tinybird

Pipes API

The Pipes API allows you to interact with your Pipes. Several services are included under Pipes:

- Pipes: list, create, update or delete your Tinybird Pipes.
- API Endpoints: publish or unpublish your Pipes as API Endpoints.
- Materialized Views: create, delete or populate Materialized Views.
- Scheduled Copy: create, delete, schedule and trigger Copy jobs.

New to Pipes? Read more about them here

All endpoints require authentication using an Auth Token with the appropriate scope.

How to create a Tinybird Pipe

Imagine you have an events Data Source named `app_events` and you want to expose an API endpoint containing the aggregation of events per day.

First, create a Pipe. Let's call this one `events_per_day`:

CREATING A PIPE FROM A DATASOURCE

```
curl \
-H "Authorization: Bearer <token>" \
-d "name=events_per_day&sql=select * from app_events" \
https://api.tinybird.co/v0/pipes
```

As shown above, creating the Pipe also defines the first transformation node SQL query, which in this case gets all the data from the Data Source. Below shows a successful request response after creating a Pipe:

PIPE SUCCESSFULLY CREATED

```
{
  "id": "t_1b501ccf34764a69aaf886bac9a7a6d8",
  "name": "events_per_day",
  "published_version": "t_878a81f0e1344a2ca990c8c1aa7dd69f",
  "published_date": "2019-06-14 09:46:07.884855",
      "nodes": [
        {
          "name": "events_per_day_0",
          "sql": "select * from app_events",
          "id": "t_878a81f0e1344a2ca990c8c1aa7dd69f",
          "dependencies": [
            "app_events"
          ],
          "materialized": false,
          "created_at": "2019-06-14 09:46:07.884849"
```

Now, we are going to add a new transformation node to perform the aggregation per date using the previously created node `events_per_day_0`. Let's use "append" endpoint.

APPENDING A NEW NODE TO TRANSFORM YOUR DATA

```
curl -d "select toDate(timestamp) date, count() event_count from events_per_day_
  -H "Authorization: Bearer <token>" \
  https://api.tinybird.co/v0/pipes/events_per_day/nodes
```

Whenever a transformation node is added to a Pipe, you receive a response like the one below, summarizing and giving you an autogenerated name for the node, as well as some metadata such as the dependencies with other transformation nodes:

SUCCESSFUL RESPONSE

```
{
  "name": "events_per_day_1",
  "sql": "select toDate(timestamp) date, count() event_count from events_per_day
  "id": "t_5164622050b244338ea2b79c19bd1e57",
  "dependencies": [
    "events_per_day_0"
  ],
  "materialized": false,
  "created_at": "2019-06-14 09:58:08.311812"
}
```

In order to make a Pipe publicly accessible through the API, you need to enable your desired transformation node as an API Endpoint. Remember, Pipes only support one enabled node each, so enabling one will make previously-enabled nodes inaccessible.

ENABLING A TRANSFORMATION NODE AS AN API ENDPOINT

```
curl \
  -X PUT \
  -H "Authorization: Bearer <token>" \
  -d t_878a81f0e1344a2ca990c8c1aa7dd69f \
  https://api.tinybird.co/v0/pipes/events_per_day/endpoint
```

When enabling a transformation node as an API Endpoint, a JSON containing the full Pipe description is sent as the response.

SUCCESSFUL RESPONSE

```
{
  "id": "t_1b501ccf34764a69aaf886bac9a7a6d8",
  "name": "events_per_day",
  "published_version": "t_5164622050b244338ea2b79c19bd1e57",
  "published_date": "2019-06-14 10:17:01.201962",
  "nodes": [
    {
      "name": "events_per_day_0",
      "sql": "select * from app_events",
      "id": "t_878a81f0e1344a2ca990c8c1aa7dd69f",
      "dependencies": [
        "app_events"
      ],
      "materialized": false,
      "created_at": "2019-06-14 10:17:01.201784"
```

Once the Pipe is created and we've enabled a transformation node as an Endpoint, we can easily integrate it into our 3rd party application.

Using Query API (see the Query API docs) you can query the Pipe using its name just like a regular table in a SQL query: `SELECT * FROM events_per_day where date > yesterday()`

QUERYING A PIPE USING SQL

```
curl \
-H "Authorization: Bearer <token>" \
-d 'SELECT * FROM events_per_day where date > yesterday()' \
'https://api.tinybird.co/v0/sql'
```

If you don't need to run any special operation against your Pipe, you can just use the Pipe data endpoint accessible at `https://api.tinybird.co/v0/pipes/events_per_day.json`. It's an alias for `SELECT * FROM events_per_day`

Pipes are updated in real-time, so as you insert the new data in `app_events` Data Source, every Pipe using it `events_per_day` is updated.

In order to be able to share this endpoint, you can create a READ token. You can add a new token for the pipe as follows:

ADDING A READ TOKEN TO AN ENDPOINT PIPE

```
curl -X POST "https://api.tinybird.co/v0/tokens/?name=events_per_day_token&scope
```

Pipes

GET /v0/pipes/?

Get a list of pipes in your account.

GETTING A LIST OF YOUR PIPES

```
curl -X GET \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes"
```

Pipes in the response will be the ones that are accessible using a particular token with read permissions for them.

