
  }  
}
```

### Response



```

{
  "data": {
    "proposals": [
      {
        "id": "QmWbpCtwdLzXuLKnMW4Vv4MPFd2pdPX71YBKPasfZxqLUS",
        "title": "Select Initial Umbrella Metapool",
        "body": "Eventually, we hope that anyone will be able to crea",
        "choices": [
          "Option 1: BlueChips MetaPool",
          "Option 2: Hot New Projects MetaP",
          "Option 3: Integrated DegenV2 Met"
        ],
        "start": 1620676800,
        "end": 1620806400,
        "snapshot": "12408670",
        "state": "closed",
        "author": "0xEC3281124d4c2FCA8A88e3076C1E7749CfEcb7F2",
        "space": {
          "id": "yam.eth",
          "name": "Yam Finance"
        }
      },
      ...
    ]
  }
}

```

Try on [GraphiQL](#)

## Get a single vote

⚠ Choices are indexed 1-based. The first choice has index 1.

## Arguments

id `string`

## Example

## Request

```
query {  
  vote (  
    id: "QmeU7ct9Y4KLrh6F6mbT1eJNMkeQKMSnSujEfMCfbRLCMp"  
  ) {  
    id  
    voter  
    vp  
    vp_by_strategy  
    vp_state  
    created  
    proposal {  
      id  
    }  
    choice  
    space {  
      id  
    }  
  }  
}
```

## Response

```
{  
  "data": {  
    "vote": {  
      "id": "QmeU7ct9Y4KLrh6F6mbT1eJNMkeQKMSnSujEfMCfbRLCMp",  
      "voter": "0x96176C25803Ce4cF046aa74895646D8514Ea1611",  
      "created": 1621183227,  
      "proposal": {  
        "id": "QmPvbwguLfcVryzBRrbY4Pb9bCtxURagdv1XjhtFLf3wHj"  
      },  
      "choice": 1,  
      "space": {  
        "id": "spookyswap.eth"  
      }  
    }  
  }  
}
```

Try on [GraphiQL](#)

## Get votes

⚠ Choices are indexed 1-based. The first choice has index 1.

### Arguments

first `number`  
skip `number`  
where:  
- id `string`  
- id\_in `array`  
- space: `string`  
- space\_in: `array`  
- voter: `string`  
- voter\_in: `array`  
- proposal: `string`  
- proposal\_in: `array`  
orderBy `string`  
orderDirection `asc` or `desc`

### Example

Request

```
query {  
  votes (  
    first: 1000  
    skip: 0  
    where: {  
      proposal: "QmPvbwguLfcVryzBRrbY4Pb9bCtxURagdV1XjhtFLf3wHj"  
    }  
    orderBy: "created",  
    orderDirection: desc  
  ) {  
    id  
    voter  
    vp  
    vp_by_strategy  
    vp_state  
    created  
    proposal {  
      id  
    }  
    choice  
    space {  
      id  
    }  
  }  
}
```

## Response

```
{
  "data": {
    "votes": [
      {
        "id": "QmeU7ct9Y4KLrh6F6mbT1eJNMkeQKMSnSuJEfMCfbRLCMp",
        "voter": "0x96176C25803Ce4cF046aa74895646D8514Ea1611",
        "created": 1621183227,
        "proposal": {
          "id": "QmPvbwguLfcVryzBRrbY4Pb9bCtxURagdV1XjhtFLf3wHj"
        },
        "choice": 1,
        "space": {
          "id": "spookyswap.eth"
        }
      },
      {
        "id": "QmZ2CV86QH6Q6z7L6g7yJWS3HfgD9aQ3uTYMXkMa5trHf",
        "voter": "0x2686EaD94C5042e56a41eDde6533711a4303CC52",
        "created": 1621181827,
        "proposal": {
          "id": "QmPvbwguLfcVryzBRrbY4Pb9bCtxURagdV1XjhtFLf3wHj"
        },
        "choice": 1,
        "space": {
          "id": "spookyswap.eth"
        }
      },
      ...
    ]
  }
}
```

Try on [GraphiQL](#)

## Get voting power

### Arguments

voter

space

proposal

## Example

### Request

```
query {  
  vp (  
    voter: "0xF8305E140ac520225DAf050e2f71d5fBcC543e7"  
    space: "fabien.eth"  
    proposal: "0x4903dd16990de740b7dc7effe1a0bc8bd49a510a04992bc30596"  
  ) {  
    vp  
    vp_by_strategy  
    vp_state  
  }  
}
```

### Response

```
{  
  "data": {  
    "vp": {  
      "vp": 1,  
      "vp_by_strategy": [  
        1  
      ],  
      "vp_state": "final"  
    }  
  }  
}
```

Try on [GraphiQL](#)

## Get follows

### Arguments

first

skip

where:

- id string
  - id\_in array
  - space: string
  - space\_in: array
  - follower: string
  - follower\_in: array
- orderBy string
- orderDirection asc or desc

## Example

### Request

