

Sample 1: Authentication

Client: Crystallize

Context: When I first began working with the client, their documentation pages were few in number. Some had only a few sentences on them. This is an example of a page I expanded.

Process: I learned and tested authentication processes on my own, getting feedback from developers as needed. Writing the documentation with eCommerce developers as my target audience, I tried to include as much relevant information as possible: the how, the what, and the why. This document was drafted and published within Crystallize; the client used their own platform as their content management system (CMS). This was a living document in that I updated the screenshots and any other relevant information whenever Crystallize changed. Since Crystallize was based in a continuous integration/continuous development (CI/CD) environment, software updates often occurred on a weekly basis.

Tools: Crystallize, GraphQL, authentication

Result: From touching base with customers via Slack, I knew that Crystallize users greatly appreciated the more detailed documentation. I often took their feedback into consideration when prioritizing documentation improvements.

[DEV GUIDE] Authentication

Authentication in Crystallize

Authentication is the process of recognizing a user's identity. Inside Crystallize, there are some APIs and services that you must authenticate before using.

Authentication ("I am who I say I am") is distinct from authorization ("I'm allowed to do what I'm trying to do"). We have a separate page regarding [authorization in Crystallize](#).

APIs and Their Default Configuration

When you create a new tenant in Crystallize, you get a few APIs out of the box. The following APIs are open for access without authentication by default:

- [Catalogue API](#)
- [Search API](#)

The following APIs use **PIM account access key pair authentication**. You'll need to set valid X-Crystallize-Access-Token-Id and X-Crystallize-Access-Token-Secret headers on requests. Visit the [Access Tokens](#) documentation to learn more:

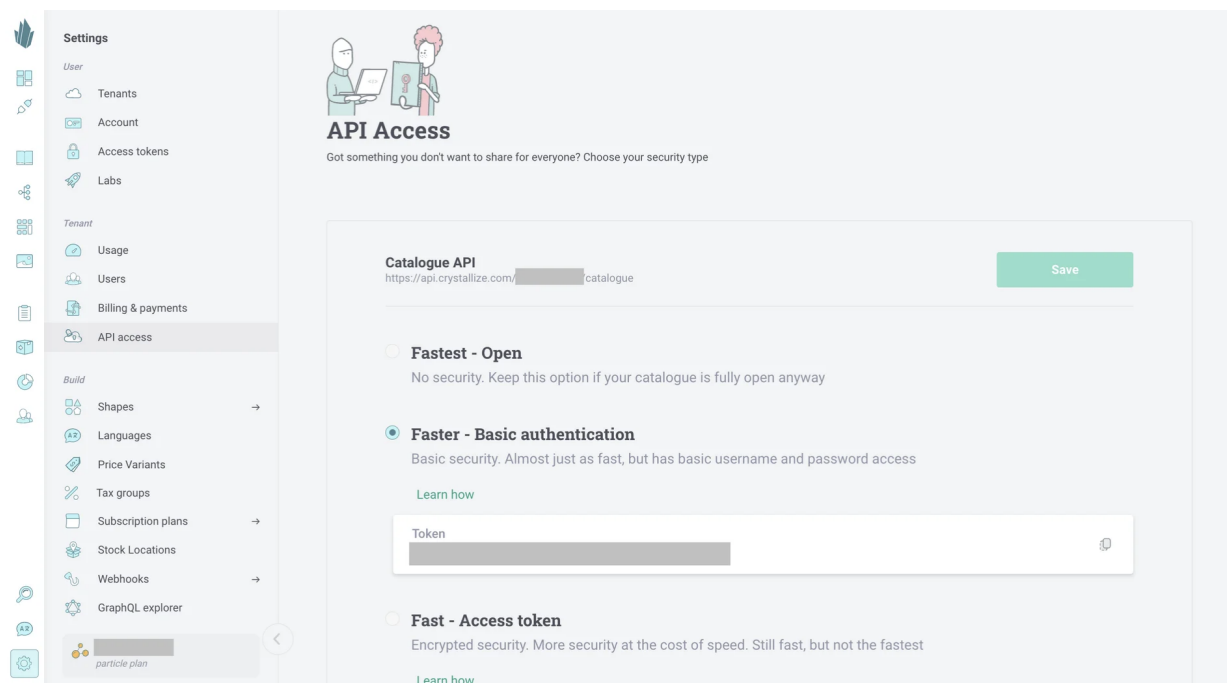
- [Core API](#)
- [Order API](#)
- [PIM API](#)
- [Shop API](#)
- [Subscription API](#)