SUCCESSFUL RESPONSE

```
{
    "pipes": [{
        "id": "t_55c39255e6b548dd98cb6da4b7d62c1c",
        "name": "my_pipe",
        "description": "This is a description",
        "endpoint": "t_h65c788b42ce4095a4789c0d6b0156c3",
            "created_at": "2022-11-10 12:39:38.106380",
            "updated_at": "2022-11-29 13:33:40.850186",
            "parent": null,
            "nodes": [{
                "id": "t_h65c788b42ce4095a4789c0d6b0156c3",
                "name": "my_node",
                "sql": "SELECT col_a, col_b FROM my_data_source",
                "description": null,
                "materialized": null,
                "cluster": null,
                "tags": {},
                "created_at": "2022-11-10 12:39:47.852303",
                "updated_at": "2022-11-10 12:46:54.066133",
                "version": 0,
                "project": null,
                "result": null,
                "ignore_sql_errors": false
                "node_type": "default"
            }],
            "url": "https://api.tinybird.co/v0/pipes/my_pipe.json"
        }]
    }
```

Request parameters

KEY	TYPE	DESCRIPTION
dependencies	boolean	The response will include the nodes dependent data sources and pipes, default is `false`
attrs	String	comma separated list of the pipe attributes to return in the response. Example: `attrs=name,description`
node_attrs	String	comma separated list of the node attributes to return in the response. Example `node_attrs=id,name`

Pipes id's are immutable so you can always refer to them in your 3rd party applications to make them compatible with Pipes once they are renamed.

For lighter JSON responses consider using the `attrs` and `node_attrs` params to return exactly the attributes you need to consume.

POST /v0/pipes/?

Creates a new Pipe. There are 3 ways to create a Pipe

CREATING A PIPE PROVIDING FULL JSON

```
curl -X POST \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    -H "Content-Type: application/json" \
    "https://api.tinybird.co/v0/pipes" -d
    '{
        "name":"pipe_name",
        "description": "my first pipe",
        "nodes": [
            {"sql": "select * from my_datasource limit 10", "name": "node_00", "
            {"sql": "select count() from node_00", "name": "node_01" }
        ]
    }'
```

If you prefer to create the minimum Pipe, and then append your transformation nodes you can set your name and first transformation node's SQL in your POST request

CREATING A PIPE WITH A NAME AND A SQL QUERY

```
curl -X POST \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes?name=pipename&sql=select%20*%20from%20ever
```

Pipes can be also created as copies of other Pipes. Just use the `from` argument:

CREATING A PIPE FROM ANOTHER PIPE

```
curl -X POST \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes?name=pipename&from=src_pipe"
```

Bear in mind, if you use this method to overwrite an existing Pipe, the endpoint will only be maintained if the node name is the same.

POST /v0/pipes/(.+)/nodes

Appends a new node to a Pipe.

ADDING A NEW NODE TO A PIPE

```
curl \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    -d 'select * from node_0' "https://api.tinybird.co/v0/pipes/:name/nodes?name
```

Appends a new node that creates a Materialized View

ADDING A MATERIALIZED VIEW USING A MATERIALIZED NODE

```
curl \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    -d 'select id, sum(amount) as amount, date from my_datasource' "https://api.
```

Request parameters

KEY	TYPE	DESCRIPTION
name	String	The referenceable name for the node.
description	String	Use it to store a more detailed explanation of the node.
token	String	Auth token. Ensure it has the `PIPE:CREATE` scope on it
type	String	Optional. Available options are {`standard` (default), `materialized`, `endpoint`}. Use `materialized` to create a Materialized View from your new node.
datasource	String	Required with `type=materialized`. Specifies the name of the destination Data Source where the Materialized View schema is defined.

KEY	TYPE	DESCRIPTION
override_datasource	Boolean	Optional. Default `false` When the target Data Source of the Materialized View exists in the Workspace it'll be overriden by the `datasource` specified in the request.
populate	Boolean	Optional. Default `false`. When `true`, a job is triggered to populate the destination Data Source.
populate_subset	Float	Optional. Populate with a subset percent of the data (limited to a maximum of 2M rows), this is useful to quickly test a materialized node with some data. The subset must be greater than 0 and lower than 0.1. A subset of 0.1 means a 10 percent of the data in the source Data Source will be used to populate the Materialized View. Use it together with `populate=true`, it has precedence over `populate_condition`
populate_condition	String	Optional. Populate with a SQL condition to be applied to the trigger Data Source of the Materialized View. For instance, `populate_condition='date == toYYYYMM(now())'` it'll populate taking all the rows from the trigger Data Source which `date` is the current month. Use it together with `populate=true`. `populate_condition` is not taken into account if the `populate_subset` param is present. Including in the `populate_condition` any column present in the Data Source `engine_sorting_key` will make the populate job process less data.
unlink_on_populate_error	String	Optional. Default is `false`. If the populate job fails the Materialized View is unlinked and new data won't be ingested in the Materialized View.
engine	String	Optional. Engine for destination Materialized View. Requires the `type` parameter as

KEY	TYPE	DESCRIPTION
		`materialized`.
engine_*	String	Optional. Engine parameters and options. Requires the `type` parameter as `materialized` and the `engine` parameter. Check Engine Parameters and Options for more details

SQL query for the transformation node must be sent in the body encoded in utf-8

Response codes

CODE	DESCRIPTION
200	No error
400	empty or wrong SQL or API param value
403	Forbidden. Provided token doesn't have permissions to append a node to the pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found
409	There's another resource with the same name, names must be unique | The Materialized View already exists | `override_datasource` cannot be performed

DELETE /v0/pipes/(.+)/nodes/(.+)

Drops a particular transformation node in the Pipe. It does not remove related nodes so this could leave the Pipe in an unconsistent state. For security reasons, enabled nodes can't be removed.

