COVA™

CANNABIS RETAIL SOFTWARE

Product Pricing

Explanations and examples

2024

Cova Product Pricing

Product Pricing within Cova is very flexible. Prices can be as simple as a regular price that is consistent company wide, or as complex as tiered shelf price for a specific pricing group at a specific location that is on sale for an hour on a particular day. This document aims to define these terms as they relate to Cova and demonstrate scenarios and usage that are common across retailers.

Definitions

Company: A business, firm, or corporation that may operate one or many locations as physical brick-and-mortar stores and/or virtual ecommerce presence. Most calls to Cova APIs require a *CompanyId*, which is included in every integration onboarding package we supply.

Location: A physical store or dispensary customers can visit to purchase cannabis and accessories. A company may have many locations. Each location has its own inventory and may define its own prices. Many Cova API endpoints require a location id, sometimes represented as *EntityId*.

Room: Cova supports the concept of rooms. This is a way to segregate inventory within a location. A typical setup would consist of a “Back Room” where new stock is unboxed and received into the system, and a “Sales Floor” used to fulfill sales to end customers. Note: a location can have any number of rooms, but only one room can be designated for processing sales. Rooms do not support different pricing.

Groups, Divisions, and Entity: A company can define a structure within Cova. We often refer to this as the Company Tree. The tree may have many branches of Groups and/or Divisions. Figure 1 below shows a Company (COVA Café), with a Division (Canada), and many Groups (Alberta, Ontario, etc.). Locations can exist at any level within the tree. The tree is dynamic and can have any number of Divisions and Groups nested within it, or none. An Entity is any node along the tree. “Canada” is an entity, “Ontario” is an entity, and “Hamilton” is also an entity which we commonly refer to as a Location. Only Location entities can have inventory and process sales.

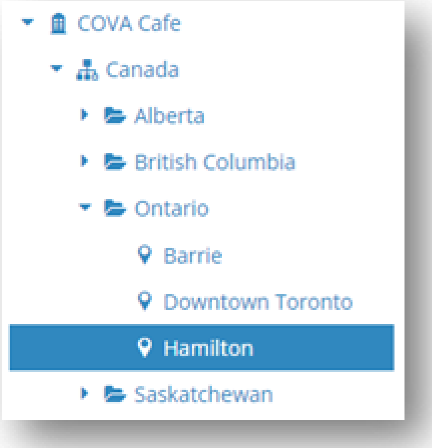
Tier: A Tier is a pre-defined limit at which a particular price becomes valid. For example, a pre-roll flower item may sell for $4 each and have a pricing tier at a quantity of 3 with a price of $10 (i.e. 3 for $10). When the customer chooses to buy three of these items, the total for the line would be $10 instead of $12 if purchased on a separate transaction. Tiered prices can be weight-based or unit-based.

Pricing Group: A retailer may choose to give specific groups of customers a special price on some items. Popular choices may include Veterans and Employees, but these are optional and maintained by the retailer. These prices can be considered a Sale Price, not a Promotion or Discount.

Shelf: A shelf is a way to group products and have them share pricing. A simple example is “Top Shelf” for excellent quality pre-rolled cannabis products that may sell for, say, $5 each. Shelf pricing is commonly used in conjunction with tiered pricing. Combining examples, the “Top Shelf” may have a tier at quantity 3 for $10. If a customer purchases any three items from the Top Shelf, the price is $10.

Sale Pricing, Promotion, Discount: Sale Pricing is used to apply a reduced price to an item. Promotions and Discounts differ in that they are recorded, maintained, and displayed separately, and are not reflected in the Pricing APIs or pricing setup in Hub.

