

Data Engineering with Dagster — Week 3

GraphQL [Video]

GraphQL [Video]

GraphQL

GraphQL

There is one more way to invoke a job. At the beginning of the class we talked about the origins of Dagster and mentioned that Nick Schrock helped create GraphQL at Facebook. It should not be surprising then that Dagster uses a GraphQL API. Just about everything you can do in the UI or via the command line can be done with GraphQL.

GraphQL deserves its own class and we will not spend too much time going into the GraphQL layer. But it is good to know it exists. If you ever want to play around with the API or run some example queries you can access the playground by going to the Dagit host at the `/graphql` endpoint. Using our Docker setup this would be <http://localhost:3000/graphql>.

co:rise

▶ 0:00 / 2:57



GRAPHQL USE CASES

While most of your Dagster runs might be executed either ad hoc or through a schedule/sensor to start, you can view the GraphQL layer as another way to trigger jobs. Interacting with the GraphQL API is a great way to programmatically launch jobs by systems outside of Dagster.

One example from a previous company was we backed the creation and deployment of an ML model as a Dagster job and part of our CI/CD process was to launch a run whenever a merge into main in Github occurred.