REMOVING A NODE FROM A PIPE

```
curl -X DELETE "https://api.tinybird.co/v0/pipes/:name/nodes/:node_id"
```

Response codes

CODE	DESCRIPTION

CODE	DESCRIPTION
204	No error, removed node
400	The node is published. Published nodes can’t be removed
403	Forbidden. Provided token doesn’t have permissions to change the last node of the pipe, it needs ADMIN or IMPORT
404	Pipe not found

PUT /v0/pipes/(.+)/nodes/(.+)

Changes a particular transformation node in the Pipe

EDITING A PIPE’S TRANSFORMATION NODE

```
curl -X PUT \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    -d 'select * from node_0' "https://api.tinybird.co/v0/pipes/:name/nodes/:noc
```

Request parameters

KEY	TYPE	DESCRIPTION
name	String	new name for the node
description	String	new description for the node
token	String	Auth token. Ensure it has the `PIPE:CREATE` scope on it

Please, note that the desired SQL query should be sent in the body encoded in utf-8.

Response codes

CODE	DESCRIPTION
200	No error
400	Empty or wrong SQL

CODE	DESCRIPTION
403	Forbidden. Provided token doesn't have permissions to change the last node to the pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found
409	There's another resource with the same name, names must be unique

GET /v0/pipes/(.+)\.(json|csv|ndjson|parquet)

Returns the published node data in a particular format.

GETTING DATA FOR A PIPE

```
curl -X GET \
    -H "Authorization: Bearer <PIPE:READ token>" \
    "https://api.tinybird.co/v0/pipes/:name.format"
```

Request parameters

KEY	TYPE	DESCRIPTION
q	String	Optional, query to execute, see API Query endpoint
output_format_json_quote_64bit_integers	int	(Optional) Controls quoting of 64-bit or bigger integers (like UInt64 or Int128) when they are output in a JSON format. Such integers are enclosed in quotes by default. This behavior is compatible with most JavaScript implementations. Possible values: 0 — Integers are output without quotes. 1 — Integers are enclosed in quotes. Default value is 0
output_format_json_quote_denormals	int	(Optional) Controls representation of inf and nan

KEY	TYPE	DESCRIPTION
		on the UI instead of null e.g when dividing by 0 - inf and when there is no representation of a number in Javascript - nan. Possible values: 0 - disabled, 1 - enabled. Default value is 0
output_format_parquet_string_as_string	int	(Optional) Use Parquet String type instead of Binary for String columns. Possible values: 0 - disabled, 1 - enabled. Default value is 0

The `q` parameter is a SQL query (see Query API). When using this endpoint to query your Pipes, you can use the `_` shortcut, which refers to your Pipe name

Available formats

FORMAT	DESCRIPTION
csv	CSV with header
json	JSON including data, statistics and schema information
ndjson	One JSON object per each row
parquet	A Parquet file. Some libraries might not properly process `UInt*` data types, if that's your case cast those columns to signed integers with `toInt*` functions. `String` columns are exported as `Binary`, take that into account when reading the resulting Parquet file, most libraries convert from Binary to String (e.g. Spark has this configuration param: `spark.sql.parquet.binaryAsString`)

POST /v0/pipes/(.+)\.(json|csv|ndjson|parquet)

Returns the published node data in a particular format, passing the parameters in the request body. Use this endpoint when the query is too long to be passed as a query string parameter.

When using the post endpoint, there are no traces of the parameters in the pipe_stats_rt Data Source.

See the get endpoint for more information.

GET /v0/pipes/(.+\.pipe)

Get pipe information. Provided Auth Token must have read access to the Pipe.

GETTING INFORMATION ABOUT A PARTICULAR PIPE

```
curl -X GET \
    -H "Authorization: Bearer <PIPE:READ token>" \
    "https://api.tinybird.co/v0/pipes/:name"
```

`pipe_id` and `pipe_name` are two ways to refer to the pipe in SQL queries and API endpoints the only difference is `pipe_id` never changes so it'll work even if you change the `pipe_name` (which is the name used to display the pipe). In general you can use `pipe_id` or `pipe_name` indistinctly:

SUCCESSFUL RESPONSE

```
{
    "id": "t_bd1c62b5e67142bd9bf9a7f113a2b6ea",
    "name": "events_pipe",
    "pipeline": {
        "nodes": [{
            "name": "events_ds_0"
            "sql": "select * from events_ds_log__raw",
            "materialized": false
        }, {
            "name": "events_ds",
            "sql": "select * from events_ds_0 where valid = 1",
            "materialized": false
        }]
    }
}
```

You can make your Pipe's id more descriptive by prepending information such as `t_my_events_table.bd1c62b5e67142bd9bf9a7f113a2b6ea`

DELETE /v0/pipes/(.+\.pipe)

Drops a Pipe from your account. Auth token in use must have the `DROP:NAME` scope.

DROPPING A PIPE

```
curl -X DELETE \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes/:name"
```

PUT /v0/pipes/(.+\.pipe)

Changes Pipe's metadata. When there is another Pipe with the same name an error is raised.

EDITING A PIPE

```
curl -X PUT \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes/:name?name=new_name"
```

Request parameters

KEY	TYPE	DESCRIPTION
name	String	new name for the pipe
description	String	new Markdown description for the pipe
token	String	Auth token. Ensure it has the `PIPE:CREATE` scope on it

GET /v0/pipes/(.+)

Get pipe information. Provided Auth Token must have read access to the Pipe.