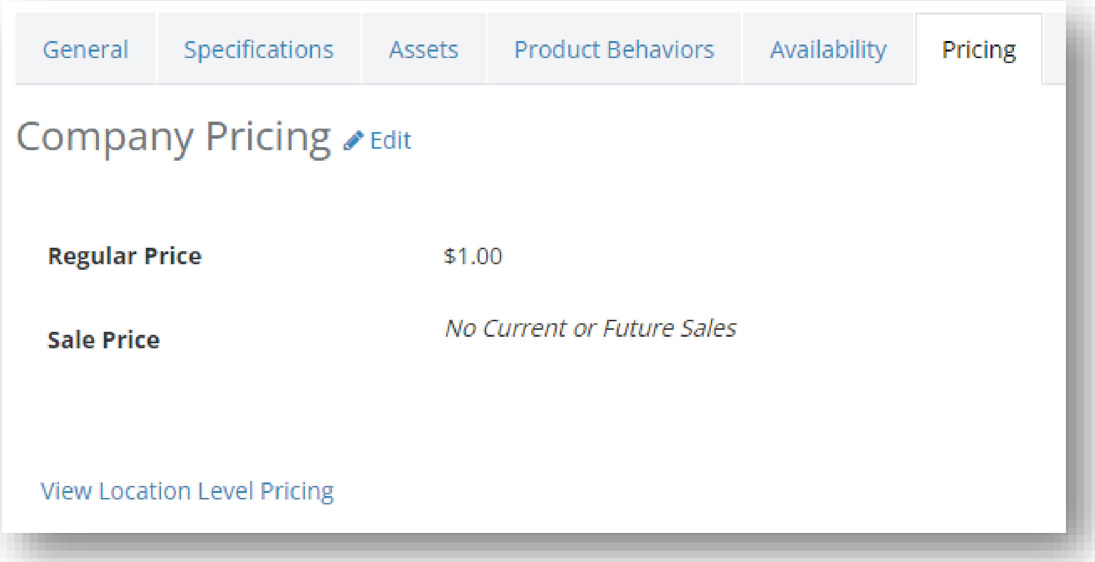
Figure 1 Company Tree example



Regular Price

The simplest case is a product with only a regular price, defined at the company level:

Figure 2: Cova hub, pricing tab for a product

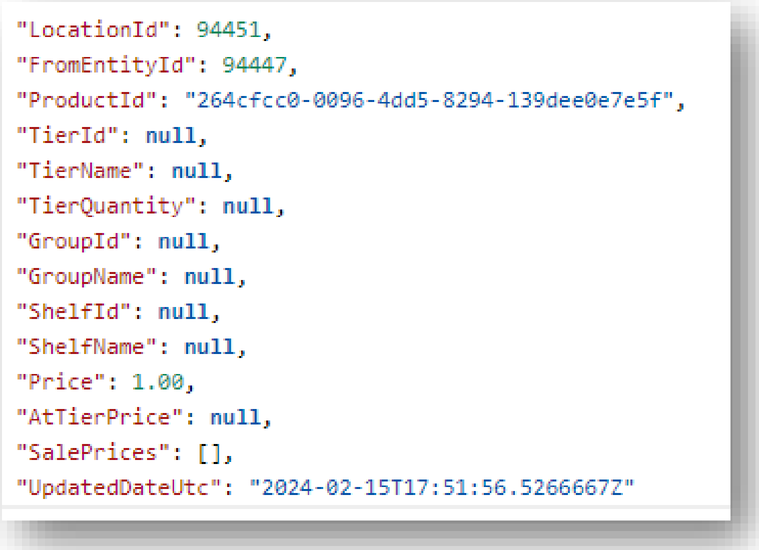


Company Pricing

Regular Price	$1.00
Sale Price	*No Current or Future Sales*

The API response when requesting pricing for this item is represented in Figure 3. The *FromEntityId* returned reflects the company tree node where applicable pricing is set. *ProductId* is the product identifier. Every other field is null but the *Price*.

Figure 3: Single ProductPrice resource



```
"LocationId": 94451,
"FromEntityId": 94447,
"ProductId": "264cfcc0-0096-4dd5-8294-139dee0e7e5f",
"TierId": null,
"TierName": null,
"TierQuantity": null,
"GroupId": null,
"GroupName": null,
"ShelfId": null,
"ShelfName": null,
"Price": 1.00,
"AtTierPrice": null,
"SalePrices": [],
"UpdatedDateUtc": "2024-02-15T17:51:56.5266667Z"
```

Product Prices and Inheritance

Figure 3 shows the simplest case for a pricing record. Let us assume an API call was made where *CompanyId* is 94447 and *LocationId* is 94451. Notice that the *FromEntityId* returned is the same as the *CompanyId*.

Pricing can be established at any point in the company tree. The pricing service will look for a price at the *LocationId* then proceed up the tree until it finds a price. The *FromEntityId* returned in the above response is the lowest point in the tree where a price was set. Using the company tree Figure 1, assume the *LocationId* we requested pricing for was Hamilton. A price was not set there, so the service examines the Ontario node, then the Canada node, and finally the COVA Café node where a price of $1 has been set.

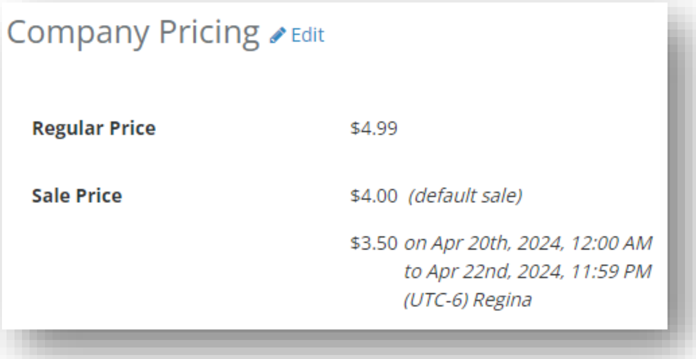
Consider the following scenario:

Currently, the price is set at the company level (all locations) for $1. The District Manager of Ontario may choose to set the price of the item in Figure 3 at $3 by entering a price at the Ontario node in the tree. The store manager at Hamilton may choose to sell this product at a different price, say $2.50, by entering a location level price. Requesting Hamilton prices (by its *LocationId*) you would receive $2.50. Any other store in the Ontario group would return $3, and every other store in the company would return $1.

