



NCS

JSON API Guide

Contents

About this Document	3
Changes	3
Authentication	4
RESTful	4
Process by User Type	5
Server Responses	5
Documentation Table of Contents	8
Appendix A - Definitions	36
Appendix B – Troubleshoot	43

Release Versions

Version Number	Release Date	Release Contents
1.0	05/2019	

Before You Start...

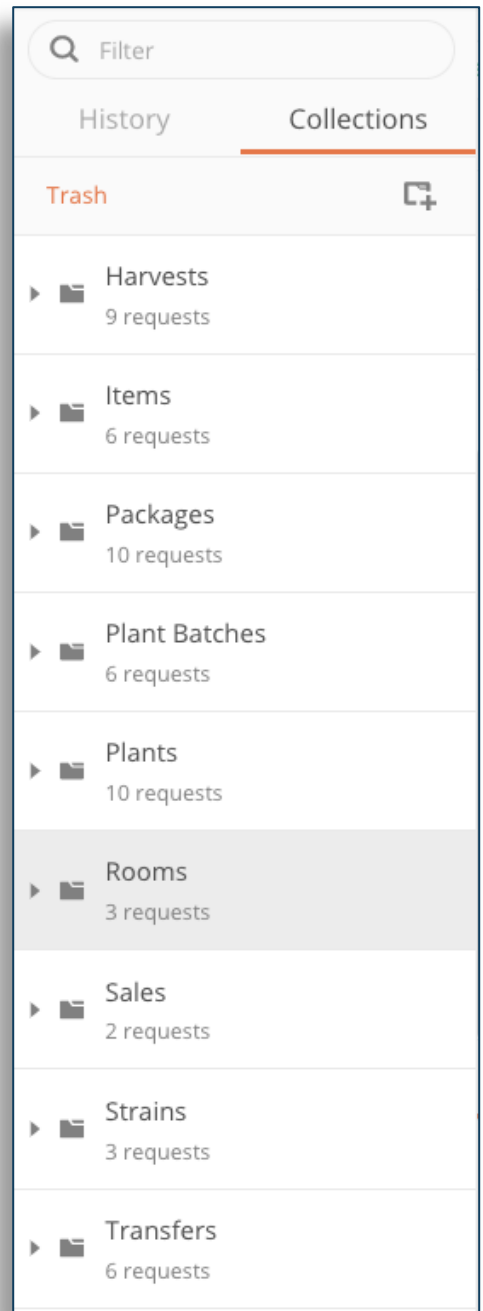
Welcome to the NCS Analytics POS API JSON manual. This manual serves as a comprehensive guide to help you understand the functions and key data needed to successfully integrate to the NCS system.

The purpose of the NCS POS API is to provide data to the NCS Platform for use by an NCS client. It is not meant to function as a seed to sale system or replace any other tracking systems, it simply accepts the information you already have.

Updates/Changes

All updates or changes made to the API will be documented here. NCS will communicate upcoming changes prior to release in an API Bulletin. Questions regarding changes can be emailed to api@ncsanalytics.com.

Since 05/01/2019



API Process by User Type

All license types - cultivators, manufacturers, and retailers, require basic system setup information prior to loading daily activity. This includes Rooms, Strains, Item Categories and Items. These Collections are a baseline foundation for successful calls as you navigate throughout the system and will need to be referenced when making Harvests or Packages, for example.

For Cultivators

After creating Rooms, Strains, Item Categories and Items you will make your Plant Batches and Plants. Please note that you must create a Plant Batch first before moving on to the Plants Collection. This is a crucial step in being able to successfully complete a Harvest later on. In the Plants Collection, you will be able to Move Plants, Change Growth Phases of Plants (assign a tag or label), Manicure and Destroy Plants. Note that a tag or label is required for any plant included in a Harvest.

For Manufacturers

After creating Rooms, Strains, Item Categories and Items you can create packages, sell and initiate transfers. You may choose to include the intermediary products during the manufacturing process in Items or in Packages.

For Retailers

After Rooms, Strains, Item Categories and Items you can establish packages, record sales and initiate transfers. Each incoming transfer of inventory will need to be recorded as well as a new package created to record the contents of the new inventory.

Server Responses

In this section we will review errors and give some suggested fixes. Further details on each error can be found later in this guide as well as in the online API documentation.

Server Response	Reason
200 OK	The server was able to process the request successfully
400 Bad Request	An error has occurred while executing your request. Check the payload of the call for errors, stray commas, typos, etc.
401 Unauthorized	The token you provided is not authorized or invalid. Or you have not provided your IP address for whitelisting.
404 Not Found	No vendor or location found for submitted request.
406 Not Found	No related data found. Have you already posted strains, plant batches, labels, or other data that is required by the call.
500 Internal Server Error	An error has occurred while executing your request. Payload may be too large. Break up request and re-submit.

Documentation Table of Contents

Rooms

POST/pos/rooms/v1/create.....	8
POST/pos/rooms/v1/update	8
GET/pos/rooms/v1/all	8

Strains

POST/pos/strains/v1/create	9
POST/pos/strains/v1/update	9
GET/pos/strains/v1/all	9

Item Categories and Items

POST/pos/items/v1/create/categories	10
POST/pos/items/v1/create	11
POST/pos/items/v1/update	12
POST/pos/items/v1/update/categories	12
GET/pos/items/v1/all	13
GET/pos/items/v1/categories	14

Plant Batches

POST/pos/plantbatches/v1/createplantings	14
POST/pos/plantbatches/v1/createpackages.....	15
POST/pos/plantbatches/v1/destroy.....	15
POST/pos/plantbatches/v1/changegrowthphase.....	15
GET/pos/plantbatches/v1/active.....	16
GET/pos/plantbatches/v1/inactive.....	16
GET/pos/plantbatches/v1/plantid/{id}	17

Plants

POST/pos/plants/v1/create/plantings	17
POST/pos/plants/v1/changegrowthphases	18
POST/pos/plants/v1/moveplants	18
POST/pos/plants/v1/manicureplants	18
POST/pos/plants/v1/harvestplants	19
POST/pos/plants/v1/destroyplants	19
GET/pos/plants/v1/all	19
GET/pos/plants/v1/vegetative	20

