## ## Overview

OrderCloud.io is a cloud-hosted B2B eCommerce platform exposed entirely via a RESTful API. It enables rapid development of custom, secure, and scalable B2B eCommerce solutions. Spin up a fully functional B2B app in minutes and customize it to limits of your imagination.

## ## Benefits of a RESTful API

Whether it’s the central hub of a mobile purchasing app or one small piece of a larger microservices-based solution, the benefits of a RESTful API vs. other approaches are clear.

* OrderCloud.io enables developers to use their tech stack of choice. Plus, releases are automatic and non-breaking; there’s a clear, inherent boundary between our code and yours.
* Unlike on-premise solutions, server infrastructure and data security are taken care of for you. Start small and scale up only as your business usage grows.
* Unlike SaaS, your ability to customize user interfaces and workflows is virtually limitless.
* Nearly all modern web and mobile front-end frameworks are designed to work well out of the box with JSON-over-HTTP services that follow RESTful patterns.
* OrderClould.io plays nicely with other similarly architected services. We are entering the age of [microservices](http://martinfowler.com/articles/microservices.html), where best-of-breed solutions are not built from scratch. Rather, they are assembled from the best parts. Our “part” is B2B eCommerce and Order Management, a complex domain in which we have spent 19 years building and refining an incredibly rich and flexible data model.

## How OrderCloud.io Supports Complex B2B eCommerce

## ### Buyer Organizations and Contracts

In B2B, buyer organizations matter more than in B2C. The seller typically establishes a relationship with the buyer long before the first order is ever placed. Product catalogs and pricing specific to that buyer are pre-agreed upon. Users, shipping addresses, cost centers, payment methods, and other “things” within the organization are generally configured ahead of time.

OrderCloud.io provides rich support for modeling a typical B2B buyer organization. Products can be created once and made available for purchase to any number of buyers, but details such as pricing, inventory, specs, and other configuration details can be customized per buyer according to their specific contract.

## ### Organizational Structure, Roles, and Visibility

People within the buyer organization may play different roles in the purchasing process and see different things. If one user places an order over a certain amount, it may require approval from their manager. Another user in a branch office may only see that office’s address in their shipping options. Still another may have the ability to administer users, addresses, or even products.

OrderCloud.io provides fine-grained control over specific roles and visibility. Roles and “things” belonging to the organization are *assigned* at user, group, or organization level. Groups provide a powerful and flexible way to organize users by role, department, location, or any other set of common traits. Users can belong to any number of groups, and will “inherit” roles and visibility of things based on their group memberships. This greatly simplifies administration as the number of users in the organization grows.

## ### Custom (But Fast) Buying Experience

Multi-tiered assignments of roles and other things comprise a powerful mechanism for customizing the buyer experience down to the individual user. But once configured, the shopper (and developer for that matter) wants a fast and intuitive experience that’s more concerned with *what* they see than *how* they see it.

OrderCloud.io provides a group of endpoints, commonly referred to as the **Me** routes, whose main job it is to flatten these assignments and “just show me what I can see”, regardless of how the assignments are made (user/group/organization level). The Me routes greatly simplify the process of building out the custom buying experience.

## ### Complex Products and Pricing

Products in the B2B world often come with more detailed specs, configuration options, and pricing than are typical in B2C.

OrderCloud.io provides the ability to accurately model:

* Different pricing for different buyer organizations (or different groups or users within the same organization)
* Buyer-specific variants of the same product
* Quantity based discounts
* Markups/markdowns tied to product buying options, or combinations of options that drive the user to unique SKUs
* Rules-driven promotions

### Complex Ordering Processes

In B2B, order flow isn’t always as simple as a buyer submitting an order and a seller fulfilling it. On the buyer side, the order may need to go through a multi-tier approval process. The rules around what conditions require an approval (user, order total, specific products, quantities, etc.), who is allowed to approve, and at what stage in the workflow, can get complex.

OrderCloud.io’s built-in rules engine allows you to express these rules as formulas that get evaluated each time an order is submitted or approved. When an approval triggers no further rules, the order is passed on to the seller.