Sale Prices

Retailers can define multiple sale prices for items. If a sale has no start/end dates it is considered a default sale, which is valid if no other sale prices apply by date. Only one default sale price is allowed, and a default sale is not required. Any number of dated sale prices are allowed. The sale dates cannot overlap.

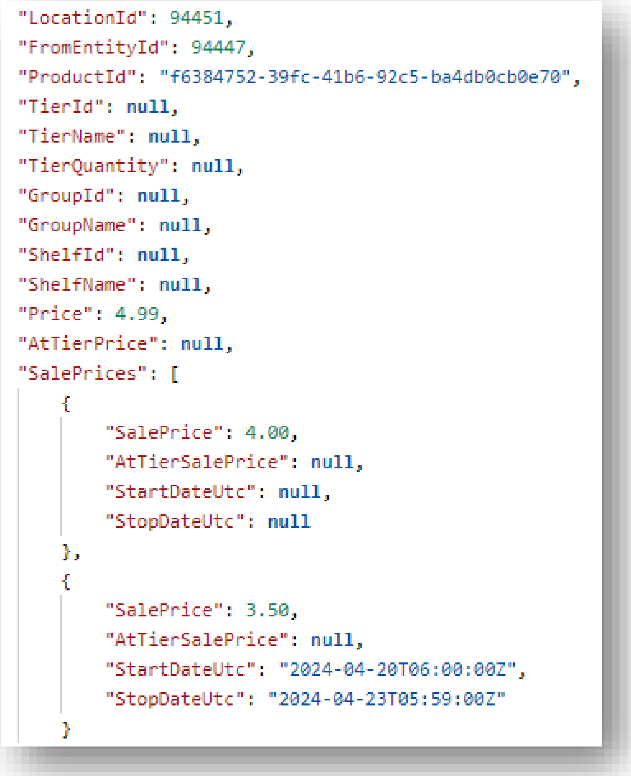
Figure 4: Sale pricing



Company Pricing

Regular Price	$4.99
Sale Price	$4.00 *(default sale)*
	$3.50 *on Apr 20th, 2024, 12:00 AM to Apr 22nd, 2024, 11:59 PM (UTC-6) Regina*

Figure 5: Sale Pricing example JSON

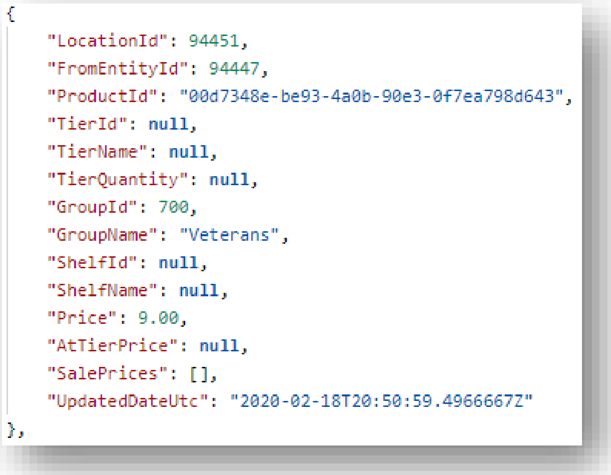


```
"LocationId": 94451,
"FromEntityId": 94447,
"ProductId": "f6384752-39fc-41b6-92c5-ba4db0cb0e70",
"TierId": null,
"TierName": null,
"TierQuantity": null,
"GroupId": null,
"GroupName": null,
"ShelfId": null,
"ShelfName": null,
"Price": 4.99,
"AtTierPrice": null,
"SalePrices": [
    {
        "SalePrice": 4.00,
        "AtTierSalePrice": null,
        "StartDateUtc": null,
        "StopDateUtc": null
    },
    {
        "SalePrice": 3.50,
        "AtTierSalePrice": null,
        "StartDateUtc": "2024-04-20T06:00:00Z",
        "StopDateUtc": "2024-04-23T05:59:00Z"
    }
```

Pricing Group

Pricing group is defined at the beginning of this document. Figure 6 shows a price for the Veterans pricing group. If a Cova customer belongs to a pricing group, the *PricingGroupId* property will be populated on their customer profile.