GET/pos/plants/v1/flowering	20
GET/pos/plants/v1/{id}	21

Harvests

POST/pos/harvests/v1/createpackages	22
POST/pos/harvests/v1/create	22
POST/pos/harvests/v1/update	23
POST/pos/harvests/v1/removewaste	23
POST/pos/harvests/v1/finish	23
POST/pos/harvests/v1/unfinish	23
GET/pos/harvests/v1/active	24
GET/pos/harvests/v1/inactive	24
GET/pos/harvests/v1/onhold	25

Packages

POST/pos/packages/v1/create	25
POST/pos/packages/v1/create/plantings	26
POST/pos/packages/v1/change/item	26
POST/pos/packages/v1/adjust	26
POST/pos/packages/v1/finish	26
POST/pos/packages/v1/unfinish	27
POST/pos/packages/v1/remediate	27
GET/pos/packages/v1/active	27
GET/pos/packages/v1/inactive	28
GET/pos/packages/v1/onhold	29

Sales

POST/pos/sales/v1/receipts	30
GET/pos/sales/v1/receipts	30

Transfers

POST/pos/transfers/v1/incoming.....	31
POST/pos/transfers/v1/outgoing.....	32
POST/pos/transfers/v1/rejected.....	32
GET/pos/transfers/v1/incoming.....	33
GET/pos/transfers/v1/outgoing.....	34
GET/pos/transfers/v1/rejected.....	35

Rooms

In this section we will review how to add and modify rooms. The Rooms Collection is the foundation of your information and a critical place to start in this process. Examples of Rooms are “Plants Room”, “Drying Room”, “Harvest Room”, “Drying Room”, etc.

POST/pos/rooms/v1/create

The first step you will take when starting the API process is creating a room. Rooms can be physical rooms within a location, or they can be artificial based on how a licensee chooses to organize their inventory.

```
{
  "Name": "Harvest Room"      ← required
}
```

POST/pos/rooms/v1/update

If a room needs to be updated to a different name, first locate the ‘id’ associated to the room and update the name as needed.

```
{
  "Id": 5,                    ← required
  "Name": "Harvesting Room"  ← required
}
```

GET/pos/rooms/v1/all

Return all rooms that have been created for a location.

```
{
  "id": 19,
  "name": "Concentrates Room"
},
{
  "id": 20,
  "name": "Edibles Room"
},
{
  "id": 21,
  "name": "Infused Room"
}
```


Strains

In this section we will review how to add and modify strains. The Strains Collection set up will be very similar to the rooms process you just completed.

[POST/pos/strains/v1/create](#)

You must populate the system with the strains for each location.

```
{
  "Name": "Strain1",           ← required
  "TestingStatus": "None",
  "ThcLevel": 0.5000,
  "CbdLevel": 0.5000,
  "IndicaPercentage": 50.0,
  "SativaPercentage": 50.0
}
```

[POST/pos/strains/v1/update](#)

This is where you will update a strain you have previously created.

```
{
  "Id": 1,                     ← required
  "Name": "Strain1",
  "TestingStatus": null,
  "ThcLevel": 0.1865,
  "CbdLevel": 0.1075,
  "IndicaPercentage": 25.0,
  "SativaPercentage": 75.0
}
```

[GET/pos/strains/v1/all](#)

Returns all strains created for a Location.

```
{
  "id": 36,
  "name": "Concentrates Strain",
  "testingStatus": "None",
  "thcLevel": 0.5,
  "cbdLevel": 0.5,
  "indicaPercentage": 50,
  "sativaPercentage": 50,
  "genetics": null
}
```

Item Categories and Items

In this section we will review how to add and modify Item Categories and Items. It is important to create item categories first and then create individual items.

<POST/pos/items/v1/create/categories>

Before creating any item, you must first create the categories for your inventory. Common categories are:

Capsule	Infused Butter/Oil	Seeds
Edible (volume-each)	Infused Butter/Oil (each)	Seeds (each)
Edible (volume)	Kief	Tincture
Edible (weight-each)	Leaf	Tincture (each)
Edible (weight)	Other Concentrate (volume-each)	Topical (volume-each)
Extract (volume-each)	Other Concentrate (volume)	Topical (volume)
Extract (volume)	Other Concentrate (weight-each)	Topical (weight-each)
Extract (weight-each)	Other Concentrate (weight)	Topical (weight)
Extract (weight)	Pre-Roll Flower	Vape Cartridge
Flower	Pre-Roll Infused	Waste
Fresh Cannabis Plant	Pre-Roll Leaf	
Immature Plant		

An item category is a way to indicate the type of product your inventory is associated with. The “name” is what you will continue to use when you create subsequent items. Quantity type identifies the kind of quantity for example “CountBased”, “WeightBased”, or “VolumeBased”.

```
{
  "Name": "Edible (weight-each)",           ← required
  "ProductCategoryType": "Edible (weight-each)", ← required
  "QuantityType": "CountBased",           ← required
  "DefaultLabTestingState": "NotSubmitted",
  "RequiresApproval": false,
  "RequiresAdministrationMethod": false,
  "RequiresStrain": true,
  "RequiresUnitCbdPercent": false,
  "RequiresUnitCbdContent": false,
  "RequiresUnitThcPercent": false,
  "RequiresUnitThcContent": false,
  "RequiresUnitVolume": false,
  "RequiresUnitWeight": false,
  "RequiresServingSize": false,
  "RequiresSupplyDurationDays": false,
  "UnitQuantityMultiplier": null,
  "UnitQuantityUnitOfMeasureName": null,
  "RequiresIngredients": false,
  "RequiresProductPhoto": false,
  "CanContainSeeds": false,
  "CanBeRemediated": false
},
```

POST/pos/items/v1/create

Creating items is where you build out your inventory, regardless of the type of license you hold. Before creating Items, you must create Item Categories. If you attempt to post an item with an item category that does not exist, you will receive an error. "Strain Name" must be from an existing record for that license.

```
{
  "ProductCategoryName": "Other Concentrate (weight-each)", ← required
  "Name": "Wax", ← required
  "UnitOfMeasureName": "Grams", ← required
  "AdministrationMethod": null,
  "StrainName": "Strain1", ← required, must be existing Strain
  "IntermediaryProductUsed": "true", ← required: if true, Intermediaries required
                                     below*

  "UnitCbdPercent": null,
  "UnitCbdContent": null,
  "UnitCbdContentUnitOfMeasureName": null,
  "UnitThcPercent": null,
  "UnitThcContent": null,
  "UnitThcContentUnitOfMeasureName": null,
  "UnitVolume": null,
  "UnitVolumeUnitOfMeasureName": null,
  "UnitWeight": null,
  "UnitWeightUnitOfMeasureName": null,
  "ServingSize": null,
  "SupplyDurationDays": null,
  "Ingredients": null,
  "Intermediaries": [ ← required*
    {
      "Id": 3, ← required if existing
      "Name": "KushOG", ← required
      "Type": "Flower" ← required
    }
  ]
}
```

Items with intermediary items have unique rules. Intermediaries can be used an unlimited number of times when creating items. The first time you use an intermediary, you will not include an id:

```
"Intermediaries": [ ← required*
  {
    "Name": "KushOG", ← required
    "Type": "Flower" ← required
  }
]
```

After posting this intermediary, you will be able to request the associated ID and use it again if need be. The ID is systematically generated and retrieved by using the GET command.

