DataSubscriber Download API

06/29/2026 9:34 am CDT

Overview

As with all things in service**minder**, there are settings controlled by the organization and higher levels of control at the brand-level. The distinction between organization (franchisee) and brand (franchisor) allows for separation of what can be done in the system and also how that data is reported. The same holds true with the Download API.

The Data Subscriber endpoints support creating requests for data downloads and also polling for status and fetching the results. All of the formats of the downloads as well as filtering parameters match those found on the web under Dashboard > Downloads.

> **Note**: For DataSubscriber downloads created through the API, the **custom field** columns are based on the brand's **root organization**. This is the **Managing Organization**, if one is set. Otherwise, it's the organization with the lowest org ID in the brand. Only active custom fields from the root organization are included in the download.

The data will still be CSV format so you'll have to do some conversion on your side to move it into tables.

The **DataSubscriber Download API** is at the brand-level. The franchisor can set this up for all organizations at once. Only a System Admin can generate datasubscriber download keys. Please reach out to support or your Account Manager at service**minder** to start this process.

This article will review:

- Request a Download
- Poll for Status of a Download
- Use a Webhook for Download Completions
- Download a Download

Request a Download

All download requests go to the same endpoint and select the type of download using the Kind parameter. The response will include a DownloadId parameter that you will use to check status as well as fetch the results.

https://serviceminder.io/api/datasubscriber/startdownload

The following sections show examples of requesting downloads for each of the 6 supported types.

Core Properties (Common for all)

```
{
   "ApiKey": "data-subscriber-api-key",

   "CreatedFrom": "2022-5-1",
   "CreatedThrough": "2022-5-31",
   "UpdatedFrom": "2022-5-1",
   "UpdatedThrough": "2022-5-31",

   "RowId": optional-integer
}
```

The RowId parameter can be used for paging. You are limited to the number of rows that can be returned per download. The default is 25,000 records per download. To download everything, use the RowId value to page through all of the records. More details are available here.

These are example payloads for the different "kinds" of downloads. Each download kind can have different parameters and filters so they do vary slightly. You can find all of the available filters for each kind on our API documentation page (search for "datasubscriber" to skip to the relevant info. Each kind hyperlinks to its own object.)

Channels and Campaigns

```
{
   "Kind": "channels-campaigns",
   "ChannelsCampaigns":
   {
      "IncludeExcludedOrgs": true
   }
}
```

Contacts

```
{
   "Kind": "contacts",
   "Contacts":
   {
      "WithEmails": true,
      "WithoutEmails": true,
      "IncludeCustomFields": true
   }
}
```

Appointments

```
{
   "Kind": "appointments",
   "Appointments":
   {
      "Scheduled": true,
      "Completed": true,
      "Queued": true,
      "Canceled": true
   }
}
```

Payments

```
{
   "Kind": "payments",
   "DatedFrom": "2022-5-1",
   "DatedThrough": "2022-5-31"
}
```

Proposals

```
{
   "Kind": "proposals",
   "Proposals":
   {
      "IncludeBundled": true,
      "IncludeTags": false,
      "IncludeLines": true,
      "IncludeCustomFields": true
   }
}
```

Invoices

```
{
   "Kind": "invoices",
   "DatedFrom": "2022-5-1",
   "DatedThrough": "2022-5-31"
}
```

Invoice Lines

```
{
   "Kind": "invoice-lines",
   "UpdatedFrom": "2022-1-1",
   "UpdatedThrough": "2022-6-30",
   "InvoiceLines":
   {
      "Open": false,
      "Paid": true,
      "Voided": true,
      "Unapproved": false,
      "IncludeExcludedOrgs": false
   }
}
```

Services

```
{
   "ApiKey": "data-subscriber-api-key",
   "Kind": "services"
}
```

Poll for Status of a Download

Call the downloadstatus enpoint to check on the status of a download:

https://serviceminder.io/api/datasubscriber/downloadstatus

```
{
   ApiKey: "data-subscriber-api-key",
   DownloadId: "1071"
}
```

Use a Webhook for Download Completions

Instead of periodically polling the `/downloadstatus` endpoint, you can configure a web endpoint that we will notify when a download is complete. This allows you to immediately act on the download event, then check the status to confirm or go straight to downloading the resulting file.

The endpoint must:

- Be publicly reachable.
- Accept an HTTP POST request.

The payload we send is in JSON format and looks like this:

```
{
  "message": "download-completed",
  "id": <download-id>
}
```

Webhooks can be added to any existing DataSubscriber API keys and you don't need to create a new key to enable webhook support.

Why use webhooks instead of polling?

Both methods are valid. The advantage of webhooks is that you don't need to build a process that repeatedly checks all pending downloads. Instead, your webhook is triggered immediately when each download finishes.

However, note that there is **no retry mechanism** for webhooks. If a webhook call fails, you won't receive that notification. For this reason, we recommend keeping a polling process as a fallback—just at a reduced frequency. For example, you could poll every 15 minutes (instead of every minute) to catch any missed webhooks, while still benefiting from immediate notifications when webhooks succeed.

A simple analogy

- **Polling**: texting your friend every 5 minutes to ask, "Are you ready yet?"
- **Webhooks**: your friend texts you as soon as they're ready.

Download a Download

To get the results of a completed download, call the getdownload endpoint. Unlike the other endpoints, this endpoint will be the CSV file contents (so not JSON formatted).

https://serviceminder.io/api/datasubscriber/getdownload

```
{
   ApiKey: "data-subscriber-api-key",
   DownloadId: "1071"
}
```

You can review the full list of parameters and objects in our API documentation here in the DataSubscriber section.