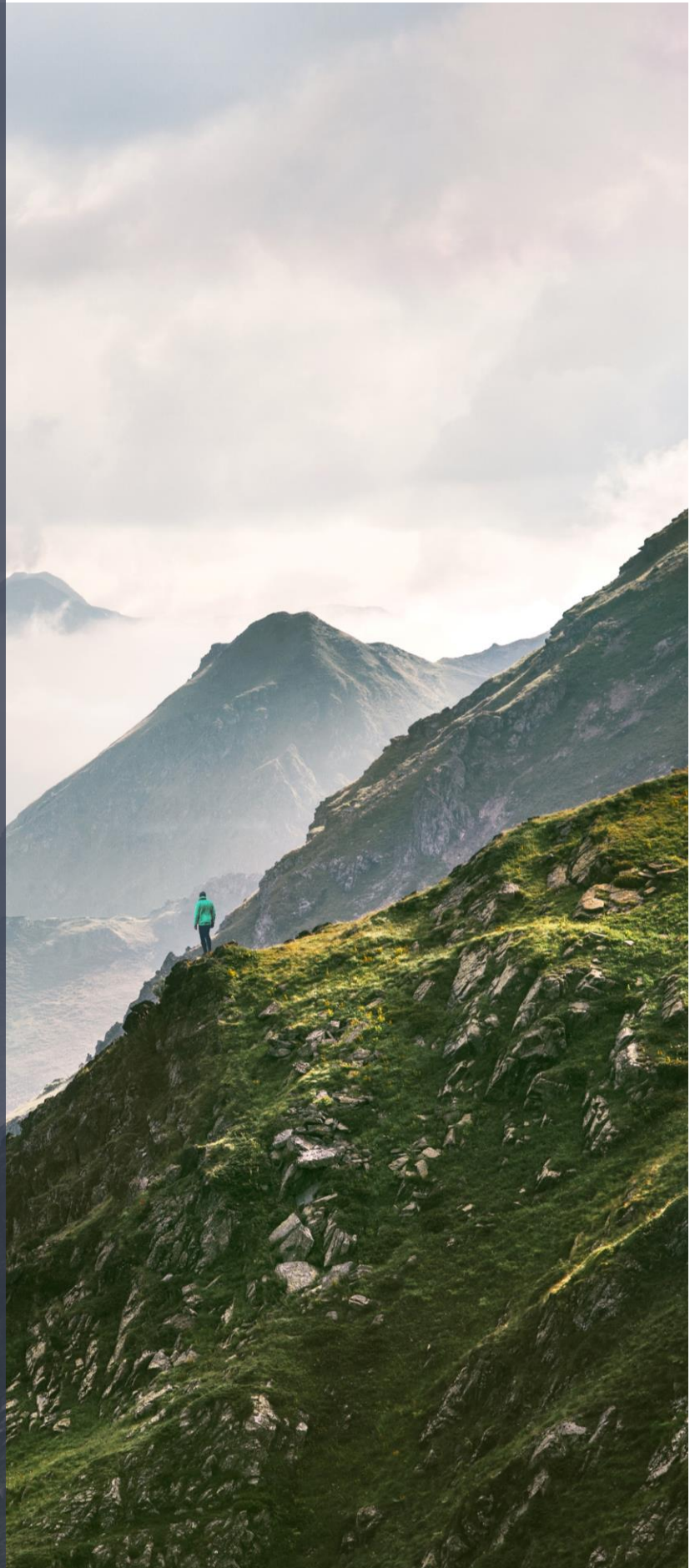# COVA

CANNABIS RETAIL SOFTWARE

Data Platform

Explanations and examples

2024

Cova Data Platform APIs

The Data Platform was created to simplify, combine, and streamline many repetitive API calls for Catalog Items, Pricing, and Inventory. These endpoints allow the flexibility to choose the level of detail on product information, images, availability, pricing, package details, and more, all in a single paged call. This document outlines the currently available endpoints and how they can be used to simplify your integration with Cova.

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, commonly represented as *EntityId*. The Data Platform APIs will use *LocationId*

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.