POST/pos/items/v1/update

Used to update an item after it is created. If intermediaries already exist, you must include the id associated with them, which you can retrieve using the GET/items/all.

```
/
  "Id": 172,                               ← required existing ID
  "ProductCategoryName": "Edible (weight-each)", ← required
  "Name": "Chocolate Cookie",              ← required
  "UnitOfMeasureName": "Each",             ← required
  "AdministrationMethod": null,
  "StrainName": "Strain1",                 ← required, must be existing Strain
  "IntermediaryProductUsed": true,
  "IntermediaryProductType": null,
  "UnitCbdPercent": null,
  "UnitCbdContent": null,
  "UnitCbdContentUnitOfMeasureName": null,
  "UnitThcPercent": null,
  "UnitThcContent": null,
  "UnitThcContentUnitOfMeasureName": null,
  "UnitVolume": null,
  "UnitVolumeUnitOfMeasureName": null,
  "UnitWeight": null,
  "UnitWeightUnitOfMeasureName": null,
  "ServingSize": null,
  "SupplyDurationDays": null,
  "Ingredients": null,
  "Intermediaries": [
    {
      "Id": 4,                               ← required if existing
      "Name": "Butter",                       ← required
      "Type": "Edible (weight-each)"         ← required
    },
    {
      "Name": "Oil",                           ← required
      "Type": "Edible (weight-each)"         ← required
    }
  ]
}
```

POST/pos/items/v1/update/categories

Used to update an item category that has already been created.

```
{
  "Id": 1,                                   ← required
  "Name": "Edible (weight-each)",           ← required
  "ProductCategoryType": "Edible (weight-each)", ← required
  "QuantityType": "CountBased",             ← required
  "DefaultLabTestingState": "NotSubmitted",
  "RequiresApproval": false,
  "RequiresAdministrationMethod": false,
  "RequiresStrain": true,
  "RequiresUnitCbdPercent": false,
  "RequiresUnitCbdContent": false,
  "RequiresUnitThcPercent": false,
}
```

```
"RequiresUnitThcContent": false,
"RequiresUnitVolume": false,
"RequiresUnitWeight": false,
"RequiresServingSize": false,
"RequiresSupplyDurationDays": false,
"UnitQuantityMultiplier": null,
"UnitQuantityUnitOfMeasureName": null,
"RequiresIngredients": false,
"RequiresProductPhoto": false,
"CanContainSeeds": false,
"CanBeRemediated": false
},
```

GET/pos/items/v1/all

Returns all items that exist for a license as well as get the IDs for each item and their intermediaries.

```
{
  "id": 338,
  "name": "Wax",
  "productCategoryName": "Extract (weight)",
  "productCategoryType": null,
  "quantityType": null,
  "unitOfMeasureName": "Grams",
  "administrationMethod": null,
  "strainName": "Strain1",
  "intermediaryProductUsed": true,
  "unitCbdPercent": null,
  "unitCbdContent": null,
  "unitCbdContentUnitOfMeasureName": null,
  "unitThcPercent": null,
  "unitThcContent": null,
  "unitThcContentUnitOfMeasureName": null,
  "unitVolume": null,
  "unitVolumeUnitOfMeasureName": null,
  "unitWeight": null,
  "unitWeightUnitOfMeasureName": null,
  "servingSize": null,
  "supplyDurationDays": null,
  "ingredients": null,
  "intermediaries": [
    {
      "id": 30,
      "name": "Hydrocarbon Wax",
      "type": "Concentrate"
    }
  ]
}
```

GET/pos/items/v1/categories

Returns all item categories that exist for a license.

```
{
  "id": 22,
  "name": "Other Concentrate (weight)",
  "productCategoryType": "Other Concentrate (weight)",
  "quantityType": "WeightBased",
  "defaultLabTestingState": "NotSubmitted",
  "requiresApproval": false,
  "requiresAdministrationMethod": false,
  "requiresStrain": true,
  "requiresUnitCbdPercent": false,
  "requiresUnitCbdContent": false,
  "requiresUnitThcPercent": false,
  "requiresUnitThcContent": false,
  "requiresUnitVolume": false,
  "requiresUnitWeight": false,
  "requiresServingSize": false,
  "requiresSupplyDurationDays": false,
  "unitQuantityMultiplier": null,
  "unitQuantityUnitOfMeasureName": null,
  "requiresIngredients": false,
  "requiresProductPhoto": false,
  "canContainSeeds": false,
  "canBeRemediated": false
}
```

Plant Batches

In this section we will review how to add and modify Plant Batches. All plants must start in a batch. This step will become critical in order to harvest your plants later.

POST/pos/plantbatches/v1/createplantings

As a cultivator (grower) you will create a plant batch to start your grow cycle. You can create a batch from things like seeds or clones. Batch name is a unique field and creating a batch is required. When you complete this call, the system automatically creates the individual plants for you.

```
{
  "Name": "Clone Batch April 2019",      ← required
  "Type": "Clones",
  "Count": 100,                          ← required
  "StrainName": "Strain1",              ← required, must be existing
                                         Strain
  "RoomName": "Clones Room",           ← must be existing room
  "PlantedDate": "2019-04-01"
}
```

POST/pos/plantbatches/v1/createpackages

One of three ways to create a package. Creating a package from plant batches directly would be used when you are packaging immature plants at any point during the growing cycle. You must assign a package label and indicate the number of plants you are packaging from the batch. You will also include the item that is being packaged- this must be an existing item for that license.

```
{
  "PlantBatchId": 271,           ← required
  "PlantBatchName": "Clone Batch April 2019", ← required existing Plant Batch
  "Quantity": 5,                ← required
  "RoomName": "Clones Room",    ← must be existing Room
  "ProductName": "Immature Plants", ← required existing Item
  "Label": "UNIQUEPACKAGELABEL", ← required unique Package Label
  "PackagedDate": "2019-05-01"
}
```