Enabling Authentication for the Search and Catalogue APIs

In most cases, it's OK to leave the Search and Catalogue APIs open for access. But if you store sensitive information in Crystallize, you may want to limit access by enabling authentication. This can be done within the [Crystallize App](#). Click the **Settings** button on the left-hand side of the screen, then click the **API Access** label. For each API, you can choose between **Fast - Access Token** and **Faster - Basic authentication**. We recommend **Faster - Basic authentication** for two reasons:

- The API speed will not be affected much, typically only adding a few extra milliseconds to the request.
- The static token is connected with the tenant, not the PIM user, which is easier to manage in a team context.

Once you've made your selection, click the **Save** button.



Applying Authentication to API Requests

When you enable API authentication, the APIs can no longer be accessed publicly without secret access tokens that you manage. When using **Faster - Basic authentication**, you need to add a `X-Crystallize-Static-Auth-Token` header value to each request. Here's an example using the Fetch API in Javascript:

```
fetch('https://api.crystallize.digital/<tenant-identifier>/search', {
  method: 'post',
  headers: {
    'Content-Type': 'application/json',
    'X-Crystallize-Static-Auth-Token':
      'insert-your-static-token-here'
  },
  body: JSON.stringify(...)
})
```

Using Authenticated APIs from the Frontend

Once you turn on authentication for any API that previously lacked it, you cannot make fetch requests from the browser any longer. For each request to the API, you need to set the appropriate tokens. This will break any frontend that was relying on such requests to be executed without authentication.

After enabling authentication, each request needs to be proxied through an API which will forward the request to the Crystallize APIs, ensuring the proper tokens are added so that the request is authenticated. The proxy API can also intercept requests and remove any sensitive content they may contain, or apply rate limiting if you want to throttle the amount of requests going to Crystallize.

Examples for how to proxy requests can be found in the [Service API boilerplate](#), and each of our open-source [boilerplates](#) have examples of how to target the proxy API instead of hitting the Crystallize APIs directly.

Check Out Our Authentication Livestream

 [Uncovering the A's of Security: Authentication and Authorization](#)

Sample 2: Importing Data

Client: Crystallize

Context: This was a sorely-needed resource for a task most customers faced when getting started with Crystallize: they had to bring their existing eCommerce data into the platform, and had no idea what options existed or how to use them.

Process: I collaborated with a developer to test and write this document. We detailed all of the available data import options along with their advantages and drawbacks. This document was drafted and published within Crystallize; the client used their own platform as their content management system (CMS). This was a living document in that I updated the screenshots and any other relevant information whenever Crystallize changed. Since Crystallize was based in a continuous integration/continuous development (CI/CD) environment, software updates often occurred on a weekly basis.

Tools: Crystallize, GraphQL

Result: With the help of this guide, customers could flesh out their Crystallize data catalogues faster than ever.

[DEV GUIDE] Importing Data

Importing Data into Crystallize

Whether you're moving to Crystallize from another system, starting from scratch with a great deal of data to add, or copying an existing [tenant](#), Crystallize offers many flexible and time-saving methods for importing data.

Before You Begin

There are many ways to import data into Crystallize, catering to different scenarios:

- Initial (or recurring) import from an external source that may include:
 - Performing batch editing operations, e.g. adding a whole spreadsheet's worth of products to your tenant at once.
 - Integrating seamlessly between Crystallize and third-party applications that support file formats like XLS, JSON, CSV, XML, HTML, or Markdown.
 - Manual import/export.
- Automated import/export for backup, [CI/CD](#), or local development that may include:
 - Duplicating tenants for backup purposes.
 - Creating staging/development environments.

