Building a Data Warehouse

09/15/2025 12:53 pm CDT

Overview

In order for a brand to have custom reporting against your service**minder** data, the data needs to be pulled out of service**minder** and loaded into a different database. Depending on how "current" the brand wants data to be, there are a couple of different approaches to implementation.

All of the methods will require some software development on the Brand side to interact with our API - to request and download the data, as well as to process and load the data into the brand database. While we can provide guidance as to best practices we've observed from other brands implementing similar reporting systems, ultimately the brand is responsible for managing its side.

For datasubscriber download examples and information click here.

If your brand doesn't have the resources for data management, consider our custom reporting solution: serviceminder|Insights. *Please speak with your Account Manager for more information.*

Periodic Snapshots

If data does not need to be continually refreshed, then a process may be followed to download all of the needed data and just replace the current data with the new data. Depending on the volume of data, as well as the paging size for downloads, paging may need to be implemented. That topic is described in more detail here.

We recommend building this as the first step in implementing a reporting database. Even if filtering is used to only get recently- modified data, there are still things that happen on the service**minder** side that may result in some of the data not being present in the downloads when filtering by Updated. These include:

- global database updates we do at your request (data cleanup, global configuration changes, etc.)
- deletes -- deletes are typically hard deletes and as a result will not show up as an "update."

So even if using filtering to load recently created/updated data, a brand should still have a mechanism to "refresh everything."

Fetching Recent Data

To get data which has been changed since the last download, use the same endpoints used for snapshots, but *now* filter by UpdatedAt. Set the range to start where the previous range ended. Using this, data may be fetched for a week or a day or even a few hours. Depending on data volume, paging may not need to be added to these downloads.

Finding About Deletes

The DataSubscriber Feed mechanism may be used to download a list of deletes. The Feed API is more event driven

-- when deletes happen in service**minder** and the DataSubscriber Feed is enabled, we'll queue those deletes so they can periodically be fetched using the processes described here.

Users must periodically clear their Feed to improve performance. Once a feed has been processed, a brand can send us the Id of the last event, and we'll clear out any events with a lower Id.

Warehouse API

Please see the below link to the API documentation that service**minder** uses to send transaction data to Warehouse. In the normal flow, as transactions are generated in service**minder**, the data will flow to Warehouse via this API.

This API is generated programmatically and will update automatically when new tables/columns are added.

Warehouse API