Figure 6: Pricing group price



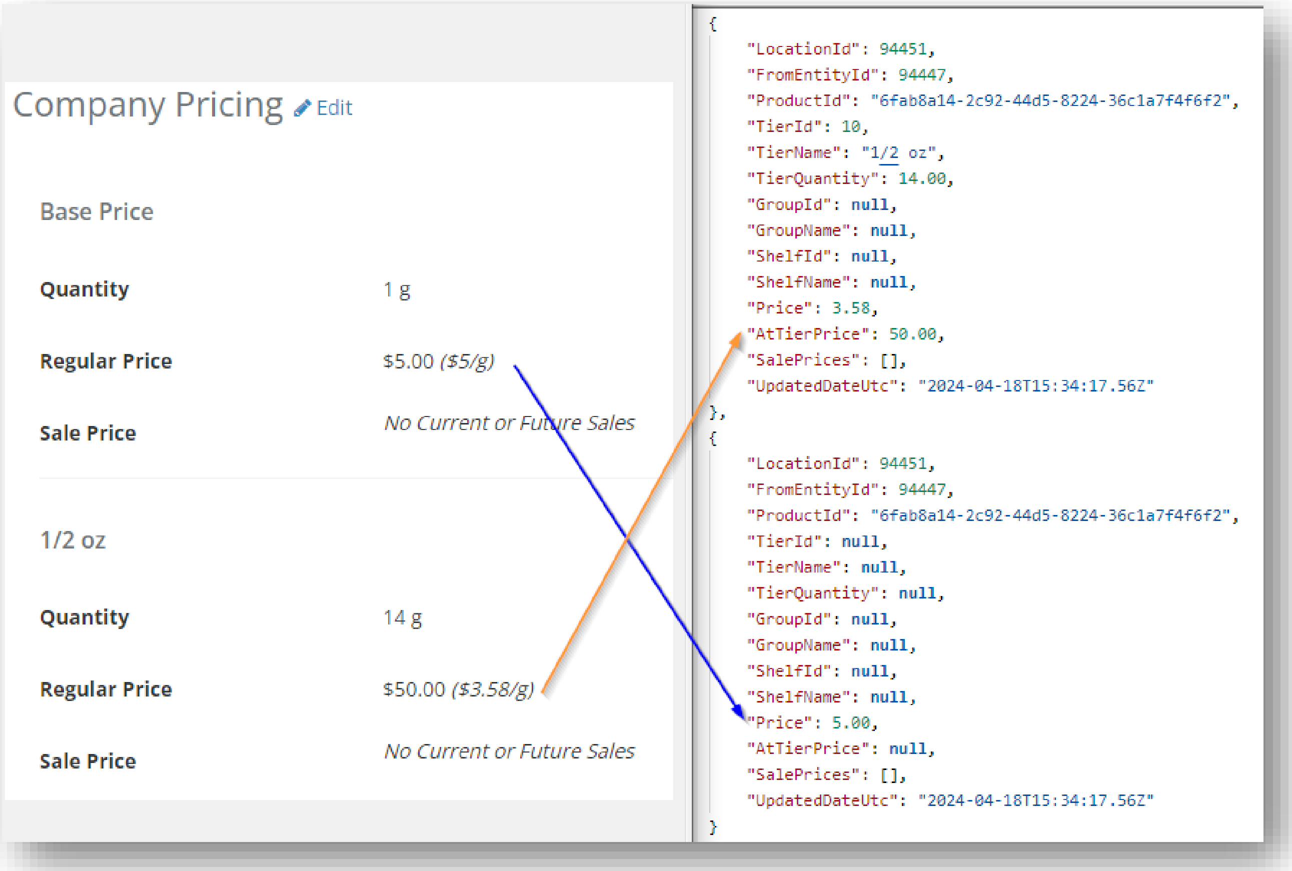
```
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "00d7348e-be93-4a0b-90e3-0f7ea798d643",
    "TierId": null,
    "TierName": null,
    "TierQuantity": null,
    "GroupId": 700,
    "GroupName": "Veterans",
    "ShelfId": null,
    "ShelfName": null,
    "Price": 9.00,
    "AtTierPrice": null,
    "SalePrices": [],
    "UpdatedDateUtc": "2020-02-18T20:50:59.4966667Z"
},
```

Tiered Price

Tier is defined at the start of this document. Figure 7 shows a weight-based product that has base pricing and a price for one tier; ½ oz. Left side of the image is from Cova Hub, right side is the JSON representation. If a consumer purchases any quantity under 14.0g (*TierQuantity*) the price is $5 per gram. Purchasing 14g, the price is $50.00, and purchasing more than 14g requires some calculation, which is detailed under "*Price* vs *AtTierPrice*".

The company may define tiers at, say, 3.5g, 5g, 7g, etc., but a product with tiered pricing is not required to supply a price for each tier.

Figure 7: Tier price example



Company Pricing Edit

Base Price

Quantity	1 g
Regular Price	$5.00 *($5/g)*
Sale Price	*No Current or Future Sales*

1/2 oz

Quantity	14 g
Regular Price	$50.00 *($3.58/g)*
Sale Price	*No Current or Future Sales*

```
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "6fab8a14-2c92-44d5-8224-36c1a7f4f6f2",
    "TierId": 10,
    "TierName": "1/2 oz",
    "TierQuantity": 14.00,
    "GroupId": null,
    "GroupName": null,
    "ShelfId": null,
    "ShelfName": null,
    "Price": 3.58,
    "AtTierPrice": 50.00,
    "SalePrices": [],
    "UpdatedDateUtc": "2024-04-18T15:34:17.56Z"
},
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "6fab8a14-2c92-44d5-8224-36c1a7f4f6f2",
    "TierId": null,
    "TierName": null,
    "TierQuantity": null,
    "GroupId": null,
    "GroupName": null,
    "ShelfId": null,
    "ShelfName": null,
    "Price": 5.00,
    "AtTierPrice": null,
    "SalePrices": [],
    "UpdatedDateUtc": "2024-04-18T15:34:17.56Z"
}
```

Price vs AtTierPrice

For tiered prices use the *AtTierPrice* instead of *Price*, as calculations using *AtTierPrice* will be more accurate. In the above example, the retailer wants to sell the item at $50 for ½ oz. This works out to $3.58 per gram, rounded up.

