Documentation

Getting Started

What is Segment?
How Segment Works
Getting Started Guide
A Basic Segment Installation
Planning a Full Installation
A Full Segment Installation
Sending Data to Destinations
Testing and Debugging
What's Next
Use Cases

Guides

Connections

Unify

Engage

Privacy

Protocols

Segment App

API

Partners

Glossary

Config API

Help

Keep the following in mind as you set up the dbt extension:

- The extension supports dbt Core v1.7.
- You can use Snowflake, Databricks, Redshift, Postgres, and BigQuery as Reverse ETL sources.
- dbt models aren't synchronized from the dbt cloud. The model sync connects to a Git repository that loads models into Segment for use with Reverse ETL.
- You can connect to GitHub using a GitHub App, token, or SSH.
- For GitLab and Bitbucket, use SSH to connect.

Set up Git dbt Models and dbt Cloud

To set up the dbt extension, you'll need:

- an existing dbt account with a Git repository
- for job syncs, dbt cloud with jobs already created

Git repository and dbt Models setup

Follow these steps to connect the Git repository that stores your dbt Models:

1. In your Segment workspace, navigate to **Settings > Extensions**.
2. Click **Set up Git sync**.
3. On the **Configure service credentials** page, select a service and protocol, add your GitHub App, SSH private key or GitHub token, then click **Next**.
4. In the **Connect source** window, select an existing Reverse ETL warehouse source from the dropdown, then click **Save**.

After you've saved your setup, you can configure your Git repository's settings to your needs by changing the repository, branch, dbt version, default schema, and project path.

dbt Cloud setup

You can also use dbt Cloud to schedule Reverse ETL syncs after a dbt Cloud job successfully runs.

To set up dbt Cloud:

1. In your Segment workspace, navigate to **Settings > Extensions**.
2. Click **Manage dbt Cloud**.
3. Add your dbt Cloud API key or dbt Personal Access Token and an optional custom subdomain, then click **Save**.

> **Adding a custom subdomain**
>
> By default, dbt sets the subdomain to cloud. To identify your custom subdomain, open your URL and copy the portion before `.getdbt.com`. For example, if your domain was `https://subdomain.getdbt.com/`, your subdomain would be `subdomain`.

dbt Cloud Webhooks

The dbt Cloud integration allows you to schedule Reverse ETL syncs based on a dbt Cloud job. When a dbt Cloud job is selected under the Reverse ETL scheduling section, Segment creates a webhook in the dbt Cloud account that will initiate to run the Reverse ETL sync when the job is scheduled.

In order to create the webhook, ensure that you have webhook permissions associated with the dbt Cloud token in the previous step.

Model syncs

After you set up dbt, Segment runs an initial sync to load models from your connected Git repository. This initial sync lets you use the most recent models when you set up Reverse ETL. In addition to Segment's initial dbt sync, you can also trigger manual dbt model syncs.

Use a model with Reverse ETL

After you've successfully set up dbt with a warehouse and connected to your Git repository, you can select dbt models for use with Reverse ETL by following these steps:

1. In your Segment workspace, navigate to **Connections > Sources** and select the Reverse ETL tab.
2. Click **+Add Reverse ETL source** , select your source, then click **Add Model**.
3. Click **dbt Models** as your modeling method, then select and preview a model from the dbt model dropdown.
4. Add a primary key, then click **Preview your model**.
5. Click **Next**.
6. Enter your **Model Name**, then click **Create Model**.

To change a connected model, ensure that you've removed it from all active Reverse ETL syncs.

Git Connections

Git Connections enable Segment to sync data with your preferred Git repository through supported like SSH and token-based authentication.

> Git Sync and the dbt integration operate independently. You don't need to set up Git Sync to use dbt, and dbt Cloud can trigger its own syncs without relying on Git Sync.

Supported connection types

Segment supports the following credential types for setting up a Git Connection:

- **SSH**: Compatible with GitHub, GitLab, and Bitbucket, SSH provides a secure method for connecting to your repository.
- **Git token**: Git tokens are supported across GitHub, GitLab, and Bitbucket, enabling token-based authentication for added flexibility.
- **GitHub App**: For GitHub users, GitHub App integrations offer enhanced security and functionality. This method is exclusive to GitHub and supports additional features, like CI checks.

Reusing Git Connections

Segment lets you set up multiple Git Connections, allowing you to reuse credentials across both dbt and Git Sync. You can either use the same credential for multiple configurations or create separate Git Connections for each product and environment as needed.

If you plan to reuse a Git token across both dbt and Git Sync, ensure it has the necessary read and write permissions for both integrations.

Setting Up CI checks

> **CI check availability**
>
> CI checks are available only with the GitHub App connection.

CI checks in Segment help prevent breaking changes to active dbt models. Avoid changing dbt models currently in use with an active Reverse ETL sync, since changes could disrupt existing mappings and active syncs.

When CI checks are enabled, Segment monitors model changes in your Git repository. If a model already linked to an active Reverse ETL sync gets modified, Segment automatically rejects the change to maintain data integrity.

To enable CI Checks, authorize a GitHub App credential for your Git connection. Once connected, you can enable CI Checks in the dbt model sync configuration section.

Troubleshooting dbt Extensions

The following table lists common dbt Extension errors, as well as their solutions:

ERROR	ERROR MESSAGE	SOLUTION

ERROR	ERROR MESSAGE	SOLUTION
Failed sync	`Sync Failed: Incorrect dbt Project File Path: dbt project file not found`	Verify that the path to your `dbt_project.yml` file is relative to the repository root, excluding the root branch. For example, use `project/dbt_project.yml` instead of `main/project/dbt_project.yml`.
Failed sync	`Sync Failed: remote: Write access to repository not granted`	Verify that the account associated with the token has a write role in the repository settings. Fine-grained tokens may require specific roles, depending on your Git provider.

This page was last modified: 11 Dec 2024

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

Visit our Support page

Help improve these docs!

Edit this page

Request docs change

Was this page helpful?

Yes

No