POST/pos/plantbatches/v1/destroy

If plants within a batch need to be destroyed for any reason, you will record the name, number count, and date they were destroyed here. Plants can be destroyed at any time during the growth phase.

```
{
  "Name": "Clone Batch April 2019", ← required existing Plant Batch
  "DestroyedCount": 20,              ← required
  "DestroyedNote": "Mold",
  "DestroyedDate": "2019-04-23"     ← required
}
```

POST/pos/plantbatches/v1/changegrowthphase

If you need to change the growth phase for an entire plant batch, you will be able to do so here. It does not require the entire plant count to be altered but you cannot dictate which plant IDs will be altered. If you choose to add "NewTag" here, you will be adding the same tag to all plants being altered. If you would like to add an individual plant tag use the calls in the plants collection. Common growth phases would be "Vegetative" and "Flowering".

```
{
  "Name": "Clone Batch April 2019", ← required existing Plant Batch
  "Count": 25,                       ← required
  "NewTag": null,
  "GrowthPhase": "Vegetative",      ← required
  "RoomName": "Plants Room",        ← must be existing Room
  "GrowthDate": "2019-04-15"
}
```

GET/pos/plantbatches/v1/active

Returns plant batches that are active for a license.

```
{
  "id": 271,
  "name": "Clone Batch April 2019",
  "type": "Clones",
  "roomId": 2,
  "roomName": "Clones Room",
  "strainId": 2,
  "strainName": "Strain1",
  "count": 100,
  "liveCount": 70,
  "packagedCount": 5,
  "harvestedCount": 0,
  "destroyedCount": 20,
  "sourcePackageId": null,
  "sourcePackageLabel": null,
  "sourcePlantId": null,
  "sourcePlantLabel": null,
  "plantedDate": "2019-04-23T00:00:00",
  "lastModified": null,
  "createDate": "2019-04-23T20:51:48.5906072",
  "updateDate": "2019-04-23T20:51:48.5906072"
}
```

← Count is the initial count of Plants in the Batch
← Live count will adjust as you Package, Harvest or Destroy Plants in a Batch

GET/pos/plantbatches/v1/inactive

Returns any inactive plant batches for a licensee.

```
{
  "id": 68,
  "name": "BATCH1",
  "type": null,
  "roomId": 0,
  "roomName": "Flower Room",
  "strainId": null,
  "strainName": null,
  "count": 0,
  "liveCount": 0,
  "packagedCount": 0,
  "harvestedCount": 0,
  "destroyedCount": 0,
  "sourcePackageId": null,
  "sourcePackageLabel": null,
  "sourcePlantId": null,
  "sourcePlantLabel": null,
  "plantedDate": null,
  "lastModified": null,
  "createDate": "2019-04-05T20:42:55.7077087",
  "updateDate": "2019-04-05T20:42:55.7077087"
}
```


GET/pos/plantbatches/v1/plantid/{id}

A way to get all plant IDs associated to a plant batch.

```
{
  "id": 304,
  "plantbatchname": "Concentrates Batch April 2019",
  "plantCount": 5,
  "plantid": [
    {
      "id": 328333,
      "label": null
    },
    {
      "id": 328334,
      "label": null
    },
    {
      "id": 328335,
      "label": null
    },
    {
      "id": 328336,
      "label": null
    },
    {
      "id": 328337,
      "label": null
    }
  ]
}
```

Plants

In this section we will review how to add and modify plants. Plants will automatically populate once you created your batch. To view your already populated plants, go to 'GET/all' under plants on the left side navigation panel.

POST/pos/plants/v1/create/plantings

A way to add additional plants to an existing plant batch. This action can only add one individual plant at a time, so "plantcount" must be '1'.

```
{
  "Label": "UNIQUEPACKAGELABEL_2",           ← required unique Plant Label
  "PlantBatchName": " Clone Batch April 2019", ← required existing Plant Batch
  "PlantBatchTypeName": "Clone",
  "PlantCount": 1,                            ← required and must be '1'
  "RoomName": "Clones Room",                 ← must be existing Room
  "StrainName": "Strain1",                   ← required, must be existing Strain
  "PlantedDate": "2019-04-01T13:11:03Z"
},
```

POST/pos/plants/v1/changegrowthphases

Where you will change growth phases for individual plants. You also will add individual plant tags in this call. Common growth phases would be “Vegetative” and “Flowering”.

```
{
  "Id": 320956,           ← required individual Plant ID
  "Label": null,         ← if individual plant has an existing Label
  "NewTag": "PLANTTAG_1", ← NEW unique Plant Tag
  "GrowthPhase": "Flowering",
  "NewRoom": "Flower Room", ← must be existing Room
  "GrowthDate": "2019-04-21"
}
```

POST/pos/plants/v1/moveplants

Where you will move plants from one room to another. Either plant ID or plant label are required, you do not need both.

```
{
  "Id": null,           ← required if no Label
  "Label": "PLANTLABEL_1", ← required if no ID
  "RoomName": "Flower Room_2" ← must be existing Room
},
```

POST/pos/plants/v1/manicureplants

At any point during the grow cycle, sellable material can be manicured from a plant with a unique plant label. Any manicured weight must be recorded. If you are manicuring several plants, you will first create a manicure harvest in order to collect the weights from all of the plants. From there you will package the manicured weight in preparation for transfer and sale.

```
{
  "Label": "PLANTLABEL_1", ← required existing Plant Label
  "ManicuredWeight": 89.7, ← required
  "ManicuredUnitOfWeightName": "Grams", ← required
  "RoomName": "Flower Room_2", ← must be existing Room
  "HarvestName": "Manicure Harvest_1", ← required if moving into a manicure Harvest
  "ManicuredDate": "2019-04-23"
},
```

POST/pos/plants/v1/harvestplants

This is how you will harvest plants into an existing harvest. A harvest must first be created before you can harvest individual plants. A plant must have a unique plant label before you can harvest it.

```
{
  "Label": "PLANTLABEL_1",           ← required existing Plant Label
  "HarvestedWetWeight": 200.99,      ← required
  "HarvestedUnitOfWeightName": "Grams", ← required
  "RoomName": "Flower Room_2",      ← must be existing Room
  "HarvestName": "Harvest_April2019", ← required, must be existing Harvest
  "HarvestedDate": "2019-04-23"
}
```