Consider the following calculations based on Figure 8.

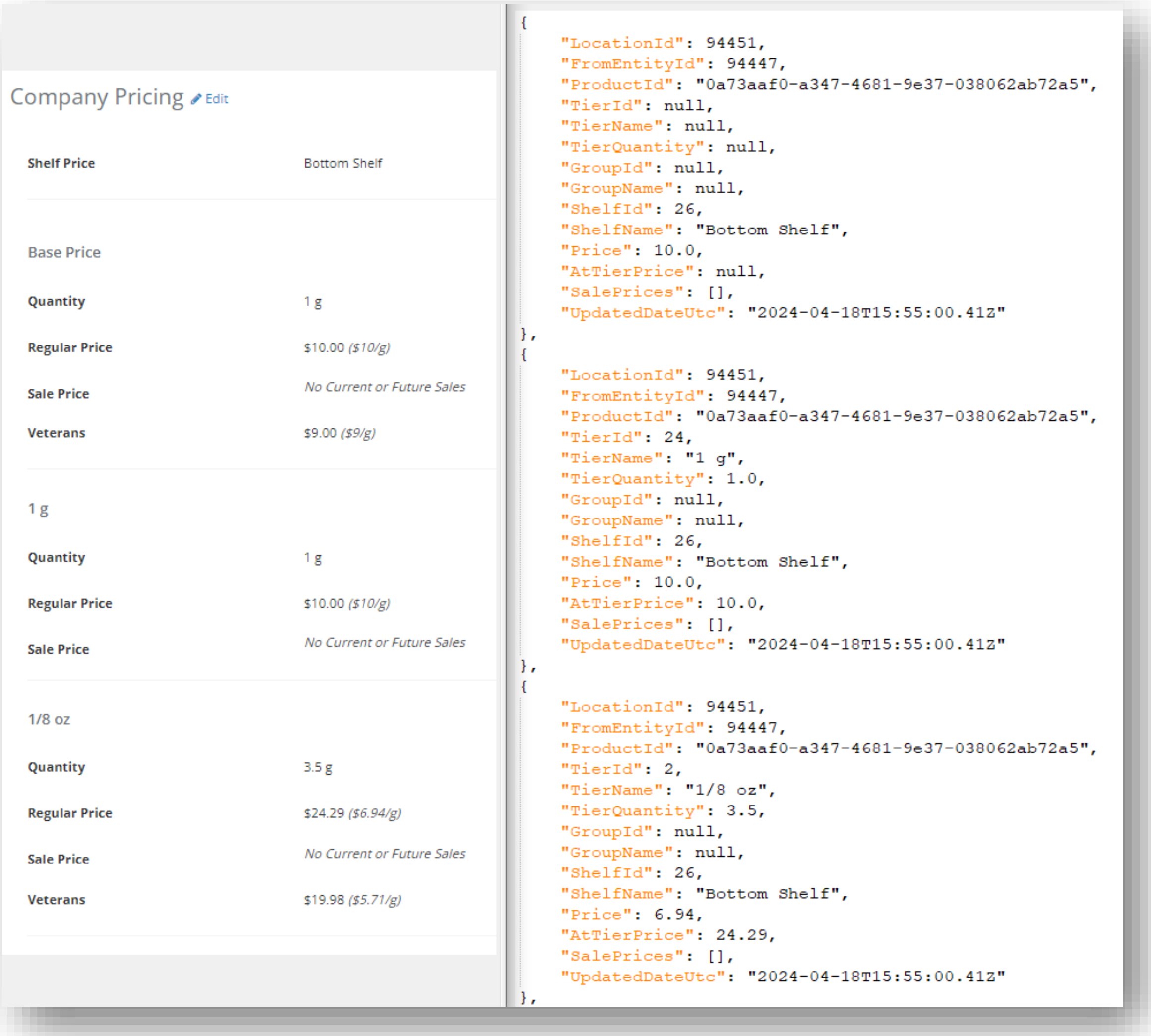
Purchase of 14g
Using *Price*: 3.58 * 14 = $50.12 (reg price * qty)
Using *AtTierPrice*: (50 / 14) * 14 = $50 ([*atTierPrice* / *TierQuantity*] * qty)

Purchase of 20g:
Using *Price*: 3.58 * 20 = $71.60 (reg price * qty)
Using *AtTierPrice*: (50 / 14) * 20 = $71.43 ([*atTierPrice* / *TierQuantity*] * qty)

Shelf Price

A shelf is defined at the start of this document. A shelf price is not drastically different than any other price described to this point. The main thing to consider is that all catalog items belonging to a shelf should be grouped together (by *ShelfId*) if you need to consider quantity. For example, in Figure 8, catalog items on the "Bottom Shelf" are also tiered. Assuming Products "A" and "B" are on the "Bottom Shelf", a consumer purchasing 2g of "A" and 1.5g of "B" would be purchasing 3.5g total from that shelf and will qualify for the 1/8oz price of $24.29. Figure 8 shows Cova Hub on the left and the JSON data on the right. Again, note the "*Price* vs *AtTierPrice*" info above.