If you're performing your initial import of data, your first step should always be to build a strong [content model](#). This is to instruct Crystallize how your data will be structured. We offer a [content modeling kit](#) along with helpful advice within our [Best Practices](#) section. Once you've properly established your structure—components, relations, [shapes](#), items, and [catalogue](#)—you're ready to import data.

Requirements and Notes

- You'll need [access tokens](#) for the Crystallize tenant(s) you work with.
- You must be the owner or a Tenant Admin of the tenant(s) you work with. You can read more about [authorization in Crystallize](#) and [managing user roles and permissions](#) by following the links.
- All operations performed will count against your tenant's API call usage metric. (You can read more about [monitoring tenant usage](#) in the User Guide.)
- [Webhooks](#) and [fulfilment pipelines](#) are not included in the import-export process and need to be transferred via the [PIM API](#). They are not part of the different SDKs and transfer methods described here.

Import Using the Crystallize CLI

The [Crystallize command line interface \(CLI\)](#) is an open-source application. (You can [contribute to the CLI](#) here.) Within the CLI, you can run the following command to create a “spec” file containing the target tenant’s data:

```
npx @crystallize/cli-next@latest dump
~/my-projects/mydumpfolder tenantIdentifier
```

The CLI has many options that you can see by using the `--help` flag. For instance, you may want to add `--exclude-orders=false` and `--exclude-customers=false` to include Orders and Customers in the spec file. (This is disabled by default.)

Once you have your spec file, you can import it into a target tenant with the following command:

```
npx @crystallize/cli-next@latest import
~/my-projects/mydumpfolder/spec.json tenantIdentifier
```

Please note that if you use the CLI, there’s currently no option to exclude anything to be imported. If there’s anything you don’t want to import, you have two options:

- Remove it from the spec file (after the dump command and before the import).
- Use the [import-utilities tools](#) instead, which are described below.

Import Using the PIM API Tenant Copy

This method does everything server-side. Currently, it’s available for Crystal plan customers only, but will eventually be open to all activated accounts. (If you have any questions about your Crystallize plan, you can [contact our sales team](#).)

In the [PIM API](#), use the following mutation to trigger the cloning of a tenant:

```
mutation {
  tenant {
    createCopyTask(
      desiredIdentifier: "newidentifierformyclone"
      desiredName: "newnameformyclone"
      exclude: assets
      overwrite: true|false
      tenantId: "0987654123123"
    ) {
      id
    }
  }
}
```

With this command, you can copy an existing tenant and either create a new tenant with the data, or overwrite an existing tenant that you own by specifying `overwrite: true`. This method of copying a tenant is the fastest, and it will aggregate usage on the source tenant.

Import Using Import-Utilities

The [Crystallize import-utilities](#) are a set of tools that enable powerful import and export operations between Crystallize tenants, data specification, and conversion between a solid range of file formats. As free open-source code, you can use import-utilities in conjunction with the Crystallize APIs you already have and adjust it to support your own unique applications.

Whether you need to perform data dumps, automate backups, or pull data from Crystallize to convert and use somewhere else, you have the flexibility to do pretty much anything. See for yourself the many examples already coded for you at [our example repository](#).

You can also check out our import-utilities livestream for more information:

▶ Duplicating tenants using Crystallize utils

Import Using Individual Mutations

You could manually import data using the `product.create`, `product.update`, and all the other mutations available within the [PIM API](#). This is not recommended for most use cases, as you will miss optimization, throttling, and many things you get out of the box by using import-utilities.

Import with the Crystallize App

The [Crystallize App](#) supports [Apps](#), which are custom external applications that you can link to via a URL. Anyone may create and [add Apps to their tenant](#). Crystallize offers a number of open-source Apps as well. One of these is the [Crystallize Import App](#), a user interface to import/export data via field mapping.