Paging/Paged: The APIs listed below support and require paging as part of the request body. Paging simply means that you cannot get all possible results in one call and will therefore be required to call the endpoint once for each page of data. These are represented by *Skip* and *Top* integer properties. Skip represents how many results to ignore before returning the Top N results. Five hundred (500) is the maximum result size that can be returned at one time and is therefore the maximum value accepted for *Top*. To retrieve all the paged data, begin with Skip = 0 and for each subsequent call, increment the *Skip* property by the value of *Top* until no more results are returned.

Syncing product detail, pricing, availability, etc.

Cova has curated a comprehensive set of essential data commonly used in constructing ecommerce sites, menu boards, and various cannabis product-driven offerings. This encompasses vital information such as product names and attributes, images, pricing, applicable taxes, and real-time stock levels. Upon establishing a relationship between you, our valued partner, and a Cova retailer, the initial step involves synchronizing all product data, storing these values within your system. Subsequently, to ensure ongoing accuracy, it is critical to stay informed of any changes in product properties, pricing updates, and changes in available inventory for sale. In the event of any oversights or processing failures, individual items can be synchronized separately, offering a flexible and robust solution for data consistency. These three synchronization scenarios are detailed below.

Initial Sync

This endpoint is designed to gather all aggregated data, based on the parameters set in the request body. It is intended for first-time synchronization or occasional, on-demand updates. However, **it is crucial to note that continuous usage of this endpoint is strongly discouraged**. If used as a complete system refresh, it *should not* be exposed to end users, rather as part of an internal tool to run on demand only if absolutely necessary.

`POST` dataplatform/v1/Companies/{CompanyId}/DetailedProductData

Body:

```
{
    "LocationId": {{LocationId}},
    "IncludeProductSkusAndUpcs": false,
    "IncludeProductSpecifications": true,
    "IncludeClassifications": false,
    "IncludeProductAssets": false,
    "IncludeAvailability": true,
    "IncludePackageDetails": false,
    "IncludePricing": true,
    "IncludeTaxes": false,
    "IncludeAllLifecycles": false,
    "IncludeArchived": false,
    "InStockOnly": true,
    "SellingRoomOnly": true,
    "Skip": 0,
    "Top": 500
}
```

- Request body properties are defined below.
- The LocationId field can be null for this request. The results returned will include data for all locations where applicable (i.e. IncludeAvailability, IncludePricing). As product records are shared among the locations of a retailer, using a null value for LocationId may be most efficient for an initial company-wide synchronization.

- Paging fields (Skip, Top) can be used to balance between call volume and response size. Top will control the count of Products returned, but the collections within each product for pricing and availability are not restricted. Because of this it is highly recommended to allow some configuration in these fields. When syncing data for a retailer with a large location count, a large product catalog, and complex pricing, the response payload can be very large if you choose to always request the maximum of 500 items.

Incremental Sync (Deltas/Updates)

To ensure the consistency of your data, utilize the DetailedProductData/UpdatedAsOf endpoint to retrieve items that have changed since a specified point in time, encompassing changes in product properties, prices, stock levels, and more. We advise polling this endpoint at intervals of once per minute, setting the "as of" date to approximately two to three minutes in the past. This time overlap mitigates the risk of overlooking changes attributed to factors such as date calculations, latency, or race conditions. However, be aware that this approach may result in receiving data from previous calls until the change date falls outside the requested range. Note that the Product UpdatedDateUtc does not reflect the date of changes to pricing, inventory levels, packages, etc.

The recommendation above to sync every minute is the maximum recommended frequency for a near real time experience. Consider the needs of your system and whether a more relaxed cadence is acceptable or even desired to minimize processing and network traffic.

```
POST dataplatform/v1/Companies/{CompanyId}/DetailedProductData/UpdatedAsOf/2023-10-11T17:20:09.000
```
Body:
```
{
    "LocationId": {{LocationId}},
    "IncludeProductSkusAndUpcs": false,
    "IncludeProductSpecifications": true,
    "IncludeClassifications": false,
    "IncludeProductAssets": false,
    "IncludeAvailability": true,
    "IncludePackageDetails": false,
    "IncludePricing": true,
    "IncludeTaxes": false,
    "IncludeAllLifecycles": false,
    "IncludeArchived": false,
    "InStockOnly": false, //← Recommended
    "SellingRoomOnly": true,
    "Skip": 0,
    "Top": 500
}
```

- As of date is UTC.
- LocationId can be null in this request to return data for all locations.
- Use InStockOnly:false to ensure Availability data is returned when reaching zero quantity.
- When *anything* related to the product changes, all current data for that item is returned whether it has specifically changed or not (if specified in the request body)

Targeted Sync (item specific)

The data platform also has an endpoint that can target between one and 500 products directly. This is useful for scenarios where an item or items are reported to be out of sync.

