## ## Introduction

This guide illustrates how to build up, pay for, and ultimately submit a very basic order.

<div class="note"><strong>04/01/2017:</strong> Order submit logic will now validate using `Payment.Accepted=true`.<br> All orders without an accepted payment will fail.</div>

## ## Prerequisites

In order to follow along with this guide, you will need to have a user who has access to a product. This requires that you:

* [Create And Assign A Product](http://documentation.ordercloud.io/use-case-guides/product-catalog-management/create-and-assign-a-product)
* [Create And Assign Price Schedule](http://documentation.ordercloud.io/use-case-guides/product-catalog-management/create-and-assign-price-schedule)
* [Make An Assignment](<http://documentation.ordercloud.io/use-case-guides/product-catalog-management/make-an-assignment>)

## 1. Create an Order

An order has no required fields upon creation. In fact, it may make the most sense to simply create a blank order and update it as information is gathered from your user, e.g., when entering a shipping address, billing info, etc.

This order creation step will define the order ID. If an ID is not provided, a random unique one will be automatically generated for you. This is true for anything being created on our platform, read more about it in the [Basic API Features](http://documentation.ordercloud.io/platform-guides/basic-api-features/conventions#WriteableIDs) guide.

<div class="api-reference">API Reference: [Create New Order](https://documentation.ordercloud.io/api-reference#Orders\_Create)</div>

```

POST https://api.ordercloud.io/v1/orders/{direction} HTTP/1.1

Authentication: Bearer put\_access\_token\_here

Content-Type: application/json

{

"ID": "…",

"FromCompanyID": "…",

"FromUserID": "…",

"BillingAddressID": "…",

"ShippingAddressID": "…",

"Comments": "…",

"ShippingCost": 0.0,

"TaxCost": 0.0,

"xp": null

}

```

## ## 2. Create Line Items

Now that we have an order we can begin creating line items as a part of that order. A line item in the most basic sense is simply a product with a quantity. You need at least one line item to complete an order but can create as many as you’d like.

Defining line items independently of an order allows us to retrieve line item information only when we need it.

<div class="api-reference">API Reference: [Create New Line Item](http://documentation.ordercloud.io/api-reference#LineItems\_Create)</div>

```

POST https://api.ordercloud.io/v1/orders/{direction}/{orderID}/lineitems HTTP/1.1

Authentication: Bearer put\_access\_token\_here

Content-Type: application/json

{

"ID": "…",

"ProductID": "…",

"Quantity": 0,

"UnitPrice": null,

"CostCenter": "…",

"DateNeeded": null,

"ShippingAccount": "…",

"ShippingAddressID": "…",

"ShipFromAddressID": "…",

"Specs": [

{

"SpecID": "…",

"OptionID": "…",

"Value": "…"

}

],

"xp": null

}

```

## ## 3. Create a Payment

Now that we have added the line items to our order, we are ready to pay. To do this, we will create a payment. Although not covered in this guide, an order can have any number of payments of any type, and a single payment can be composed of multiple transactions.

There are three payment types:

* `PurchaseOrder`
* `CreditCard`
* `SpendingAccount`

A credit card and spending account will also need to include their respective ID’s in the request body. To keep things simple we will use a payment `Type` of `PurchaseOrder` for the entire order. This means that you are leaving it up to the seller to invoice the customer to pay for the order.

<div class="note">`Amount` is optional and will autogenerate the order total minus any existing partial payments.</div>

<div class="api-reference">API Reference: [Create New Payment](https://documentation.ordercloud.io/api-reference#Payments\_Create)</div>

```

POST https://api.ordercloud.io/v1/orders/{direction}/{orderID}/payments HTTP/1.1

Authentication: Bearer put\_access\_token\_here

Content-Type: application/json

{

"ID": "RetailPurchaseOrder",

"Type": "PurchaseOrder",

"Description": null,

"Amount": 19.99,

"xp": null

}

```

## ## 4. Submit Order

The final step is to submit an order. Submitting an order will simply change the IsSubmitted value on an order to true, allowing us to ship the order. For more complex scenarios you may have an approval process that automatically triggers this call once a predetermined set of conditions is true.

<div class="api-reference">API Reference: [Submit Order](https://documentation.ordercloud.io/api-reference#Orders\_Submit)</div>

```

POST https://api.ordercloud.io/v1/orders/{direction}/{orderID}/submit HTTP/1.1

Authentication: Bearer put\_access\_token\_here

Content-Type: application/json

```

Unlike the other calls we made, this one will not require a request body.

# ## Conclusion

Congratulations! You have successfully submitted a very basic order. You should now have an understanding of requirements for submitting an order and hopefully had a chance to see where you can customize the ordering experience to fit your needs. Continue reading our guides to gain a deeper understanding.