Check Out Our Importing and Synchronizing Data Livestream

▶ Headless PIM with Crystallize #5 Importing & Synchronizing Product Data

Sample 3: Subscription Contracts

Client: Crystallize

Context: Creating subscriptions within the client's eCommerce enterprise solution was a process that bridged several areas of the platform (subscriptions, orders, fulfillment, webhooks). It wasn't immediately obvious what one had to do and in what order.

Process: I tested and detailed the operations that one would execute programmatically and within the Crystallize App user interface, seeking feedback from developers as needed. I included screenshots and links to other pages throughout the documentation. This document was drafted and published within Crystallize; the client used their own platform as their content management system (CMS). This was a living document in that I updated the screenshots and any other relevant information whenever Crystallize changed. Since Crystallize was based in a continuous integration/continuous development (CI/CD) environment, software updates often occurred on a weekly basis.

Tools: Crystallize, GraphQL

Result: Subscriptions are one of the most popular features of the Crystallize platform. Content like this reduces the learning curve, improving user experience and engagement.

[USER GUIDE] Subscription Contracts

Managing Subscription Contracts in the Crystallize App

[Subscription contracts](#) tie [subscription plans](#) and [customers](#) together. Each contract represents an agreement with a customer regarding what product they're buying and how they're paying for it. The contract is what you use to [keep track of usage](#) and renewals.

Creating a Subscription Contract

Subscription contracts can be created programmatically with the [Subscription API](#) or with the [Crystallize App](#). In order to create subscription contracts, you first must ensure the following:

- You have one or more [customers](#) defined within your [tenant](#).
- You've created at least one subscription plan and have attached it to at least one [product variant](#) in your tenant's [catalogue](#). See our [subscription plan](#) documentation for more information.
- You have the requisite permissions for Customers, the Tree, and Subscription Contracts (read, create, etc). Refer to our documentation on [roles and permissions](#) for more information.

Within the App, click the **Customers** button on the left-hand panel. You can also bring up the [command palette](#) with [CTRL or ⌘] + K and type or select "Customers." Select the customer you'd like to work with, either by clicking on their name within the left-hand panel or by clicking on their identifier within the spreadsheet view. With the customer record open, click the action button (...) at the top of the screen and choose **Add subscription contract**. If the customer currently has no orders or contracts, there will also be a **Subscription contract** button at the top of their record that you can click to get started.

Using the catalogue pane on the left, browse to a product that has variants with subscription plan(s) attached. Once you've found one, drag and drop it into the **Drag & drop subscription** area of the screen. Finally, select the desired period from the drop-down menu and click the **Select** button beside the subscription you wish to add.