```
query {  
  follows(  
    first: 10,  
    where: {  
      follower: "0xeF8305E140ac520225DAf050e2f71d5fBcC543e7"  
    }  
  ) {  
    follower  
    space {  
      id  
    }  
    created  
  }  
}
```

### Response

```
{
  "data": {
    "follows": [
      {
        "follower": "0xeF8305E140ac520225DAf050e2f71d5fBcC543e7",
        "space": {
          "id": "gnosis.eth"
        },
        "created": 1629732280
      },
      {
        "follower": "0xeF8305E140ac520225DAf050e2f71d5fBcC543e7",
        "space": {
          "id": "aavegotchi.eth"
        },
        "created": 1629725098
      },
      {
        "follower": "0xeF8305E140ac520225DAf050e2f71d5fBcC543e7",
        "space": {
          "id": "yam.eth"
        },
        "created": 1629723970
      },
      {
        "follower": "0xeF8305E140ac520225DAf050e2f71d5fBcC543e7",
        "space": {
          "id": "balancer.eth"
        },
        "created": 1629723960
      }
    ]
  }
}
```

Try on [GraphiQL](#)

## Get users

## Arguments



first `number`  
skip `number`  
where:  
- id: `string`  
- id\_in: `array`  
orderBy `string`  
orderDirection `asc` or `desc`

## Example

### Request

```
query {  
  users(first: 10, where: { id_in: ["0xF78108c9BBaF466dd96BE41be728Fe:  
    id  
    name  
    about  
    avatar  
  }  
}
```

### Response

```
{  
  "data": {  
    "users": [  
      {  
        "id": "0xF78108c9BBaF466dd96BE41be728Fe3220b37119",  
        "name": "John Doe",  
        "about": "Lorem ipsum dolor sit amet consectetur adipisicing (",  
        "avatar": "ipfs://QmNXTswsKJEHHEmGCgQKEAqbq3ib1eBCd4U8SRPgcuv",  
      }  
    ]  
  }  
}
```

Try on [GraphiQL](#)

# Get roles

## Arguments

first `number`

skip `number`

where:

- address: `string`

orderBy `string`

orderDirection `asc` or `desc`

### Request

```
query {  
  roles(where:{address:"0xE029Ef62e47E394BC852EFf633eB5aa4A223ECa6"})  
    space  
    permissions  
  }  
}
```

### Response

```
{
  "data": {
    "roles": [
      {
        "space": "fabien.eth",
        "permissions": [
          "moderator"
        ]
      },
      {
        "space": "zuzazuber.eth",
        "permissions": [
          "moderator"
        ]
      }
    ]
  }
}
```

Try on GraphiQL

## Aliases

TBD

## Get messages

Messages are all the actions (votes, proposals, space settings etc..) that was confirmed on Snapshot, it also include the order on which these actions were confirmed with the field "mci". These messages can be used to replay the whole Snapshot hub API.

## Arguments

first

skip

where:

- timestamp `string`
- space `array`
- space\_in: `array`
- type: `string`
- type\_in: `string`

orderBy `string`

orderDirection `asc` or `desc`

## Example

### Request

```
query {  
  messages (  
    first: 20  
    where: { space: "ens.eth" }  
    orderBy: "mci"  
    orderDirection: desc  
  ) {  
    id  
    address  
    ipfs  
    receipt  
    type  
    mci  
  }  
}
```

### Response

