3. Getting started with Dbt

Getting Started with DBT

Installation and Setup

There are two main ways to use DBT:

1. **DBT Cloud**: A web-based service that provides a development environment, job scheduler, and documentation hosting
2. **DBT Core**: The open-source command-line version that you can run locally or on your servers

Installing DBT Core

For local development, you'll need to install DBT Core:

Using pip (Python package manager):

bash

```bash
pip install dbt-core

# Install the adapter for your data warehouse
# For Snowflake:
pip install dbt-snowflake

# For BigQuery:
pip install dbt-bigquery

# For Redshift:
pip install dbt-redshift

# For PostgreSQL:
pip install dbt-postgres
```

Using Homebrew (macOS):

bash

```
brew update
brew install dbt

# Install the adapter for your data warehouse
brew install dbt-snowflake  # or another adapter
```

Setting Up a New DBT Project

Once DBT is installed, you can create a new project:

bash

```
# Create a new DBT project
dbt init my_dbt_project

# Change to the project directory
cd my_dbt_project
```

Project Structure

A typical DBT project has the following structure:

```
my_dbt_project/
├── analyses/            # Ad-hoc analyses and queries
├── dbt_project.yml      # Project configuration file
├── macros/              # Reusable SQL snippets
├── models/              # SQL models for transformations
│   ├── staging/         # Models that clean and standardize source data
│   ├── intermediate/    # Models that join and transform staging models
│   └── marts/           # Business-specific models organized by
department
├── packages.yml         # External package dependencies
├── seeds/               # CSV files for loading reference data
├── snapshots/           # Snapshot configurations for slowly changing
dimensions
└── tests/               # SQL-based tests
```

```
    ├── generic/         # Generic tests applied to multiple models
    └── singular/        # Specific tests for individual models
```

Configuration Files

DBT projects use several configuration files:

1. dbt_project.yml

This is the main project configuration file:

```yaml
name: 'my_dbt_project'
version: '1.0.0'
config-version: 2

profile: 'my_profile'

source-paths: ["models"]
analysis-paths: ["analyses"]
test-paths: ["tests"]
data-paths: ["seeds"]
macro-paths: ["macros"]
snapshot-paths: ["snapshots"]

target-path: "target"
clean-targets:
  - "target"
  - "dbt_modules"
  - "dbt_packages"

models:
  my_dbt_project:
    staging:
      materialized: view
    marts:
      materialized: table
```

2. profiles.yml

This file contains connection information for your data warehouse. It's typically located in `~/.dbt/profiles.yml`:

```yaml
my_profile:
  target: dev
  outputs:
    dev:
      type: snowflake
      account: your_account
      user: your_username
      password: your_password
      role: your_role
      database: your_database
      warehouse: your_warehouse
      schema: dev_schema
      threads: 4
    prod:
      type: snowflake
      account: your_account
      user: your_username
      password: your_password
      role: your_role
      database: your_database
      warehouse: your_warehouse
      schema: prod_schema
      threads: 8
```

3. packages.yml

This file specifies external packages that your project depends on:

yaml

```yaml
packages:
  - package: dbt-labs/dbt_utils
    version: 0.8.0
  - package: calogica/dbt_expectations
    version: 0.5.0
```