GETTING INFORMATION ABOUT A PARTICULAR PIPE

```
curl -X GET \
    -H "Authorization: Bearer <PIPE:READ token>" \
    "https://api.tinybird.co/v0/pipes/:name"
```

`pipe_id` and `pipe_name` are two ways to refer to the pipe in SQL queries and API endpoints the only difference is `pipe_id` never changes so it'll work even if you change the `pipe_name` (which is the name used to display the pipe). In general you can use `pipe_id` or `pipe_name` indistinctly:

SUCCESSFUL RESPONSE

```
{
    "id": "t_bd1c62b5e67142bd9bf9a7f113a2b6ea",
    "name": "events_pipe",
    "pipeline": {
        "nodes": [{
            "name": "events_ds_0"
            "sql": "select * from events_ds_log__raw",
            "materialized": false
        }, {
            "name": "events_ds",
            "sql": "select * from events_ds_0 where valid = 1",
            "materialized": false
        }]
    }
}
```

You can make your Pipe's id more descriptive by prepending information such as `t_my_events_table.bd1c62b5e67142bd9bf9a7f113a2b6ea`

DELETE /v0/pipes/(.+)

Drops a Pipe from your account. Auth token in use must have the `DROP:NAME` scope.

DROPPING A PIPE

```
curl -X DELETE \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes/:name"
```

PUT /v0/pipes/(.+)

Changes Pipe's metadata. When there is another Pipe with the same name an error is raised.

EDITING A PIPE

```
curl -X PUT \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes/:name?name=new_name"
```

Request parameters

KEY	TYPE	DESCRIPTION
name	String	new name for the pipe
description	String	new Markdown description for the pipe
token	String	Auth token. Ensure it has the `PIPE:CREATE` scope on it

GET /v0/pipes/(.+)/nodes/(.+)/explain

Return the explain plan and the whole debug query for the node.

If no node is specified, the most relevant node of the pipe will be used:

- The endpoint node for endpoints.
- The node that materializes for materialized views.
- The copy node for Copy pipes.
- The last node for general pipes.

It accepts query parameters to test the query with different values.

GETTING THE EXPLAIN PLAN

```
curl -X GET \
    -H "Authorization: Bearer <PIPE:READ and DATASOURCE:READ token>" \
    "https://api.tinybird.co/v0/pipes/:pipe_name/nodes/:node_name/explain?depart

or

curl -X GET \
    -H "Authorization: Bearer <PIPE:READ and DATASOURCE:READ token>" \
    "https://api.tinybird.co/v0/pipes/:pipe_name/explain?department=Engineering"
```

SUCCESSFUL RESPONSE

```
{
    "debug_query": "SELECT country, department FROM (SELECT * FROM employees AS
    "query_explain": "Expression ((Projection + Before ORDER BY)) Filter ((WHERE
}
```

Request parameters

KEY	TYPE	DESCRIPTION
token	String	Auth token. Ensure it has the `PIPE:READ` and the proper `DATASOURCE:READ` scopes on it.
pipe_name	Float	The name or id of the pipe.
node_name	String	Optional. The name or id of the node to explain. If not provided, the most relevant node of the pipe will be used.
params	String	Optional. The value of the parameters to test the query with. They are regular URL query parameters.

Response codes

CODE	DESCRIPTION
200	No error
400	Could not get a node to run the explain plan
403	Forbidden. Provided token doesn't have permissions to run the explain plan, it needs `ADMIN` or `PIPE:READ` and `DATASOURCE:READ`
404	Pipe not found, Node not found

GET /v0/pipes/(.+)/explain

Return the explain plan and the whole debug query for the node.

If no node is specified, the most relevant node of the pipe will be used:

- The endpoint node for endpoints.

- The node that materializes for materialized views.
- The copy node for Copy pipes.
- The last node for general pipes.

It accepts query parameters to test the query with different values.

GETTING THE EXPLAIN PLAN

```
curl -X GET \
    -H "Authorization: Bearer <PIPE:READ and DATASOURCE:READ token>" \
    "https://api.tinybird.co/v0/pipes/:pipe_name/nodes/:node_name/explain?depart

or

curl -X GET \
    -H "Authorization: Bearer <PIPE:READ and DATASOURCE:READ token>" \
    "https://api.tinybird.co/v0/pipes/:pipe_name/explain?department=Engineering"
```

SUCCESSFUL RESPONSE

```
{
    "debug_query": "SELECT country, department FROM (SELECT * FROM employees AS
    "query_explain": "Expression ((Projection + Before ORDER BY)) Filter ((WHERE
}
```

Request parameters

KEY	TYPE	DESCRIPTION
token	String	Auth token. Ensure it has the `PIPE:READ` and the proper `DATASOURCE:READ` scopes on it.
pipe_name	Float	The name or id of the pipe.
node_name	String	Optional. The name or id of the node to explain. If not provided, the most relevant node of the pipe will be used.
params	String	Optional. The value of the parameters to test the query with. They are regular URL query parameters.