POST dataplatform/v1/Companies/{CompanyId}/DetailedProductData/ByProductIdList

Body:

```
{
    "LocationId": {{LocationId}},
    "IncludeProductSkusAndUpcs": false,
    "IncludeProductSpecifications": true,
    "IncludeClassifications": false,
    "IncludeProductAssets": false,
    "IncludeAvailability": true,
    "IncludePackageDetails": false,
    "IncludePricing": true,
    "IncludeTaxes": false,
    "IncludeAllLifecycles": false,
    "IncludeArchived": false,
    "InStockOnly": true,
    "SellingRoomOnly": true,
    "ProductIds": [
        "{{ProductId-1}}",
        "{{ProductId-2}}",
        …
        "{{ProductId-500}}"
    ]
}
```

Note about empty arrays

An empty JSON array property under a product may mean <u>one</u> of two things:

1. Your request body specified to exclude the data. i.e. `"IncludeProductAssets": false` would return an empty array in the Assets property: `"Assets": []`
2. There is no data for that property, possibly due to records being deleted (or going out of stock), which may signify to remove the data from your system, hide products, etc.

Before removing data from your cache, ensure you have requested the data be returned.

Request Body Properties

Name	Type	Description
LocationId	int32	If provided only data for the provided location will be returned. If null is provided information for all locations will be returned.
IncludeProductSkusAndUpcs	boolean	Indicates whether to include product SKUs and UPCs.

IncludeProductSpecifications	`boolean`	Indicates whether to include product specifications.
IncludeClassifications	`Boolean`	Indicates whether to include the classification structure collection
IncludeProductAssets	`boolean`	Indicates whether to include product assets.
IncludeAvailability	`boolean`	Indicates whether to include availability data.
IncludePackageDetails	`boolean`	Indicates whether to include package details for any packages associated with the product.
InStockOnly	`boolean`	Indicates whether to only include products/packages that have stock (quantity greater than zero). Only applies if IncludeAvailability is true.
SellingRoomOnly	`boolean`	Indicates whether to include product availability data for all rooms or only the default selling room. Only applies if IncludeAvailability is true.
IncludePricing	`boolean`	Indicates whether to include product pricing data.
IncludeTaxes	`boolean`	Indicates whether to include applicable tax information
IncludeAllLifecycles	`Boolean`	Indicates whether to include all lifecycles (i.e. Coming Soon, etc), or only Active products.
IncludeArchived	`Boolean`	Indicates whether to include Archived products
Skip	`int32`	Number of records to skip.
Top	`int32`	Number of records to take.

Response Body Properties

Root

Name	Type	Description
Products	`DetailedProductResource`[]	Always returned
Packages	`PackageResource`[]	Empty if "IncludePackageDetails" = false
TaxRates	`TaxRateResource`[]	Empty if "IncludeTaxes" = false
Classifications	`ClassificationResource`[]	Empty if "IncludeClassifications" = false

DetailedProductResource

Name	Type	Description
ProductId	`guid`	Used throughout Cova as the main identifier for a product
MasterProductId	`int32`	Internally generated identifier. A master product may have variants. For example, a 'Blue Shirt' master may have variants for Small, Medium, and Large. All variants of a master will have the same MasterProductId
MasterProductName	`String`	Name of the master product group. i.e. 'Blue Shirt' from the example above
VariationId	`int32`	Identifier of the variant under a master.