## ### Assisted Ordering

Among the primary goals of any eCommerce system are automation and empowering the buyer. But in B2B, sometimes there’s no substitute for the personalized service afforded by CSRs (Customer Service Representatives) at call centers.

OrderCloud.io provides a clear method of *impersonating* a buyer user. This allows the CSR to see exactly what that specific user sees and place orders on their behalf, while protecting sensitive information such as passwords.

## ### Integrations

Integrating with ERP, CRM, and other external systems is a highly desired capability in B2B on both the buyer and seller side. The ability to automate data exchange between these systems and the commerce platform greatly reduces the costs and errors associated with manual data entry.

As a RESTful API, OrderCloud.io lends itself equally well to back-end integrations as it does front-end apps. Webhooks are a powerful mechanism that allow middleware platforms and custom server applications to subscribe to events (also via HTTP/JSON but in the reverse direction) that occur within OrderCloud.io, such as order submitted, item shipped, product changed, etc..

# ## Feature Rich API

The OrderCloud API has some standard conventions that permeate the entire platform. You can expect these features to behave the same no matter which resource you're working with.

## ### SSL

API access is only allowed via HTTPS; connections on port 80 are refused entirely. This simplifies the token-based authentication.

## ### JSON Everywhere

UTF-8 encoded JSON is the only supported data format for both request and response payloads.

## ### OAuth 2.0

OrderCloud API authentication is based on the [[OAuth 2 specification](http://tools.ietf.org/html/rfc6749)](http://tools.ietf.org/html/rfc6749) and supports multiple workflows, or in OAuth terminology, grant-types.

## ### Writable IDs

Writable IDs allow you to define the unique identifier of each resource. This capability eliminates the need to define a master record in your integration. Your integration efforts are reduced and the opportunity to avoid a mapping middleware is available. We also offer what we call an incrementor, which is used to add an atomically incremented number to your object IDs. For example, "aprefix-{myIncrementorID}", will yield an order ID of aprefix-10010.

## ### XP

We recognize that OrderCloud.io’s core data model can’t cover every conceivable scenario. Predicting every column in every table that your hypothetical database might need is impractical. That’s why we created our own schema-less solution with Extended Properties (most commonly referred to as “xp”) and exposed it on virtually every API resource. For example, our User model doesn’t define a Gender attribute, but you can easily add one if you need it:

```User.xp.Gender = ‘Male’```

And later you can get a list of all users that are males:

```GET v1/buyers/exampleBuyerID/users?xp.Gender=Male```

## ### Advanced Querying

Most list endpoints support searching, sorting, filtering, and all list endpoints support pagination.

```GET v1/products?search=foo&searchOn=Name,Description&sortBy=ID&pageSize=100&page=2```

## ### Webhooks

Webhooks allow you to subscribe to any event on any API endpoint. When a webhook is triggered, a JSON payload will be sent to the configured webhook URL. They can be used to perform follow up tasks such as updating an external CRM system when a user updates their profile, or send an email through a 3rd party system when an order is submitted. Webhooks are fundamentally designed to provide maximum flexibility within the integrated solution.

# ## Security

If facilitating billions of dollars worth of transactions over 17 years in business has taught us one thing, it’s that no shortcuts can be taken when it comes to security. We are SOC 2 type II and HIPAA Security Rule compliant, and we support the PCI compliance of applications accessing our API through appropriate security controls and by using PCI validated platform partners. User credentials are one-way encrypted such that in the unlikely event of a data breach, they would be useless. All HTTP traffic is over SSL.

Authentication is handled by an API independent of the core platform. It implements four different flows defined by the OAuth2 standard. Access to specific endpoints and functionality can be restricted at a very granular level via roles, which are aggregated to form security profiles, which can be assigned (like many other things) at the company, group, or user level.

# ## Collaboration

OrderCloud makes it incredibly easy to collaborate with other developers by adding them as contributors through the Dashboard. Adding a contributor developer is as easy as obtaining the email address they used to sign up with on DevCenter. You can even define how much access you would like them to have within any given seller organization.