Response codes

CODE	DESCRIPTION
200	No error
400	Could not get a node to run the explain plan
403	Forbidden. Provided token doesn't have permissions to run the explain plan, it needs `ADMIN` or `PIPE:READ` and `DATASOURCE:READ`
404	Pipe not found, Node not found

API Endpoints

POST /v0/pipes/(.+)/nodes/(.+)/endpoint

Publishes an API endpoint

PUBLISHING AN ENDPOINT

```
curl -X POST \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/endpoint"
```

SUCCESSFUL RESPONSE

```
{
    "id": "t_60d8f84ce5d349b28160013ce99758c7",
    "name": "my_pipe",
    "description": "this is my pipe description",
    "nodes": [{
        "id": "t_bd1e095da943494d9410a812b24cea81",
        "name": "get_all",
        "sql": "SELECT * FROM my_datasource",
        "description": "This is a description for the **first** node",
        "materialized": null,
        "cluster": null,
        "dependencies": ["my_datasource"],
        "tags": {},
        "created_at": "2019-09-03 19:56:03.704840",
            "updated_at": "2019-09-04 07:05:53.191437",
            "version": 0,
            "project": null,
            "result": null,
            "ignore_sql_errors": false
        }],
        "endpoint": "t_bd1e095da943494d9410a812b24cea81",
        "created_at": "2019-09-03 19:56:03.193446",
        "updated_at": "2019-09-10 07:18:39.797083",
        "parent": null
    }
```

The response will contain a `token` if there's a **unique READ token** for this pipe. You could use this token to share your endpoint.

Response codes

CODE	DESCRIPTION
200	No error
400	Wrong node id
403	Forbidden. Provided token doesn't have permissions to publish a pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found

DELETE /v0/pipes/(.+)/nodes/(.+)/endpoint

Unpublishes an API endpoint

UNPUBLISHING AN ENDPOINT

```
curl -X DELETE \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/endpoint"
```

Response codes

CODE	DESCRIPTION

CODE	DESCRIPTION
200	No error
400	Wrong node id
403	Forbidden. Provided token doesn't have permissions to publish a pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found

Materialized Views and Populates

POST /v0/pipes/(.+)/nodes/(.+)/population

Populates a Materialized View

POPULATING A MATERIALIZED VIEW

```
curl
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    -X POST "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/population" \
    -d "populate_condition=toYYYYMM(date) = 202203"
```

The response will not be the final result of the import but a Job. You can check the job status and progress using the Jobs API.

Alternatively you can use a query like this to check the operations related to the populate Job:

CHECK POPULATE JOBS IN THE DATASOURCES_OPS_LOG INCLUDING DEPENDENT MATERIALIZED VIEWS TRIGGERED

```
SELECT *
FROM tinybird.datasources_ops_log
WHERE
    timestamp > now() - INTERVAL 1 DAY
    AND operation_id IN (
        SELECT operation_id
        FROM tinybird.datasources_ops_log
        WHERE
            timestamp > now() - INTERVAL 1 DAY
```

```
                and datasource_id = '{the_datasource_id}'
                and job_id = '{the_job_id}'
        )
ORDER BY timestamp ASC
```

When a populate job fails for the first time, the Materialized View is automatically unlinked. In that case you can get failed population jobs and their errors to fix them with a query like this:

CHECK FAILED POPULATE JOBS

```
SELECT *
FROM tinybird.datasources_ops_log
WHERE
    datasource_id = '{the_datasource_id}'
    AND pipe_name = '{the_pipe_name}'
    AND event_type LIKE 'populateview%'
    AND result = 'error'
ORDER BY timestamp ASC
```

Alternatively you can use the `unlink_on_populate_error='true'` flag to always unlink the Materialized View if the populate job does not work as expected.

Request parameters

KEY	TYPE	DESCRIPTION
token	String	Auth token. Ensure it has the `PIPE:CREATE` scope on it
populate_subset	Float	Optional. Populate with a subset percent of the data (limited to a maximum of 2M rows), this is useful to quickly test a materialized node with some data. The subset must be greater than 0 and lower than 0.1. A subset of 0.1 means a 10 percent of the data in the source Data Source will be used to populate the Materialized View. It has precedence over `populate_condition`
populate_condition	String	Optional. Populate with a SQL condition to be applied to the trigger Data Source of the Materialized View. For instance,

KEY	TYPE	DESCRIPTION
		`populate_condition='date == toYYYYMM(now())'` it'll populate taking all the rows from the trigger Data Source which `date` is the current month. `populate_condition` is not taken into account if the `populate_subset` param is present. Including in the `populate_condition` any column present in the Data Source `engine_sorting_key` will make the populate job process less data.
truncate	String	Optional. Default is `false`. Populates over existing data, useful to populate past data while new data is being ingested. Use `true` to truncate the Data Source before populating.
unlink_on_populate_error	String	Optional. Default is `false`. If the populate job fails the Materialized View is unlinked and new data won't be ingested in the Materialized View.