```
{
  "data": {
    "messages": [
      {
        "id": "0x7d99e11ffe3a333229bdcda59866bc5b66b0b7e5c4b5353862a3l",
        "address": "0xdbB1740e424C41E935599634828f5E5c4dF23D43",
        "ipfs": "bafkreiduchkd35btpv3utttrg3f2kx3g52uh44tiruawip6xlgvjl",
        "receipt": "0xaa4d557085872ed82b8bfba5af4247650395eda46ed6167:",
        "type": "follow",
        "mci": 1345000
      },
      ...
    ]
  }
}
```

Try on [GraphiQL](#)

# API Keys

To obtain higher rate limits on the Hub API use, apply for an API Key.

We want to ensure that we limit the risk of API downtime and provide a reliable and continuous service. Therefore we decided to implement API Keys to ensure that the requests come from genuine users.


 You can use **the same API key** for different Snapshot APIs.

Limits are **counted individually** per each API Service.

## Limits

 **No API Key:** 100 requests per minute.

 **With the API Key:** 2 million requests per month.

 If you are getting **close to 2M requests a month** contact our support on [Help Center](#)

## How can I get an API Key?

### 1. Apply via the request form

If you haven't already please **fill in the below form** or submit it via [the direct link](#).

### API Key Request Form

This API is rate-limited. You can experiment with it but if you'd like to use it in a production environment, please request a free API key [here](#).

production environment, please request a free API key here.

What is your project name?

Provide a short description of your project:

Link your project's homepage or github repo:

Add the telegram handle of the person responsible for your project's use of the API:

Add the email address that's best to get in touch with you:

Wallet address. Please use your 42-character hexadecimal address, not your ENS name.

## 2. Wait 72 hours

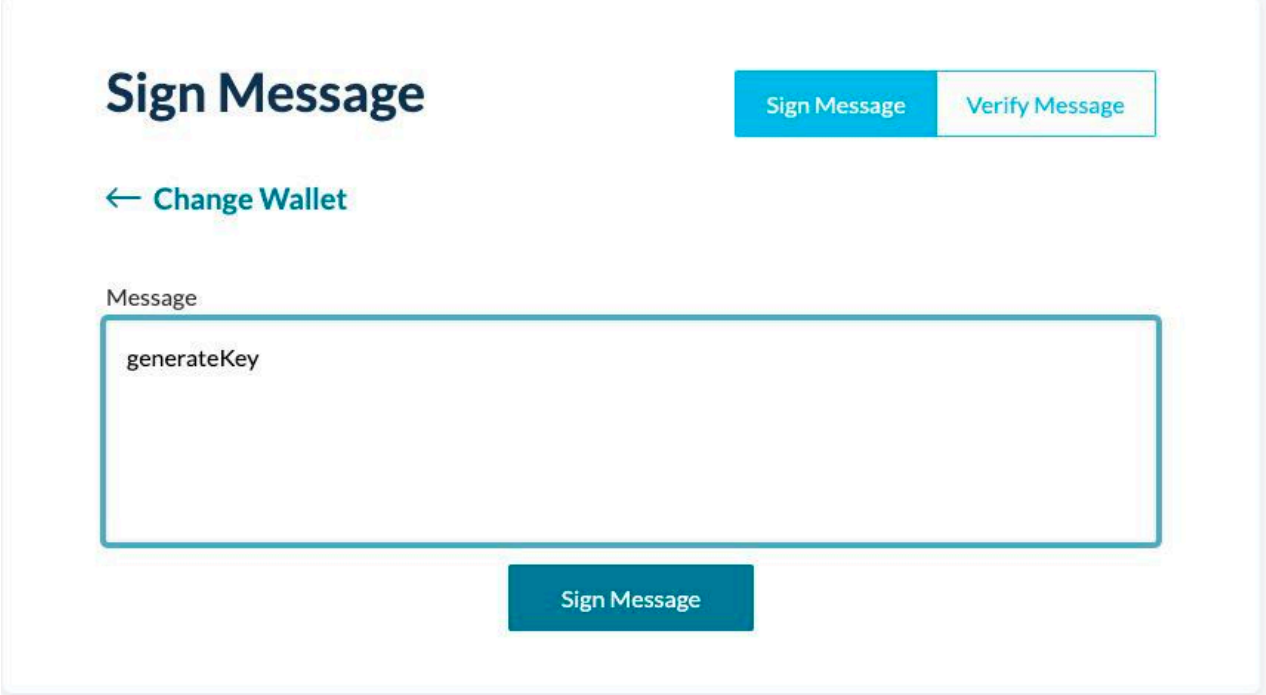
We will review your submission and **whitelist the address** you provided in the form.

**After 72 hours** have passed you can continue with the next steps. If you get any errors while generating the key, please contact our support on [Help Center](#).

## 3. Generate the API Key

After your address has been whitelisted go to <https://app.mycrypto.com/sign-message> and connect your wallet using the account you provided in the submission form above.

a) Sign the message with keyword `generateKey`:



Untitled

b) Copy the signature and run the below `curl` command.

```
curl --location 'https://keycard.snapshot.org' \
--header 'accept: application/json' \
--header 'content-type: application/json' \
--data '{
  "jsonrpc": "2.0",
  "method": "generate_key",
  "params": {
    "sig": "<SIGNATURE_HASH>"
  },
  "id": "123456789"
}'
```

Make sure to use the signature hash from step 5 in the `sig` param, **do not paste** the entire response after signing the message. As an example:



```
...  
  "method": "generate_key",  
  "params": {  
    "sig": "0x85bcabdeb3b43131364d21b32f8c74124d155009fc9d6d40901b4b7",  
  },  
  ...
```

You will receive the API Key as a response of the `curl` request. Make sure to store it securely.

## How to structure the query with my key?

The only change you need to make is to add the API Key in the headers of your request:

```
curl 'https://hub.snapshot.org/graphql?' \  
  -H 'content-type: application/json' \  
  -H 'x-api-key: <YOUR-API-KEY>' \  
  --data-raw '{"query": "\n{\n  space(id: \"snapshot.dcl.eth\") {\n    id\n    na  
  }\"}' \  
  --compressed
```

Alternatively, you can use the `apiKey` param in the query string with your key as a value:

```
curl 'https://hub.snapshot.org/graphql?apiKey=<YOUR_API_KEY>' \  
  -H 'content-type: application/json' \  
  --data-raw '{"query": "\n{\n  space(id: \"snapshot.dcl.eth\") {\n    id\n    na  
  }\"}' \  
  --compressed
```

If you are using the GraphQL interface you need to provide your key in the `headers` tab:

QUERY VARIABLES		REQUEST HEADERS	
1	{		
2	'x-api-key': 'secret'		
3	}		