Catalogue

Customer Håkon Krogh

Mobile subscription - metered

Subscription product

Mobility subscription - metered

Signed January 22 2025 12:00

Renewal February 22 2025 12:00

Active until February 22 2025 12:00

Customer

Persona

Identifier hakon@crystallize.com

Name Håkon Krogh

Periods

Recurring 1 month

Unit price 5

Total Price €5

SMS

Graduated

From (excluded)	To	Price
0	100	0 EUR
100	∞	0.1 EUR

Phone call minutes

Graduated

From (excluded)	To	Price
0	120	0 EUR
120	∞	0.1 EUR

Bandwidth

From (excluded)	To	Price
0	2	0 EUR

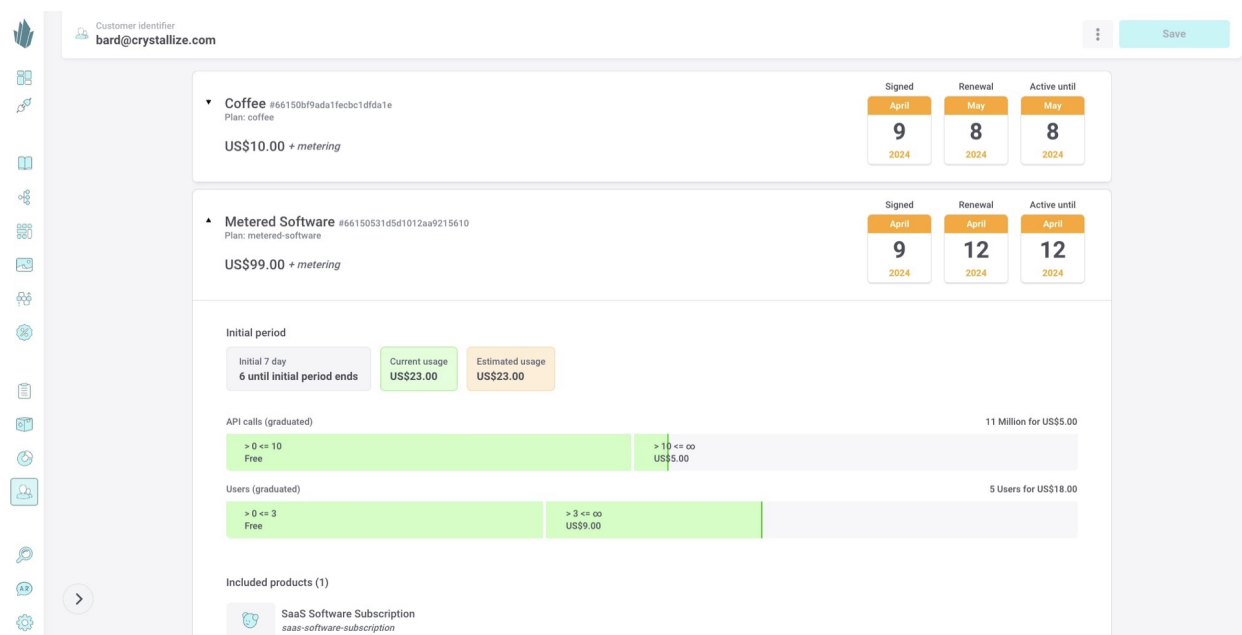
After a subscription has been selected, you can optionally adjust the following. Such adjustments will only affect this particular subscription contract and will not change the underlying subscription plan for this product variant:

- **Price variant.** Using the drop-down menu at the top right, choose one of the [price variants](#) already defined for this tenant.
- **Subscription Product.** The quantity of product variants for this contract. Adjusting this will affect the total price.
- **Signed.** The date on which the contract goes into effect. A time may also be specified in 24-hour format; if not set, it will default to 12:00 PM.
- **Renewal.** The date when the subscription contract comes up for renewal. By default, this is set to the current day plus the [subscription period](#). A time may also be specified in 24-hour format; if not set, it will default to 12:00 PM.
- **Active until.** The date when the subscription contract will expire if it's not renewed. By default, this is set to the current day plus the [subscription period](#). A time may also be specified in 24-hour format; if not set, it will default to 12:00 PM.
- **Unit Price.** You can modify the price per product variant as desired. This will affect the total price.
- **Metered Variables.** For any [metered variables](#) associated with the subscription plan, you can adjust [metered variable tier type](#), create or remove tiers, and set prices.

When finished, click the **Create** button in the top right.

Tracking Subscription Usage

Once created, the subscription contract will be visible on the Customers screen for the customer in question. Use the downward-pointing arrow on the left side of the subscription contract to access a detailed breakdown of [metered variables](#) (if applicable) and their current usage for the ongoing period. Predictive analytics provide estimates on future usage patterns and the associated total price. Leveraging data from past usage, this feature offers valuable foresight to empower informed decision-making.



Creating an Order from a Subscription Contract

Now that you have a subscription contract in place, you must create an order for it. You can either do this with the [Order API](#) or within the [Crystallize App](#). In the App, first access the appropriate customer, then click on the subscription's name to open it. From here, you can see the full details of the contract. Click the action button (...) in the top right and choose **Create Order**.