POST/pos/plants/v1/destroyplants

This call is used when a plant is no longer viable and is being removed from your inventory. The plant must already exist in the system and have a unique Id in order to be destroyed. You will need to note the reason the plant is no longer viable; some examples are mold or bugs.

```
{
  "Id": 24,                          ← required unique Plant ID
  "Label": null,
  "DestroyedNote": "Mold",           ← required
  "DestroyedDate": "2019-04-15"     ← required
}
```

GET/pos/plants/v1/all

Returns all plants for an associated location. Note: Maximum response size is around 100,000 lines. If you receive a 500 error, please locate Plant ID via GET/pos/plantbatches/v1/plantid/{id}.

```
{
  "id": 320954,
  "label": null,
  "plantBatchName": "PlantBatch1"
},
{
  "id": 320955,
  "label": null,
  "plantBatchName": "PlantBatch1"
}
```

GET/pos/plants/v1/vegetative

Returns all plants in the vegetative growth phase for an associated location. Note: Maximum response size is around 100,000 lines. If you receive a 500 error, please locate Plant ID via GET/pos/plantbatches/v1/plantid/{id}.

```
{
  "id": 320954,
  "label": null,
  "state": null,
  "growthPhase": "Vegetative",
  "plantBatchId": 0,
  "plantBatchName": "PlantBatch1",
  "plantBatchTypeName": "Clone",
  "strainId": 0,
  "strainName": "Strain1",
  "roomId": 0,
  "roomName": "Growing Room",
  "harvestId": 0,
  "harvestedUnitOfWeightName": null,
  "harvestedUnitOfWeightAbbreviation": null,
  "harvestedWetWeight": 0,
  "manicuredUnitOfWeightName": null,
  "manicuredWeight": 0,
  "harvestCount": 0,
  "isOnHold": false,
  "plantedDate": "2019-01-01T00:00:00",
  "vegetativeDate": "2019-01-08T00:00:00",
  "floweringDate": null,
  "harvestedDate": null,
  "destroyedDate": null,
  "destroyedNote": null,
  "destroyedByUserName": null,
  "lastModified": "2019-03-02T16:14:43.4458418"
}
```

GET/pos/plants/v1/flowering

Returns all plants in the flowering growth phase for an associated location. Note: Maximum response size is around 100,000 lines. If you receive a 500 error, please locate Plant ID via GET/pos/plantbatches/v1/plantid/{id}.

```
{
  "id": 320954,
  "label": "UNIQUELABEL_2",
  "state": null,
  "growthPhase": "Flowering",
  "plantBatchId": 0,
```

```

"plantBatchName": "PlantBatch1",
"plantBatchTypeName": "Clone",
"strainId": 0,
"strainName": "Strain1",
"roomId": 0,
"roomName": "Flower Room",
"harvestId": 0,
"harvestedUnitOfWeightName": null,
"harvestedUnitOfWeightAbbreviation": null,
"harvestedWetWeight": 0,
"manicuredUnitOfWeightName": null,
"manicuredWeight": 0,
"harvestCount": 0,
"isOnHold": false,
"plantedDate": "2019-01-01T00:00:00",
"vegetativeDate": "2019-01-08T00:00:00",
"floweringDate": "2019-01-15T00:00:00",
"harvestedDate": null,
"destroyedDate": null,
"destroyedNote": null,
"destroyedByUserName": null,
"lastModified": "2019-03-02T16:14:43.4458418"
}

```

[GET/pos/plants/v1/{id}](#)

Returns a specific plant by plant ID.

```

{
  "id": 320954,
  "label": "UNIQUELABEL_2",
  "state": null,
  "growthPhase": "Flowering",
  "plantBatchId": 0,
  "plantBatchName": "PlantBatch1",
  "plantBatchTypeName": "Clone",
  "strainId": 0,
  "strainName": "Strain1",
  "roomId": 0,
  "roomName": "Flower Room",
  "harvestId": 0,
  "harvestedUnitOfWeightName": null,
  "harvestedUnitOfWeightAbbreviation": null,
  "harvestedWetWeight": 0,
  "manicuredUnitOfWeightName": null,
  "manicuredWeight": 0,

```

```

    "harvestCount": 0,
    "isOnHold": false,
    "plantedDate": "2019-01-01T00:00:00",
    "vegetativeDate": "2019-01-08T00:00:00",
    "floweringDate": "2019-01-15T00:00:00",
    "harvestedDate": null,
    "destroyedDate": null,
    "destroyedNote": null,
    "destroyedByUserName": null,
    "lastModified": "2019-03-02T16:14:43.4458418"
  }

```

Harvests

In this section we will review how to create a Harvest, update a Harvest and Remove Waste from your Harvest.

[POST/pos/harvests/v1/createpackages](#)

This is the second of three ways to create a package. As a cultivator, you will use this call to prepare packages for your harvested plants. This step will come after you have created a harvest and have actually harvested your plants. You will reference your Harvest by using the Harvest Id.

```

{
  "HarvestId": 14,                ← required existing Harvest
  "Label": "packagelabel_3",     ← required
  "RoomName": null,              ← must be existing Room
  "ProductName": "Edible (weight-each)", ← required existing Item
  "Weight": 50,                  ← required
  "UnitOfMeasureName": "Grams",  ← required
  "IsProductionBatch": true,
  "ProductionBatchNumber": null,
  "ProductRequiresRemediation": false,
  "RemediationMethodId": null,
  "RemediationDate": null,
  "RemediationSteps": null,
  "PackagedDate": "2019-04-15"  ← required
}

```

[POST/pos/harvests/v1/create](#)

This is your first step in starting a harvest. This call will collect and keep track of the individual plants you will harvest. You must create the harvest before actually harvesting plants.

```

{
  "Name": "HarvestTEST2",        ← required
  "HarvestType": "Product",      ← required
  "DryingRoomId": 18,
  "DryingRoomName": "Flower Room", ← must be existing Room
  "UnitOfWeightName": "Grams",   ← required
  "IsOnHold": false,
  "HarvestStartDate": "2019-04-18", ← required
}

```

```
"FinishedDate": null,  
"ArchivedDate": null,  
"LastModified": null  
}
```