		Nullable if the master has no variants. A master with no variants is simply a regular product.
Slug	`string`	Combination of previous identifiers: 'M{MasterProductId}-V{VariationId}' example: 'M105548-V4'
CatalogSku	`string`	Generated searchable identifier on Cova Hub and POS App
IsArchived	`boolean`	An archived product is a deleted product that should be hidden from consumers
MeasurementType	`string`	Establishes how an item is sold; a single unit (each) or by weight (grams). Valid values:'SingleUnit' or 'Mass'
IsBatchTracked	`boolean`	Indicates whether the item requires a PackageId, an identifier controlled by a governing body (METRC, OCS, etc).
IsNonStock	`boolean`	Not a physical item. For example, a delivery fee
LifeCycle	`string`	Only Active products are returned. Products with Coming Soon, Inactive, Discontinued, or Archived are not returned.
ClassificationId	`int32`	Classification identifier.
ClassificationName	`string`	Classification Name
ClassificationPath	`string`	Path from the top level of the retailer classification tree. Example: Flower > Indica
Name	`string`	Product name
ShortDescription	`string`	A brief product description
LongDescription	`string`	A detailed product description
HeroShotAssetId	`guid`	Identifier of the main asset (image) Nullable
HeroShotUri	`string`	URI for the main asset (image)
SupplierSkus	SupplierSkuResource[]	Empty if IncludeProductSkusAndUpcs = false
ManufacturerSkus	ManufacturerSkuResource[]	Empty if IncludeProductSkusAndUpcs = false
Upcs	UpcResource[]	Empty if IncludeProductSkusAndUpcs = false
ProductSpecifications	SpecificationResource[]	Empty if IncludeProductSpecifications = false
Assets	AssetResource[]	Empty if IncludeProductAssets = false
Availability	AvailabilityResource[]	Empty if IncludeAvailability = false
Prices	PriceResource[]	Empty if IncludePricing = false
CreatedDateUtc	`date-time`	UTC datetime of when the product record was added
UpdatedDateUtc	`date-time`	UTC datetime of the last time the product record was modified
ApplicableTaxRates	`guid[]`	Tax rate identifiers that may be applicable for this item. See TaxRateResource below.

		Empty if IncludeTaxes = false
ClassificationTreeId	int32	Identifier of the classification tree
Manufacturer	string	Name of the Manufacturer
MSRP	Decimal	Manufacturer suggested retail price
MSRPCurrencyCode	String	Currency code for the MSRP

SupplierSkuResource

Name	Type	Description
SKU	string	Supplier SKU.
SupplierId	int32	ID of the supplier associated with the SKU. Nullable
Supplier	string	Name of the supplier associated with the SKU.
Description	string	Description associated with the supplier SKU.

ManufacturerSkuResource

Name	Type	Description
SKU	string	Manufacturer SKU.
Description	string	Description associated with the manufacturer SKU.

UpcResource

Name	Type	Description
UPC	string	
Description	string	Description associated with the UPC.

SpecificationResource

Name	Type	Description
FieldGroupId	guid	The ID of the group the specification belongs to.
FieldGroup	string	The group the specification belongs to. Example: Details
FieldId	int32	The ID of the specification.
FieldName	string	The name of the specification. Example: Retail Marijuana Product Type
StringId	string	The string id of the specification.
DisplayName	string	The display name of the specification. Example: Retail Marijuana Product Type. Use this value to find particular specification fields
Value	string	The value that is set for the field. Comma delimited for 'MultiSelect' type fields
Type	string	The type of specification in relation to Cova Hub controls. Valid values: 'TextSingleLine','SingleSelect','MultiSelect','Float','TextMultipleLine','YesNo'
Unit	string	The unit the value is measured in if applicable.

		Example: g, oz

AssetResource

Name	Type	Description
Id	guid	Unique identifier of the asset.
Name	string	Asset name.
Uri	string	Uri to the asset.
Type	string	Asset type.
IsHidden	boolean	Indicates if the asset is hidden.

AvailabilityResource

Name	Type	Description
LocationId	int32	Location (store) identifier where this item exists
RoomId	int32	Room identifier within a location. For example "Receiving Room" or "Sales Floor"
PackageId	string	Identifier typically controlled by a governing body (METRC, OCS, etc)
BatchId	string	Additional package tracking information
UnitId	int32	Internal identifier for the unit of measure. i.e. 0 is 'Single Unit'. 103 is gram, 110 is milligram
InStockQuantity	double	Quantity of this item in stock, at this location, in this room.
TotalCostInStock	double	The retailer's total cost of these items
UnitCost	double	The retailer's per-unit cost for these items
ReceivedDate	date-time	UTC datetime of when these items were received into inventory
UpdatedDateUtc	date-time	UTC datetime of when this record was last updated (i.e. quantity changed, etc).