Response codes

CODE	DESCRIPTION
200	No error
400	Node is not materialized
403	Forbidden. Provided token doesn't have permissions to append a node to the pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found, Node not found

POST /v0/pipes/(.+)/nodes/(.+)/materialization

Creates a Materialized View

CREATING A MATERIALIZED VIEW

```
curl \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    -X POST "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/materialization?
```

Request parameters

KEY	TYPE	DESCRIPTION
token	String	Auth token. Ensure it has the `PIPE:CREATE` scope on it
datasource	String	Required. Specifies the name of the destination Data Source where the Materialized View schema is defined. If the Data Source does not exist, it creates automatically with the default settings.
override_datasource	Boolean	Optional. Default `false` When the target Data Source of the Materialized View exists in the Workspace it'll be overriden by the `datasource` specified in the request.
populate	Boolean	Optional. Default `false`. When `true`, a job is triggered to populate the destination datasource.
populate_subset	Float	Optional. Populate with a subset percent of the data (limited to a maximum of 2M rows), this is useful to quickly test a materialized node with some data. The subset must be greater than 0 and lower than 0.1. A subset of 0.1 means a 10 percent of the data in the source Data Source will be used to populate the Materialized View. Use it together with `populate=true`, it has precedence over `populate_condition`
populate_condition	String	Optional. Populate with a SQL condition to be applied to the trigger Data Source of the Materialized View. For instance, `populate_condition='date == toYYYYMM(now())'` it'll populate taking all the

KEY	TYPE	DESCRIPTION
		rows from the trigger Data Source which `date` is the current month. Use it together with `populate=true`. `populate_condition` is not taken into account if the `populate_subset` param is present. Including in the `populate_condition` any column present in the Data Source `engine_sorting_key` will make the populate job process less data.
unlink_on_populate_error	String	Optional. Default is `false`. If the populate job fails the Materialized View is unlinked and new data won't be ingested in the Materialized View.
engine	String	Optional. Engine for destination Materialized View. If the Data Source already exists, the settings are not overriden.
engine_*	String	Optional. Engine parameters and options. Requires the `engine` parameter. If the Data Source already exists, the settings are not overriden. Check Engine Parameters and Options for more details

SQL query for the materialized node must be sent in the body encoded in utf-8

Response codes

CODE	DESCRIPTION
200	No error
400	Node already being materialized
403	Forbidden. Provided token doesn't have permissions to append a node to the pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found, Node not found
409	The Materialized View already exists or `override_datasource` cannot be performed

DELETE /v0/pipes/(.+)/nodes/(.+)/materialization

Removes a Materialized View

By removing a Materialized View, nor the Data Source nor the Node are deleted. The Data Source will still be present, but will stop receiving data from the Node.

REMOVING A MATERIALIZED VIEW

```
curl
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    -X DELETE "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/materializatic
```

Response codes

CODE	DESCRIPTION
204	No error, Materialized View removed
403	Forbidden. Provided token doesn't have permissions to append a node to the pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found, Node not found

Copy Pipes API

The Copy Pipes API allows you to create, delete, schedule and trigger Copy Pipes.

POST /v0/pipes/(.+)/nodes/(.+)/copy

Calling this endpoint sets the pipe as a Copy one with the given settings. Scheduling is optional.

To run the actual copy after you set the pipe as a Copy one, you must call the POST `/v0/pipes/:pipe/copy` endpoint.

If you need to change the target Data Source or the scheduling configuration, you can call PUT endpoint.

Restrictions:

- You can set only one schedule per Copy pipe.

- You can’t set a Copy pipe if the pipe is already materializing. You must unlink the Materialization first.
- You can’t set a Copy pipe if the pipe is already an endpoint. You must unpublish the endpoint first.

SETTING THE PIPE AS A COPY PIPE

```
curl -X POST \
    -H "Authorization: Bearer <PIPE:CREATE and DATASOURCE:APPEND token>" \
      "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/copy" \
    -d "target_datasource=my_destination_datasource" \
    -d "schedule_cron=*/15 * * * *"
```

Request parameters

KEY	TYPE	DESCRIPTION
token	String	Auth token. Ensure it has the `PIPE:CREATE` and `DATASOURCE:APPEND` scopes on it
target_datasource	String	Name or the id of the target Data Source.
schedule_cron	String	Optional. A crontab expression.

SUCCESSFUL RESPONSE

```
{
    "id": "t_3aa11a5cabd1482c905bc8dfc551a84d",
    "name": "my_copy_pipe",
    "description": "This is a pipe to copy",
    "type": "copy",
    "endpoint": null,
    "created_at": "2023-03-01 10:14:04.497505",
    "updated_at": "2023-03-01 10:34:19.113518",
    "parent": null,
    "copy_node": "t_33ec8ac3c3324a53822fded61a83dbbd",
    "copy_target_datasource": "t_0be6161a5b7b4f6180b10325643e0b7b",
    "copy_target_workspace": "5a70f2f5-9635-47bf-96a9-7b50362d4e2f",
    "nodes": [{
        "id": "t_33ec8ac3c3324a53822fded61a83dbbd",
        "name": "emps",
        "sql": "SELECT * FROM employees WHERE starting_date > '2016-01-01 00
```

```
            "description": null,
            "materialized": null,
            "cluster": null,
            "tags": {
                "copy_target_datasource": "t_0be6161a5b7b4f6180b10325643e0b7b",
                "copy_target_workspace": "5a70f2f5-9635-47bf-96a9-7b50362d4e2f"
            },
            "created_at": "2023-03-01 10:14:04.497547",
            "updated_at": "2023-03-01 10:14:04.497547",
            "version": 0,
            "project": null,
            "result": null,
            "ignore_sql_errors": false,
            "dependencies": [ "employees" ],
            "params": []
        }]
    }
```

DELETE /v0/pipes/(.+)/nodes/(.+)/copy

Removes the Copy type of the pipe. By removing the Copy type, nor the node nor the pipe are deleted. The pipe will still be present, but will stop any scheduled and copy settings.