On the order creation page, the contract will be listed as an order item. You're free to add additional items and update quantities and prices as desired. You can also enter any necessary metadata as key-value pairs. Once finished, click the **Create Order** button in the top right. The order will be added to the Orders screen and any [fulfilment pipeline\(s\)](#) you may have set up for new orders. Any [webhooks](#) you've previously set up for these events will fire.

The screenshot displays the 'Catalogue' section of the Crystallize App. On the left is a vertical sidebar with various icons. The main content area is titled 'Customer Håkon Krogh' and features a 'Create order' button. Below this, the 'Order items (1)' section contains a table with the following data:

Product	Qty	Tax	Unit price	Net	Gross
Mobile... mobile-subscription-metered	- 1 +	0 %	5	€5	€5
Total				€5	€5

Below the table are buttons for 'Custom item +' and 'Open catalogue', with a note 'Drag & drop products from your catalogue'. The 'Additional information' section shows the period '24/12/2024 - 24/01/2025'. The 'Meta information' section has a button to 'Add Meta information'. On the right, the 'Customer' details are shown: Persona, Identifier (hakon@crystallize.com), and Name (Håkon Krogh).

Editing Subscription Contracts

You can use the [Subscription API](#) or [Crystallize App](#) to edit subscription contracts. Within the App, first access the appropriate customer, then click on the subscription's name to open it. From here, you can see the full details of the contract.

If you wish to manually add usage, click the plus sign button (+) beside any metered variable. Enter the desired quantity, then click **Add**.

To modify the contract itself, click the action button (...) in the top right and choose **Edit contract**. You can make changes to quantities, dates/times, prices, and tiers. Click either the **Update** button to retain your changes, or **Cancel** to revoke them.

The screenshot shows a subscription management interface for a customer named Håkon Krogh. The interface is divided into several sections:

- Subscription product:** Mobile subscription - metered. Mobile subscription. mobile-subscription-metered.
- Usage:** Includes a recurring period of 2 days until renewal, current usage, and estimated usage. It also lists usage limits for SMS (volume), Phone call minutes (graduated), and Bandwidth (graduated).
- Periods:** A table showing recurring periods and prices.
- Customer:** A sidebar showing customer details.

Signed	Renewal	Active until
September 24 2024 11:10	January 24 2025 10:10	January 24 2025 10:10

Recurring period	Unit price	Total Price
Recurring every 1 month 2 days until renewal	5	€5

Usage	Usage limit	Usage price
SMS (volume)	> 0 <= 200 Free	> 200 <= ∞ €0.1
Phone call minutes (graduated)	> 0 <= 120 Free	> 120 <= ∞ €0.1
Bandwidth (graduated)	> 0 <= 2 Free	> 2 <= ∞ €1

Customer details:

- Persona: hakon@crystallize.com
- Identifier: hakon@crystallize.com
- Name: Håkon Krogh

Canceling Subscription Contracts

You can use the [Subscription API](#) or [Crystallize App](#) to cancel subscription contracts. Within the App, first access the appropriate customer and scroll down to where their contracts are listed. Click on the subscription's name to open it. From here, you can click the action button (...) in the top right and choose **Delete contract**. This action is permanent and cannot be undone, so proceed carefully with deletion.

Subscription Contract-Related Webhooks

You can (optionally) set up [webhooks](#) to subscribe to subscription contract-related events. In Crystallize, events are fired whenever contracts are

- Created
- Updated
- Deleted
- Renewed
- Canceled

It's good practice to create a webhook for the renewed event. See our [Renewing Subscription Contracts](#) documentation to learn more.

Additionally, you should create a webhook for the canceled event where you can send out a notification to the customer that the subscription has been canceled, and run other operations that you might need to do.

For more information, read about [defining webhooks](#) here.

Check Out Our Subscription Masterclass Livestreams

 Subscription Commerce Masterclass | Episode #1: Basics of Subscription Commerce

 Subscription Commerce Masterclass | Episode #2: Usage-based & multi market subscrip...

 Subscription Commerce Masterclass | Episode #3: Subscription renewals, billing & orches...