POST/pos/harvests/v1/update

Should you need to make changes to a harvest you've created, you can do so here. You can change the name of the harvest, the room name, unit of weight, etc. You must have an existing harvest and a harvest ID in order to successfully process an update.

```
{  
  "Id": 4,                               ← required existing Harvest Id  
  "HarvestType": "Product",             ← required  
  "DryingRoomId": 14,  
  "DryingRoomName": "Harvest Room",     ← must be existing Room  
  "UnitOfWeightName": "Grams",         ← required  
  "IsOnHold": false,  
  "HarvestStartDate": "2019-04-19",     ← required  
  "FinishedDate": null,  
  "ArchivedDate": null,  
  "LastModified": null  
}
```

POST/pos/harvests/v1/removewaste

During your harvest you will likely accumulate some amount of waste. You will record the weight of the waste using this call. You must have an existing harvest to attribute the waste weight to and can note that by using the harvest ID. Please note that if you put a date in the "FinishedDate" field, your harvest will no longer be considered active.

```
{  
  "Id": 254,                             ← required existing Harvest Id  
  "UnitOfWeightName": "Grams",          ← required  
  "TotalWasteWeight": 10.00,            ← required  
  "FinishedDate": null                  ← ONLY include if you would like to finish the Harvest  
}
```

POST/pos/harvests/v1/finish

When your harvest is complete, you will finish it here. Once a harvest is finished it will appear in the inactive section. You will need to use an Id from an existing harvest.

```
{  
  "Id": 254,                             ← required existing Harvest Id  
  "FinishedDate": "2019-04-15T12:45:02" ← required  
}
```

POST/pos/harvests/v1/unfinish

Unfinish allows you to reactivate a finished harvest. The harvest will appear as active.

```
{  
  "Id": 254                               ← required existing Harvest Id  
}
```

GET/pos/harvests/v1/active

You can use this call to view all the active harvests for a location.

```
{
  "id": 254,
  "name": "HarvestTEST2",
  "harvestType": "Product",
  "dryingRoomId": 18,
  "dryingRoomName": "Flower Room",
  "currentWeight": 180.99,
  "totalWasteWeight": 20,
  "plantCount": 1,
  "totalWetWeight": 200.99,
  "totalRestoredWeight": 0,
  "packageCount": 0,
  "totalPackagedWeight": 0,
  "unitOfWeightName": "Grams",
  "isOnHold": false,
  "harvestStartDate": "2019-03-18T00:00:00",
  "finishedDate": null,
  "archivedDate": null,
  "lastModified": "2019-04-23T18:44:20.9715499"
}
```

GET/pos/harvests/v1/inactive

View any finished or inactive harvests.

```
{
  "id": 14,
  "name": "Harvest1",
  "harvestType": "Product",
  "dryingRoomId": 18,
  "dryingRoomName": "Flower Room",
  "currentWeight": 114.99000000000001,
  "totalWasteWeight": 20,
  "plantCount": 0,
  "totalWetWeight": 150.99,
  "totalRestoredWeight": 0,
  "packageCount": 1,
  "totalPackagedWeight": 16,
  "unitOfWeightName": "Grams",
  "isOnHold": false,
  "harvestStartDate": "2019-03-18T00:00:00",
  "finishedDate": "2019-04-15T12:45:02",
  "archivedDate": null,
  "lastModified": "2019-04-22T19:54:38.4930312"
},
```


GET/pos/harvests/v1/onhold

To view harvests that have been paused or are on hold.

```
{
  "id": 14,
  "name": "Harvest1",
  "harvestType": "Product",
  "dryingRoomId": 18,
  "dryingRoomName": "Flower Room",
  "currentWeight": 114.99000000000001,
  "totalWasteWeight": 20,
  "plantCount": 0,
  "totalWetWeight": 150.99,
  "totalRestoredWeight": 0,
  "packageCount": 1,
  "totalPackagedWeight": 16,
  "unitOfWeightName": "Grams",
  "isOnHold": false,
  "harvestStartDate": "2019-03-18T00:00:00",
  "finishedDate": null,
  "archivedDate": "2019-04-15T12:45:02",
  "lastModified": "2019-04-
22T19:54:38.4930312"
}
```

Packages

In this section we will discuss how to create packages and the different actions you can take once a package is created.

POST/pos/packages/v1/create

Packages can be created multiple places. They can be initiated from plant batches, plants, harvests, and packages. This is the final of the three ways to create a package. You will use this call to make packages from inventory or to add to inventory after an incoming transfer is made.

```
{
  "Label": "packagelabel5",           ← required
  "RoomName": null,                  ← must be existing Room
  "ProductName": "Immature Plant",   ← required
  "Quantity": 4,                     ← required
  "UnitOfMeasureName": "Each",      ← required
  "IsProductionBatch": false,
  "ProductionBatchNumber": null,
  "ProductRequiresRemediation": false,
  "PackagedDate": "2019-04-15",
  "Ingredients": [
    "Label": "PACKAGELABEL1",       ← required if Ingredients are from existing Package
    "Quantity": 10.0,                ← required if Ingredients are from existing Package
    "UnitofMeasureName": "Grams"    ← required if Ingredients are from existing Package
  ]
}
```

POST/pos/packages/v1/create/plantings

A way to add plants to an existing plant batch from an existing package. Could be used for cultivators (growers) who purchase a package of immature plants and are preparing to plant them. The plant batch must already exist in the system, although it can be empty before planting new plants.

```
{
  "Label": "packagelabel5",           ← required
  "Quantity": 2,                     ← required
  "UnitOfMeasureName": "Grams",     ← required
  "PlantBatchName": "PlantBatch1",  ← required existing Plant Batch
  "PlantBatchTypeName": "Clone",
  "PlantCount": 1,                  ← required
  "RoomName": "Planting Room",      ← must be existing Room
  "StrainName": "Strain1",          ← required, must be existing Strain
  "PlantedDate": "2019-04-31T00:00:00Z", ← required
  "FinishedDate": null
}
```

POST/pos/packages/v1/change/item

If you need to update the name of a product within a Package you can do so using this call. Both the Package label and the Item or product must already exist in the system.

```
{
  "Label": "packagelabel5",           ← required existing Package Label
  "ProductName": "Immature Plant"    ← required existing Item
},
```

POST/pos/packages/v1/adjust

When packaging, if you need to make an adjustment to an existing package you can do so with this call. To successfully complete this call, you will need to indicate the label, the quantity, the unit of measure, the reason for an adjustment, and the date.

```
{
  "Label": "packagelabel5",           ← required existing Package Label
  "Quantity": 2,                     ← required
  "UnitOfMeasure": "Grams",          ← required
  "AdjustmentReason": "Drying",
  "AdjustmentDate": "2019-05-05",    ← required
  "AdjustmentNote": null
},
```