UNSETTING THE PIPE AS A COPY PIPE

```
curl -X DELETE \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
    "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/copy"
```

Response codes

CODE	DESCRIPTION
204	No error
400	Wrong node id
403	Forbidden. Provided token doesn’t have permissions to publish a pipe, it needs `ADMIN` or `PIPE:CREATE`
404	Pipe not found

PUT /v0/pipes/(.+)/nodes/(.+)/copy

Calling this endpoint will update a Copy pipe with the given settings: you can change its target Data Source, as well as adding or modifying its schedule.

UPDATING A COPY PIPE

```
curl -X PUT \
    -H "Authorization: Bearer <PIPE:CREATE token>" \
      "https://api.tinybird.co/v0/pipes/:pipe/nodes/:node/copy" \
    -d "target_datasource=other_destination_datasource" \
    -d "schedule_cron=*/15 * * * *"
```

Request parameters

KEY	TYPE	DESCRIPTION
token	String	Auth token. Ensure it has the `PIPE:CREATE` scope on it
target_datasource	String	Optional. Name or the id of the target Data Source.
schedule_cron	String	Optional. A crontab expression. If schedule_cron='None' the schedule will be removed from the copy pipe, if it was defined

SUCCESSFUL RESPONSE

```
{
    "id": "t_3aa11a5cabd1482c905bc8dfc551a84d",
    "name": "my_copy_pipe",
    "description": "This is a pipe to copy",
    "type": "copy",
    "endpoint": null,
    "created_at": "2023-03-01 10:14:04.497505",
    "updated_at": "2023-03-01 10:34:19.113518",
    "parent": null,
    "copy_node": "t_33ec8ac3c3324a53822fded61a83dbbd",
    "copy_target_datasource": "t_2f046a4b2cc44137834a35420a533465",
    "copy_target_workspace": "5a70f2f5-9635-47bf-96a9-7b50362d4e2f",
    "nodes": [{
        "id": "t_33ec8ac3c3324a53822fded61a83dbbd",
            "name": "emps",
            "sql": "SELECT * FROM employees WHERE starting_date > '2016-01-01 00
            "description": null,
            "materialized": null,
            "cluster": null,
            "tags": {
                "copy_target_datasource": "t_2f046a4b2cc44137834a35420a533465",
                "copy_target_workspace": "5a70f2f5-9635-47bf-96a9-7b50362d4e2f"
            },
            "created_at": "2023-03-01 10:14:04.497547",
            "updated_at": "2023-03-07 09:08:34.206123",
            "version": 0,
            "project": null,
            "result": null,
            "ignore_sql_errors": false,
            "dependencies": [ "employees" ],
            "params": []
        }]
    }
```

POST /v0/pipes/(.+)/copy

Runs a copy job, using the settings previously set in the pipe. You can use this URL to do an on-demand copy. This URL is also used by the scheduler to make the programmed calls.

This URL accepts parameters, just like in a regular endpoint.

This operation is asynchronous and will copy the output of the endpoint to an existing datasource.

RUNS A COPY JOB ON A COPY PIPE

```
curl
    -H "Authorization: Bearer <PIPE:READ token>" \
    -X POST "https://api.tinybird.co/v0/pipes/:pipe/copy?param1=test&param2=test
```

Request parameters

KEY	TYPE	DESCRIPTION

KEY	TYPE	DESCRIPTION
token	String	Auth token. Ensure it has the `PIPE:READ` scope on it
parameters	String	Optional. The value of the parameters to run the Copy with. They are regular URL query parameters.

Response codes

CODE	DESCRIPTION
200	No error
400	Pipe is not a Copy pipe or there is a problem with the SQL query
400	The columns in the SQL query don’t match the columns in the target Data Source
403	Forbidden. The provided token doesn’t have permissions to append a node to the pipe (`ADMIN` or `PIPE:READ` and `DATASOURCE:APPEND`)
403	Job limits exceeded. Tried to copy more than 100M rows, or there are too many active (working and waiting) Copy jobs.
404	Pipe not found, Node not found or Target Data Source not found

The response will not be the final result of the copy but a Job. You can check the job status and progress using the Jobs API.

SUCCESSFUL RESPONSE

```
{
    "id": "t_33ec8ac3c3324a53822fded61a83dbbd",
    "name": "emps",
    "sql": "SELECT * FROM employees WHERE starting_date > '2016-01-01 00:00:
    "description": null,
    "materialized": null,
    "cluster": null,
    "tags": {
        "copy_target_datasource": "t_0be6161a5b7b4f6180b10325643e0b7b",
        "copy_target_workspace": "5a70f2f5-9635-47bf-96a9-7b50362d4e2f"
    },
    "created_at": "2023-03-01 10:14:04.497547",
    "updated_at": "2023-03-01 10:14:04.497547",
        "version": 0,
        "project": null,
        "result": null,
        "ignore_sql_errors": false,
        "dependencies": [ "employees" ],
        "params": [],
        "job": {
            "kind": "copy",
            "id": "f0b2f107-0af8-4c28-a83b-53053cb45f0f",
            "job_id": "f0b2f107-0af8-4c28-a83b-53053cb45f0f",
            "status": "waiting",
            "created_at": "2023-03-01 10:41:07.398102",
            "updated_at": "2023-03-01 10:41:07.398128",
            "started_at": null,
            "is_cancellable": true,
            "datasource": {
                "id": "t_0be6161a5b7b4f6180b10325643e0b7b"
            },
            "query_id": "19a8d613-b424-4afd-95f1-39cfbd87e827",
            "query_sql": "SELECT * FROM d_b0ca70.t_25f928e33bcb40bd8e8999e69cb02
            "pipe_id": "t_3aa11a5cabd1482c905bc8dfc551a84d",
            "pipe_name": "copy_emp",
            "job_url": "https://api.tinybird.co/v0/jobs/f0b2f107-0af8-4c28-a83b-
        }
    }
```

POST /v0/pipes/(.+)/copy/pause

Pauses the scheduling. This affects any future scheduled Copy job. Any copy operation currently copying data will be completed.