Figure 8: Shelf price item with tiers



Company Pricing ✎ Edit

Shelf Price	Bottom Shelf
Base Price	
Quantity	1 g
Regular Price	$10.00 *($10/g)*
Sale Price	*No Current or Future Sales*
Veterans	$9.00 *($9/g)*
1 g	
Quantity	1 g
Regular Price	$10.00 *($10/g)*
Sale Price	*No Current or Future Sales*
1/8 oz	
Quantity	3.5 g
Regular Price	$24.29 *($6.94/g)*
Sale Price	*No Current or Future Sales*
Veterans	$19.98 *($5.71/g)*

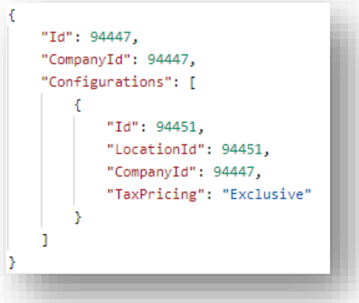
```
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "0a73aaf0-a347-4681-9e37-038062ab72a5",
    "TierId": null,
    "TierName": null,
    "TierQuantity": null,
    "GroupId": null,
    "GroupName": null,
    "ShelfId": 26,
    "ShelfName": "Bottom Shelf",
    "Price": 10.0,
    "AtTierPrice": null,
    "SalePrices": [],
    "UpdatedDateUtc": "2024-04-18T15:55:00.41Z"
},
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "0a73aaf0-a347-4681-9e37-038062ab72a5",
    "TierId": 24,
    "TierName": "1 g",
    "TierQuantity": 1.0,
    "GroupId": null,
    "GroupName": null,
    "ShelfId": 26,
    "ShelfName": "Bottom Shelf",
    "Price": 10.0,
    "AtTierPrice": 10.0,
    "SalePrices": [],
    "UpdatedDateUtc": "2024-04-18T15:55:00.41Z"
},
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "0a73aaf0-a347-4681-9e37-038062ab72a5",
    "TierId": 2,
    "TierName": "1/8 oz",
    "TierQuantity": 3.5,
    "GroupId": null,
    "GroupName": null,
    "ShelfId": 26,
    "ShelfName": "Bottom Shelf",
    "Price": 6.94,
    "AtTierPrice": 24.29,
    "SalePrices": [],
    "UpdatedDateUtc": "2024-04-18T15:55:00.41Z"
},
```

Tax Inclusive Pricing (TIP)

This scheme of pricing allows retailers to set a price for items that will include all applicable taxes. Also known as out-the-door pricing. The pricing APIs have no concept of which pricing scheme is represented in the pricing. TIP is configured at the location level, so it is possible to have a multi-location company where individual locations use TIP, while others do not. That also means those locations would have specific pricing to account for the taxes. The API call to get tax pricing configurations is described below. Default is "Exclusive" (taxes not included in the price)

```
GET taxes/v1/Companies({{CompanyId}})/TaxPricingConfiguration
```

Figure 9 tax pricing configurations



```
{
    "Id": 94447,
    "CompanyId": 94447,
    "Configurations": [
        {
            "Id": 94451,
            "LocationId": 94451,
            "CompanyId": 94447,
            "TaxPricing": "Exclusive"
        }
    ]
}
```

In the example above, location 94451 is configured with tax "Exclusive" pricing, which means taxes will be calculated and added to each product before completing the sale. The only other valid result is "Inclusive", where taxes are included in the prices.

This configuration will rarely change. It is not trivial to switch between the tax schemes as all pricing for a location must be properly adjusted. Cova will communicate to each concerned integrator before a location configuration is to be changed. Refreshing a cache of this data once daily is sufficient (do not call before creating each order). Figure 9 shows the result at the company level (all configurations). If a location is *not* listed in this configuration, it is by default a TEP (exclusive) location.

API Calls

Initial synchronization and deltas/changes/updates

A common task after establishing a relationship with a Cova retailer will be to synchronize all relevant product data to your system, then check for changed data on an ongoing basis. We offer an aggregated view of product data including product properties, specifications, stock levels, pricing, and taxes via the DataPlatform APIs. See DataPlatform at api.covasoft.net/Documentation

Targeted Pricing APIs

Synchronizing pricing data is best accomplished using the DataPlatform APIs mentioned previously. However, we also offer API endpoints that target pricing data specifically. In the following API endpoint examples, values enclosed in double curly brackets represent a variable that must be supplied. i.e. {{*CompanyId*}}. Many bulk endpoints support/require paging in the form of skip/top, where $skip refers to the number of resources to skip before returning the number of resources declared in the $top value.