PriceResource

Name	Type	Description
LocationId	int32	Location (store) identifier where this price is valid
ProductId	guid	Used throughout Cova as the main identifier for a product
TierId	int32	Tier identifier, if this product is priced in tiers. Example of tiers: 2 or fewer for $5 each; 3 to 5 for $4 each; 6 or more for $3 each Nullable if not a tiered price
TierName	string	Name of the tier, if applicable
TierQuantity	double	Minimum quantity to qualify for the tier price Nullable if not a tiered price
GroupId	int32	Identifier of a pricing group, if applicable
GroupName	string	Name of the pricing group. For example: Veterans, Employees Nullable if not a group price

ShelfId	int32	Identifier of a pricing shelf, if applicable. A Shelf is a way to apply common pricing to multiple products. For example, all items on the "Top Shelf" are $10.
ShelfName	string	Name of the pricing shelf, if applicable Nullable if not a shelf price
Price	double	Regular price for this item
AtTierPrice	double	Price for this item at the lower bound (TierQuantity). Always use this price when calculating totals for tiered items. Nullable if not a tiered price
SalePrices	SalePriceResource[]	
UpdatedDateUtc	date-time	The UTC datetime this pricing record was last modified

SalePriceResource

Name	Type	Description
SalePrice	double	The sale price for this item
AtTierSalePrice	double	Sale Price for this item at the lower bound (TierQuantity). Always use this price when calculating totals for tiered items. Nullable if not a tiered price
StartDateUtc	date-time	UTC datetime for when the sale begins Nullable
StopDateUtc	date-time	UTC datetime for when the sale ends Nullable

PackageResource

Name	Type	Description
ProductsAssociatedWith	guid[]	Collection of product identifiers where this package is referenced.
PackageId	string	The package identifier.
BatchId	string	The batch identifier.
ExpiryDate	date-time	Expiry date of the package.
Strain	string	The package strain.
SupplierLicense	string	The suppliers license number.
TestDate	date-time	The date the package was tested.
HarvestDate	date-time	The date the package was harvested.
TestingFacilityName	string	Name of the facility where the package was tested.
TestBatchNumber	string	Batch number associated with test.
SupplierName	string	Name of the supplier of the package.
SupplierId	int32	Entity Id of the supplier of the package.
TerpeneProfile	string	Terpene profile.
ReceivingId	guid	Unique identifier of the PO the package was received on.
ReceivingDate	date-time	Date that the package was received in store

		local time.
Cannabinoids	CannabinoidResource[]	Collection of cannabinoids associated with the product.
PackagedDate	date-time	The date the item was packaged.
ProductionDate	date-time	The date the item was produced.
CreatedDateUtc	date-time	The date the package was created.
UpdatedDateUtc	date-time	The date the package was last updated.

CannabinoidResource

Name	Type	Description
CannabinoidType	string	The cannabinoid. Example: THC, CBD
MeasurementType	string	How the cannabinoid is measured. Example: percent, mg
MinValue	double	The minimum value entered for the cannabinoid.
MaxValue	double	The maximum value entered for the cannabinoid.

TaxRateResource

Name	Type	Description
Id	guid	The tax identifier.
Name	string	Display name of the tax
Type	string	Percentage or FlatRate
Rate	double	The tax rate. For Percentage type, this is a decimal. (8% tax = .08). FlatRate is a dollar amount (50 cent deposit = 0.50)
Layer	int32	Identifies if the tax is compounding: 0 = non-compounding, 1 = Compounds on non-compound taxes (Level 2), 2 = Compounds after all other taxes (Level 3)
CalculatedOn	string	SoldAtPrice or BasePriceListPrice. Indicates whether the tax is calculated on the base price or the sold at price.
Locations	int32[]	A collection of Cova location identifiers where this tax is applicable.

ClassificationResource

Name	Type	Description
Id	int	Classification identifier
TreeId	int	Identifier of the classification tree
Name	string	Classification name
ParentCategories	parentCategoryResource[]	Collection of the parent categories for this classification. Ordered the same as visual order in Hub

ParentCategoryResource

Name	Type	Description

Id	int	Category identifier
Name	string	Name of the category