## ## Introduction

This guide will teach you how to create a price schedule. A Price Schedule can be thought of as a framework for how pricing for a product will be determined rather than a static price (although pricing can be static if defined as such).

With price schedules, you can do the following:

* Create price breaks based on quantity ordered
* Set pricing based on single line-items or combined line-items
* Enforce maximum and minimum order volume restrictions
* Restrict buyers to order only specific volumes of products

<div class="note">Oftentimes, businesses will offer a price break to their customers if they order a certain quantity of a product. For example, each unit may cost $10 for any quantity set between 1 and 10 units, $8 for any quantity set between 11 and 20 units, and $7 thereafter. To learn more about Price Breaks, check out the [Create Price Breaks](https://documentation.ordercloud.io/use-case-guides/product-catalog-management/create-price-breaks) guide.</div>

## ## Prerequisites

* [Create And Assign Users And User Groups](https://documentation.ordercloud.io/use-case-guides/buyer-and-seller-organization-management/create-and-assign-users-and-user-groups)
* [Make an Assignment](https://documentation.ordercloud.io/use-case-guides/buyer-and-seller-organization-management/make-an-assignment)

## ## 1. Create a Price Schedule

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

```

POST https://api.ordercloud.io/v1/priceschedules HTTP/1.1  
Authentication: Bearer put\_access\_token\_here  
Content-Type: application/json

{

"ID": "PriceScheduleX",

"Name": "Price Schedule X",

"ApplyTax": false,

"ApplyShipping": false,

"MaxQuantity": null,

"UseCumulativeQuantity": false,

"RestrictedQuantity": false,

"OrderType": "Standard",

"PriceBreaks": [

{

"Quantity": 1,

"Price": 10.0

}

],

"xp": null

}

```

`ApplyTax` This parameter helps your integration middleware determine whether or not tax should be calculated. Set this to `true` if taxes should be calculated and `false` if they shouldn’t.

`ApplyShipping` Similar to `ApplyTax`. Is used by integration middleware to determine whether or not an order should apply shipping costs.

`UseCumulativeQuantity` By default, minimum and maximum quantities take effect at the line item level, meaning the user is restricted to a particular quantity range per line item. However, UseCumulativeQuantity can be set on the Price Schedule in order to limit the user to specific quantities across the entire order.

`RestrictedQuantity` If set to true, the user will be restricted to ordering quantities in preconfigured price breaks as defined on our price schedule.

`OrderType` This defines which type of order (Standard or Replenishment), the price schedule will apply to.

<div class="note">It should be noted that a Price Schedule with only one price break will indicate the unit price regardless of the quantity:</div>

```

"PriceBreaks": [

{

"Quantity": 1,

"Price": 10.0

}

]

```

\*is the same as:\*

```

"PriceBreaks": [

{

"Quantity": 5,

"Price": 10.0

}

]

```

Only when another price break is added does quantity have meaning. For example, the following price break definition says that any quantity up to 5 is $10, and $5 thereafter:

```

"PriceBreaks": [

{

"Quantity": 1,

"Price": 10.0

},

{

"Quantity": 5,

"Price": 5.0

}

]

```

## ## 2. Save a Price Break to a Price Schedule

You also have the option to save a price break to a price schedule after the price schedule has been created.

<div class="api-reference">API Reference: [Save Price Break](https://documentation.ordercloud.io/api-reference#PriceSchedules\_SavePriceBreak)</div>

```

POST [https://api.ordercloud.io/v1/](https://api.ordercloud.io/v1/buyers/%7BbuyerID%7D/orders)priceschedules/{priceScheduleID}/PriceBreaks HTTP/1.1  
Authentication: Bearer put\_access\_token\_here  
Content-Type: application/json

{

"Quantity": 10,

"Price": 3.0

}

```

## 3. Assign A Price Schedule To A Product and Party

Price schedules can be saved on a user, user group, and buyer level.

<div class="api-reference">API Reference: [Save Assignment](https://documentation.ordercloud.io/api-reference#Products\_SaveAssignment)</div>

```

POST https://api.ordercloud.io/v1/products/assignments HTTP/1.1  
Authentication: Bearer put\_access\_token\_here  
Content-Type: application/json

{

"ProductID": "Product1",

"PriceScheduleID": "PriceScheduleX",

"BuyerID": "BuyerOrg",

"UserID": null,

"UserGroupID": null

}

```

## Conclusion

Congratulations! You should now be able to create a price schedule, save a price break, and assign a price schedule to a product and a party.