All changed prices

This call is intended to be polled throughout the day to ensure your pricing information is consistent. The data returned will be all current prices for any products where the last pricing update is greater than or equal to the *asOfDateUtc*. We recommend requesting this endpoint once per minute with the datetime five minutes in the past. This will result in pricing records that have changed in the last 5 minutes.

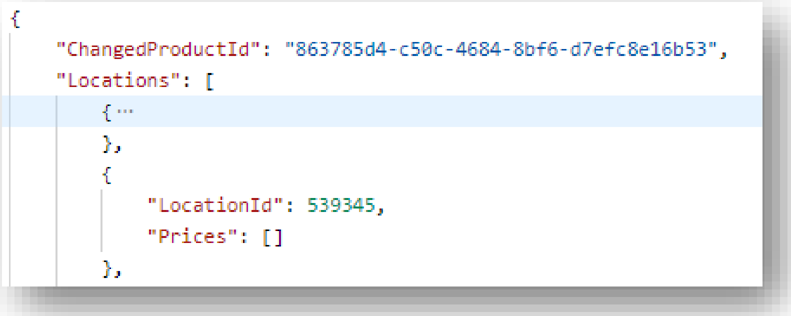
- For all locations within a company:
  `/pricing/v1/Companies/{{CompanyId}}/ChangedPrices/{{asOfDateUtc}}?skip=0&top=500`

- For a specific location:
  `/pricing/v1/Companies/{{CompanyId}}/Locations/{{LocationId}}/ChangedPrices/{{asOfDateUtc}}?skip=0&top=500`

Result: *ChangedProductPrice*[]

Note: in the context of *ChangedPrices*, an empty Prices collection means there are no prices set at the location for the productid. Figure 10 shows this scenario.

Figure 10 - No pricing at a location for a product



```
{
    "ChangedProductId": "863785d4-c50c-4684-8bf6-d7efc8e16b53",
    "Locations": [
        {…
        },
        {
            "LocationId": 539345,
            "Prices": []
        },
```

Resource and property definitions

PriceRecord

Property	Type	Notes
LocationId	Int	Cova location identifier
FromEntityId	Int	Entity in the company tree this price is inherited from
ProductId	Guid	Product identifier (a.k.a. CatalogItemId)
TierId	Int	Tier identifier (nullable)
TierName	String	Tier name (nullable)
TierQuantity	Decimal	Lower bound of the tier (nullable)
GroupId	Int	Pricing group identifier (nullable)
GroupName	String	Pricing group name (nullable)
ShelfId	Int	Pricing shelf identifier (nullable)
ShelfName	String	Pricing shelf name (nullable)
Price	Decimal	The Price
AtTierPrice	Decimal	Price at the lower bound quantity for the tier
SalePrices	SalePrices[]	
UpdatedDateUtc	Datetime	Last update to this price

SalePrices object

Property	Type	Notes
Id	integer	
StartDateUtc	datetime	
StopDateUtc	datetime	
SalePrice	decimal	
AtTierSalePrice	decimal	Sale price at the lower bound quantity for the tier

ChangedProductPrice

Property	Type	Notes
ChangedProductId	guid	Product identifier (a.k.a CatalogItemId)
Locations	LocationPrice[]	Collection of location prices

LocationPrice

Property	Type	Notes
LocationId	Int	Identifier of the Cova location (a.k.a entityId)
Prices	PriceRecord[]	Current price information for this location