POST/pos/packages/v1/finish

When you have completed a package and are ready to sell or transfer, you will finish it here. Finishing a package will make the package inactive.

```
{
  "Label": "packagelabel5",           ← required existing Package Label
  "FinishedDate": "2019-05-06T00:00:00Z" ← required
},
```

POST/pos/packages/v1/unfinish

If a package was finished by mistake, or you need to make adjustments to a finished package, you can make it active again by using the unfinish call. You will need to input the existing label for that package.

```
{
  "Label": "packagelabel5"
},
```

← required existing Package Label

POS/pos/packages/v1/remediate

If an existing package requires remediation due to testing results, you can post information about the remediation steps.

```
{
  "Label": "packagelabel5",
  "RemediationMethodName": "Further Drying",
  "RemediationDate": "2018-08-17",
  "RemediationSteps": "Used sunlight"
}
```

← required existing Package Label

GET/pos/packages/v1/active

Use this call to view the packages active in the system.

```
{
  "id": 380,
  "label": "UNIQUEPACKAGELABEL",
  "sourceHarvestNames": null,
  "roomId": 2,
  "roomName": "Clones Room",
  "quantity": 5,
  "unitOfMeasureName": "Each",
  "unitOfMeasureAbbreviation": "Each",
  "productId": 319,
  "productName": "Immature Plants",
  "productCategoryName": "Flower",
  "packagedDate": "2019-04-23T00:00:00",
  "initialLabTestingState": null,
  "labTestingState": null,
  "labTestingStateDate": null,
  "isProductionBatch": false,
  "productionBatchNumber": null,
  "isTestingSample": false,
  "isProcessValidationTestingSample": false,
  "productRequiresRemediation": false,
  "containsRemediatedProduct": false,
  "remediationMethodName": null,
  "remediationSteps": null,
  "remediationDate": null,
  "receivedFromManifestNumber": null,
  "receivedFromFacilityLicenseNumber": null,
  "receivedFromFacilityName": null,
  "receivedDateTime": null,
}
```

```
"isOnHold": false,
"archivedDate": null,
"finishedDate": null,
"adjustmentReason": null,
"adjustmentDate": null,
"adjustmentNote": null,
"plantBatchId": 271,
"plantBatchName": "Clone Batch April 2019",
"lastModified": null
}
```

GET/pos/packages/v1/inactive

To view packages that are inactive or that have been finished.

```
{
  "id": 47,
  "label": "ABC010",
  "sourceHarvestNames": null,
  "roomId": 0,
  "roomName": null,
  "quantity": -21,
  "unitOfMeasureName": "Each",
  "unitOfMeasureAbbreviation": null,
  "productId": 0,
  "productName": "Immature Plants",
  "productCategoryName": null,
  "packagedDate": "2019-02-02T00:00:00",
  "initialLabTestingState": null,
  "labTestingState": null,
  "labTestingStateDate": null,
  "isProductionBatch": false,
  "productionBatchNumber": null,
  "isTestingSample": false,
  "isProcessValidationTestingSample": false,
  "productRequiresRemediation": false,
  "containsRemediatedProduct": false,
  "remediationMethodName": null,
  "remediationSteps": null,
  "remediationDate": null,
  "receivedFromManifestNumber": null,
  "receivedFromFacilityLicenseNumber": null,
  "receivedFromFacilityName": null,
  "receivedDateTime": null,
  "isOnHold": false,
  "archivedDate": null,
  "finishedDate": "2019-02-12T20:37:11.0427922",
  "adjustmentReason": "wrong units",
  "adjustmentDate": "2019-02-02T00:00:00",
  "adjustmentNote": null,
  "plantBatchId": null,
  "plantBatchName": null,
  "lastModified": "2019-02-12T20:37:11.0427922"
}
```

GET/pos/packages/v1/onhold

Returns all packages that are on hold.

```
{
  "id": 47,
  "label": "packagelabel",
  "sourceHarvestNames": null,
  "roomId": 0,
  "roomName": null,
  "quantity": 21,
  "unitOfMeasureName": "Each",
  "unitOfMeasureAbbreviation": null,
  "productId": 0,
  "productName": "Immature Plants",
  "productCategoryName": null,
  "packagedDate": "2019-02-02T00:00:00",
  "initialLabTestingState": null,
  "labTestingState": null,
  "labTestingStateDate": null,
  "isProductionBatch": false,
  "productionBatchNumber": null,
  "isTestingSample": false,
  "isProcessValidationTestingSample": false,
  "productRequiresRemediation": false,
  "containsRemediatedProduct": false,
  "remediationMethodName": null,
  "remediationSteps": null,
  "remediationDate": null,
  "receivedFromManifestNumber": null,
  "receivedFromFacilityLicenseNumber": null,
  "receivedFromFacilityName": null,
  "receivedDateTime": null,
  "isOnHold": true,
  "archivedDate": "2019-02-12T20:37:11.0427922",
  "finishedDate": null,
  "adjustmentReason": "wrong units",
  "adjustmentDate": "2019-02-02T00:00:00",
  "adjustmentNote": null,
  "plantBatchId": null,
  "plantBatchName": null,
  "lastModified": "2019-02-12T20:37:11.0427922"
}
```

Sales

In this section we will discuss how to send sales receipt information for either external sales or for internal transfers that require sales reporting.

POST/pos/sales/v1/receipts

When you have a receipt from the sale of a package, you use this call to post the data. When entering the Transaction information, the package label must be from an existing package. total amount is the total sales amount for each transaction within the receipt.

```
{
  "SalesDateTime": "2019-03-01T12:44:53.000", ← required
  "SalesCustomerType": "Medicinal", ← required
  "PatientLicenseNumber": null,
  "CaregiverLicenseNumber": null,
  "IdentificationMethod": null,
  "Transactions": [
    {
      "PackageLabel": "brokenpackagelabel", ← required existing Package Label
      "Quantity": 3.0, ← required
      "UnitOfMeasure": "Grams", ← required
      "TotalAmount": 110.00 ← required
    }
  ]
}
```

GET/pos/sales/v1/receipts

To return data for posted receipts, use this call.

```
{
  "id": 51,
  "salesDateTime": "2019-01-21T12:44:53",
  "salesCustomerType": "Transfer",
  "patientLicenseNumber": null,
  "caregiverLicenseNumber": null,
  "identificationMethod": null,
  "createDate": "2019-02-12T20:38:42.3824084",
  "transactions": [
    {
      "id": 51,
      "packageLabel": "packagelabel",
      "quantity": 4,
      "unitOfMeasure": "Each",
      "totalAmount": 37
    }
  ]
}
```

Transfers

In this section we will discuss how to record all transfers in and out of a licensee. Some licensees will have internal transfers due to vertical integration, which still must be recorded. If a transfer is internal, a sales receipt must also be posted to capture the complete transaction of the product.

