

# DBSQL Dashboard Refresh Schedules API

**WARNING:** Direct requests to REST API should be considered unstable and entirely subject to change. Use at your own risk!

Share Scheduled ▾ Refresh ⌵

Refresh Every 1 minute ▾  
Advanced

SQL Endpoint ⓘ 19mar21-test2 (2XS) ✓ ▾

Subscribers (optional) Type a name or an email address  
The report will be emailed to subscribers every time it is updated.

Enabled ☒

Cancel Save

- Notice the requests made in Chrome Dev Tools:

Path

- /health
- /graphql
- /graphql
- /graphql
- /graphql
- /sql/api/dashboards/2835986c-2d1d-4a7d-b26b-0fa9039aa665/refresh\_schedules
- /sql/api/dashboards/2835986c-2d1d-4a7d-b26b-0fa9039aa665/refresh\_schedules
- /health
- /health
- /health
- /health
- /accounts
- /sql/api/data\_sources
- /sql/api/permissions/dashboards/2835986c-2d1d-4a7d-b26b-0fa9039aa665
- /graphql
- /graphql

30 requests 41.1 kB transferred 236 kB resources

Headers Preview Response Initiator Timing Cookies

General

Request URL: https://redash-team-dev-default.dev.databricks.com/sql/api/dashboards/2835986c-2d1d-4a7d-b26b-0fa9039aa665/refresh\_schedules

Request Method: POST

Status Code: 200

Remote Address: 44.238.187.36:443

Referrer Policy: strict-origin-when-cross-origin

Response Headers (14)

Request Headers (21)

Request Payload view source

```
{active: true, cron: "* * * * *"}
active: true
cron: "* * * * *"
```

## Creating a Refresh Schedule

To perform this with PAT tokens via REST API, replace `/sql/api` with `/api/2.0/preview/sql` to translate as follows:

```

POST /api/2.0/preview/sql/dashboards/<dashboard_id>/refresh_schedules
{
  "active": <true/false>,
  "cron": <cron schedule>
  "data_source_id": <data_source_id> (Optional - SQL endpoint refresh will use)
  "subscriptions": [
    <subscription_object_see_below>
  ]
}

```

If the data\_source\_id option is omitted, it will use the data\_source\_id associated with the underlying dashboard.

## Getting a created Refresh Schedule

To get the refresh schedule for a dashboard, do the following:

```

GET /api/2.0/preview/sql/dashboards/<dashboard_id>/refresh_schedules

```

Doing so should yield a result like follows:

```

[
  {
    "id": "<refresh_schedule_id>",
    "cron": "* * * * *",
    "active": true,
    "job_id": "<job_id_ignore_this>",
    "subscriptions": []
  }
]

```

## Modifying a Refresh Schedule

To modify the schedule just created, use the following, same payload as above POST:

```

PUT
/api/2.0/preview/sql/dashboards/<dashboard_id>/refresh_schedules/<refresh_schedule_id>
{
  "active": <true/false>,
  "cron": "<cron schedule>"
  "subscriptions": [
    <subscription_object_see_below>
  ]
}

```

```
}  
]
```

## Deleting a Refresh Schedule

To delete the schedule, use the following:

```
DELETE  
/api/2.0/preview/sql/dashboards/<dashboard_id>/refresh_schedules/<refresh_schedule_id>
```

## Subscriptions

For subscriptions, please use the following structure when POST/PUT-ing:

```
{  
  "target_type": "user",  
  "target_id": <user_id>  
}
```

OR

```
{  
  "target_type": "email",  
  "target_id": <existing_email_alert_destination_id>  
}
```

## Caveats

- Creating multiple refresh schedules per dashboard is completely unsupported
- DBSQL will not respect cron schedules that deviate from those that can be specified by the UI currently.