TaxPricingConfiguration

Property	Type	Notes
Id	Int	

CompanyId	Int	
Configurations	collection	See TaxPricingConfigurations

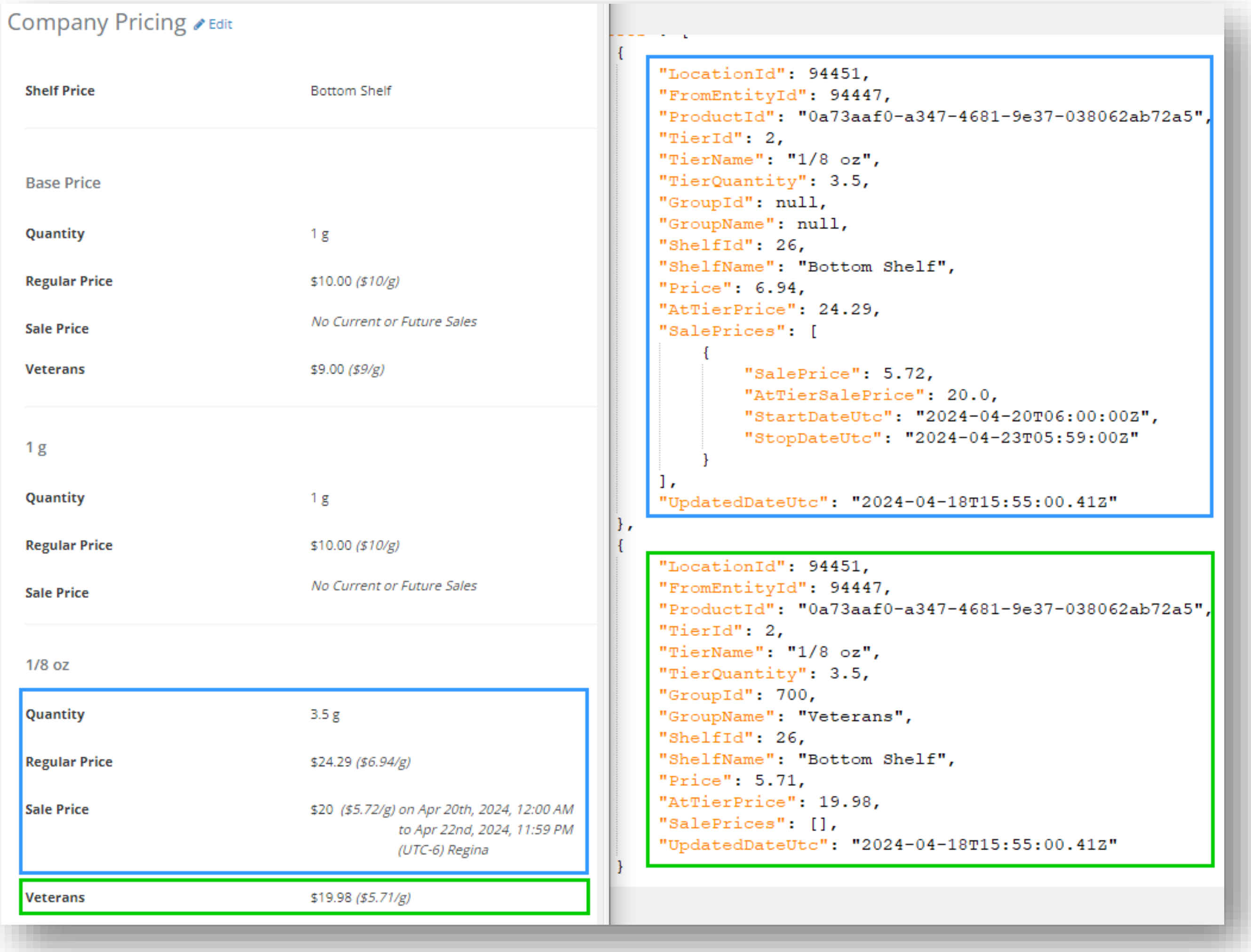
TaxPricingConfigurations

Property	Type	Notes
Id	Int	
CompanyId	Int	
LocationId	Int	
TaxPricing	String	Inclusive or Exclusive

As Complex As It Gets

The following diagram shows the 3.5g tier pricing for a Shelf priced (Bottom Shelf) item, that has a sale price defined as well as a Veterans group price. Again, left is Cova Hub, right is a truncated JSON response. Lowest price wins when building a cart, so a valid customer in the Veterans pricing group would get an 1/8th for $19.98. On an ecommerce site, the *AtTierPrice* or *AtTierSalePrice* should be shown, depending on the date, or the Veterans price if the ecomm site ties to an appropriate Cova customer profile.

Figure 11: very complex price



Shelf Price	Bottom Shelf
Base Price	
Quantity	1 g
Regular Price	$10.00 *($10/g)*
Sale Price	*No Current or Future Sales*
Veterans	$9.00 *($9/g)*
1 g	
Quantity	1 g
Regular Price	$10.00 *($10/g)*
Sale Price	*No Current or Future Sales*
1/8 oz	
Quantity	3.5 g
Regular Price	$24.29 *($6.94/g)*
Sale Price	$20 *($5.72/g) on Apr 20th, 2024, 12:00 AM to Apr 22nd, 2024, 11:59 PM (UTC-6) Regina*
Veterans	$19.98 *($5.71/g)*

```
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "0a73aaf0-a347-4681-9e37-038062ab72a5",
    "TierId": 2,
    "TierName": "1/8 oz",
    "TierQuantity": 3.5,
    "GroupId": null,
    "GroupName": null,
    "ShelfId": 26,
    "ShelfName": "Bottom Shelf",
    "Price": 6.94,
    "AtTierPrice": 24.29,
    "SalePrices": [
        {
            "SalePrice": 5.72,
            "AtTierSalePrice": 20.0,
            "StartDateUtc": "2024-04-20T06:00:00Z",
            "StopDateUtc": "2024-04-23T05:59:00Z"
        }
    ],
    "UpdatedDateUtc": "2024-04-18T15:55:00.41Z"
},
{
    "LocationId": 94451,
    "FromEntityId": 94447,
    "ProductId": "0a73aaf0-a347-4681-9e37-038062ab72a5",
    "TierId": 2,
    "TierName": "1/8 oz",
    "TierQuantity": 3.5,
    "GroupId": 700,
    "GroupName": "Veterans",
    "ShelfId": 26,
    "ShelfName": "Bottom Shelf",
    "Price": 5.71,
    "AtTierPrice": 19.98,
    "SalePrices": [],
    "UpdatedDateUtc": "2024-04-18T15:55:00.41Z"
}
```