PAUSES A SCHEDULED COPY

```
curl -X POST \
        -H "Authorization: Bearer <PIPE:CREATE token>" \
       "https://api.tinybird.co/v0/pipes/:pipe/copy/pause"
```

Response codes

CODE	DESCRIPTION
200	Scheduled copy paused correctly
400	Pipe is not copy
404	Pipe not found, Scheduled copy for pipe not found

POST /v0/pipes/(.+)/copy/resume

Resumes a previously paused scheduled copy.

RESUMES A SCHEDULED COPY

```
curl -X POST \
        -H "Authorization: Bearer <PIPE:CREATE token>" \
       "https://api.tinybird.co/v0/pipes/:pipe/copy/resume"
```

Response codes

CODE	DESCRIPTION
200	Scheduled copy resumed correctly
400	Pipe is not copy
404	Pipe not found, Scheduled copy for pipe not found

POST /v0/pipes/(.+)/copy/cancel

Cancels jobs that are working or waiting that are tied to the pipe and pauses the scheduling of copy jobs for this pipe. To allow scheduled copy jobs to run for the pipe, the copy pipe must be resumed and the already cancelled jobs will not be resumed.

CANCELS SCHEDULED COPY JOBS TIED TO THE PIPE

```
curl -X POST \
        -H "Authorization: Bearer <PIPE:CREATE token>" \
       "https://api.tinybird.co/v0/pipes/:pipe/copy/cancel"
```

Response codes

CODE	DESCRIPTION
200	Scheduled copy pipe cancelled correctly
400	Pipe is not copy
400	Job is not in cancellable status
400	Job is already being cancelled
404	Pipe not found, Scheduled copy for pipe not found

SUCCESSFUL RESPONSE

```
{
    "id": "t_fb56a87a520441189a5a6d61f8d968f4",
    "name": "scheduled_copy_pipe",
    "description": "none",
    "endpoint": "none",
    "created_at": "2023-06-09 10:54:21.847433",
    "updated_at": "2023-06-09 10:54:21.897854",
    "parent": "none",
    "type": "copy",
    "copy_node": "t_bb96e50cb1b94ffe9e598f870d88ad1b",
    "copy_target_datasource": "t_3f7e6534733f425fb1add6229ca8be4b",
    "copy_target_workspace": "8119d519-80b2-454a-a114-b092aea3b9b0",
    "schedule": {
        "timezone": "Etc/UTC",
        "cron": "0 * * * *",
        "status": "paused"
    },
    "nodes": [
        {
            "id": "t_bb96e50cb1b94ffe9e598f870d88ad1b",
            "name": "scheduled_copy_pipe_0",
            "sql": "SELECT * FROM landing_ds",
            "description": "none",
            "materialized": "none",
            "cluster": "none",
            "tags": {
                "copy_target_datasource": "t_3f7e6534733f425fb1add6229ca8be4
                "copy_target_workspace": "8119d519-80b2-454a-a114-b092aea3b9
            },
            "created_at": "2023-06-09 10:54:21.847451",
            "updated_at": "2023-06-09 10:54:21.847451",
            "version": 0,
            "project": "none",
            "result": "none",
            "ignore_sql_errors": "false",
            "node_type": "copy",
            "dependencies": [
                "landing_ds"
            ],
            "params": []
        }
    ],
    "cancelled_jobs": [
        {
            "id": "ced3534f-8b5e-4fe0-8dcc-4369aa256b11",
            "kind": "copy",
            "status": "cancelled",
            "created_at": "2023-06-09 07:54:21.921446",
            "updated_at": "2023-06-09 10:54:22.043272",
            "job_url": "https://api.tinybird.co/v0/jobsjobs/ced3534f-8b5e-4f
            "is_cancellable": "false",
            "pipe_id": "t_fb56a87a520441189a5a6d61f8d968f4",
            "pipe_name": "pipe_test_scheduled_copy_pipe_cancel_multiple_jobs
            "datasource": {
                "id": "t_3f7e6534733f425fb1add6229ca8be4b",
                "name": "target_ds_test_scheduled_copy_pipe_cancel_multiple_
            }
        },
        {
            "id": "b507ded9-9862-43ae-8863-b6de17c3f914",
            "kind": "copy",
            "status": "cancelling",
            "created_at": "2023-06-09 07:54:21.903036",
            "updated_at": "2023-06-09 10:54:22.044837",
            "job_url": "https://api.tinybird.co/v0/jobsb507ded9-9862-43ae-88
            "is_cancellable": "false",
            "pipe_id": "t_fb56a87a520441189a5a6d61f8d968f4",
            "pipe_name": "pipe_test_scheduled_copy_pipe_cancel_multiple_jobs
            "datasource": {
                "id": "t_3f7e6534733f425fb1add6229ca8be4b",
                "name": "target_ds_test_scheduled_copy_pipe_cancel_multiple_
```

```
                    }
                }
            ]
        }
```

Serverless ClickHouse

Spain
Calle del Dr. Fourquet, 27
28012 Madrid

USA
41 East 11th Street 11th floor
New York, NY 10003