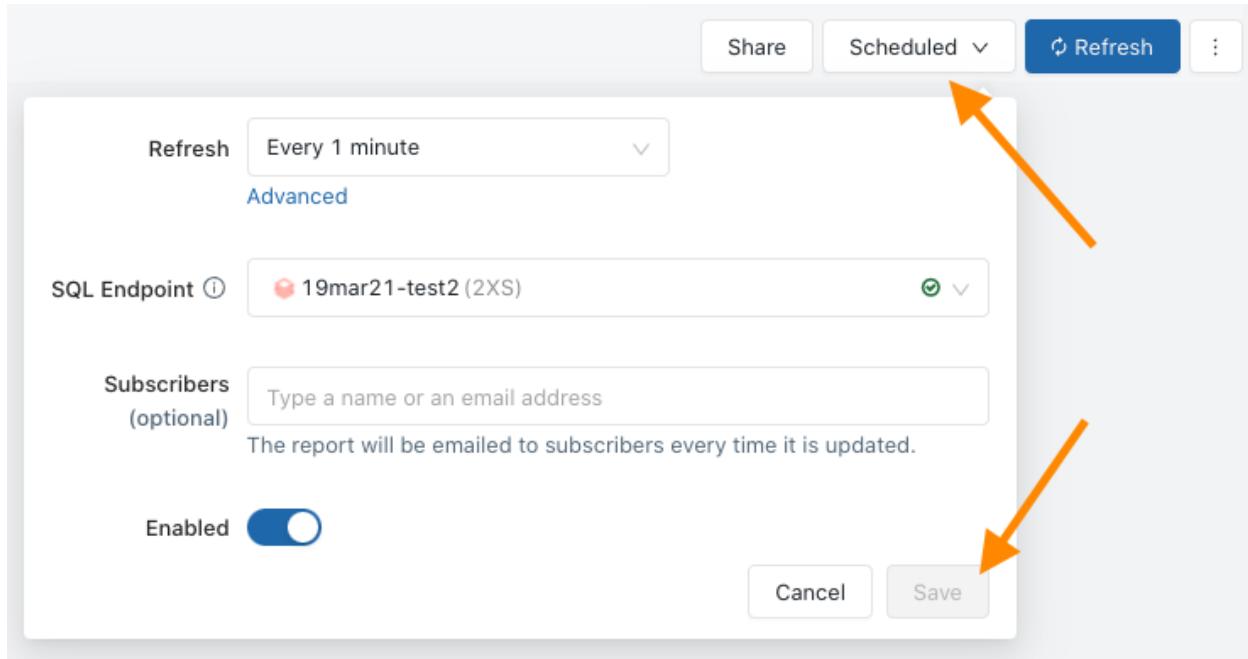


DBSQL Dashboard Refresh Schedules API

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



- Notice the requests made in Chrome Dev Tools:

The screenshot shows the Chrome Dev Tools Network tab. The left sidebar lists various API endpoints under the 'Path' heading. The right panel displays the details of a selected network request. The 'General' section shows the Request URL as `https://redash-team-dev-default.dev.databricks.com/sql/api/dashboards/2835986c-2d1d-4a7d-b26b-0fa9039aa665/refresh_schedules`, the Request Method as POST, and the Status Code as 200. The 'Request Payload' section is highlighted with a red box and contains the following JSON data:

```
{"active: true, cron: \"* * * * *\"}<br/>active: true<br/>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.