POST/pos/transfers/v1/incoming

When receiving and incoming transfer, you will use this call.

```
{
  "ManifestNumber": "000000001",           ← required and must be unique
  "ShipperFacilityLicenseNumber": "Cultivator_ABC", ← required
  "ShipperFacilityName": "Fancy Cultivator",     ← required
  "Name": null,
  "TransporterFacilityLicenseNumber": "Distributor_DEF", ← required
  "TransporterFacilityName": "Fancy Distributor", ← required
  "DriverName": "Bill Smith",
  "DriverOccupationalLicenseNumber": null,
  "DriverVehicleLicenseNumber": "CADL 1234",
  "VehicleMake": "Honda",
  "VehicleModel": "CRV",
  "VehicleLicensePlateNumber": "XYZ 987",
  "DeliveryCount": 0,                       ← required, cannot be null
  "ReceivedDeliveryCount": 0,                ← required, cannot be null
  "PackageCount": 2,                         ← required
  "ReceivedPackageCount": 2,                 ← required
  "CreatedDateTime": "2019-04-10",           ← required
  "CreatedByUserName": null,
  "LastModified": null,
  "DeliveryId": 1,                           ← required, cannot be null
  "RecipientFacilityLicenseNumber": "Retailer_GHI", ← required
  "RecipientFacilityName": "Fancy Retailer",    ← required
  "ShipmentTypeName": "Transfer",             ← required
  "ShipmentTransactionType": "Standard",
  "EstimatedDepartureDateTime": null,
  "ActualDepartureDateTime": null,
  "EstimatedArrivalDateTime": null,
  "ActualArrivalDateTime": "2019-04-11T16:49:00.000", ← required
  "DeliveryPackageCount": 0,                  ← required, cannot be null
  "DeliveryReceivedPackageCount": 0,          ← required, cannot be null
  "ReceivedDateTime": "2019-04-11T16:49:19-07:00", ← required
  "Packages": [
    {
      "Label": "packagelabel_12"              ← required unique label
    },
    {
      "Label": "packagelabel_13"              ← required unique label
    }
  ]
}
```

POST/pos/transfers/v1/outgoing

Use this call to post data for outgoing transfers.

```
{
  "ManifestNumber": "0000000002",
  "ShipperFacilityLicenseNumber": "Cultivator_ABC",
  "ShipperFacilityName": "Fancy Cultivator",
  "Name": null,
  "TransporterFacilityLicenseNumber": "Distributor_DEF",
  "TransporterFacilityName": "Fancy Distributor",
  "DriverName": "Bill Smith",
  "DriverOccupationalLicenseNumber": null,
  "DriverVehicleLicenseNumber": "CADL 1234",
  "VehicleMake": "Honda",
  "VehicleModel": "CRV",
  "VehicleLicensePlateNumber": "XYZ 987",
  "DeliveryCount": 0,
  "ReceivedDeliveryCount": 0,
  "PackageCount": 1,
  "ReceivedPackageCount": 1,
  "CreatedDateTime": "2019-04-10",
  "CreatedByUserName": null,
  "LastModified": null,
  "DeliveryId": 0,
  "RecipientFacilityLicenseNumber": "Retailer_GHI",
  "RecipientFacilityName": "Fancy Retailer",
  "ShipmentTypeName": "Sale",
  "ShipmentTransactionType": "Standard",
  "EstimatedDepartureDateTime": null,
  "ActualDepartureDateTime": "2019-04-11T16:49:00.000",
  "EstimatedArrivalDateTime": "2019-04-11T17:49:00.000",
  "ActualArrivalDateTime": null,
  "DeliveryPackageCount": 0,
  "DeliveryReceivedPackageCount": 0,
  "ReceivedDateTime": null,
  "Packages": [
    {
      "Label": "packagelabel_12"
    },
    {
      "Label": "packagelabel_13"
    }
  ]
}
```

← required and must be unique
← required
← required
← required
← required
← required
← required, cannot be null
← required, cannot be null
← required
← required
← required
← required, cannot be null
← required
← required
← required
← required
← required
← cannot be null
← cannot be null
← required unique label
← required unique label

POST/pos/transfers/v1/rejected

If for some reason you cannot accept an incoming transfer, but are required to send information regarding the transfer, use this call.

```
{
  "ManifestNumber": "0000000003",
  "ShipperFacilityLicenseNumber": "Cultivator_ABC",
  "ShipperFacilityName": "Fancy Cultivator",
  "Name": null,
  "TransporterFacilityLicenseNumber": "Distributor_DEF",
  "TransporterFacilityName": "Fancy Distributor",
  "DriverName": "Bill Smith",
  "DriverOccupationalLicenseNumber": null,
  "DriverVehicleLicenseNumber": "CADL 1234",
}
```

← required and must be unique
← required
← required
← required
← required
← required


```

"VehicleMake": "Honda",
"VehicleModel": "CRV",
"VehicleLicensePlateNumber": "XYZ 987",
"DeliveryCount": 0,
"ReceivedDeliveryCount": 0,
"PackageCount": 1,
"ReceivedPackageCount": null,
"CreatedDateTime": "2019-04-10",
"CreatedByUserName": null,
"LastModified": null,
"DeliveryId": 0,
"RecipientFacilityLicenseNumber": "Retailer_GHI",
"RecipientFacilityName": "Fancy Retailer",
"ShipmentTypeName": "Sale",
"ShipmentTransactionType": "Standard",
"EstimatedDepartureDateTime": null,
"ActualDepartureDateTime": null,
"EstimatedArrivalDateTime": null,
"ActualArrivalDateTime": null,
"DeliveryPackageCount": 0,
"DeliveryReceivedPackageCount": 0,
"ReceivedDateTime": null,
"Packages": [
  {
    "Label": "packagelabel_12"
  },
  {
    "Label": "packagelabel_13"
  }
]

```

← required, cannot be null
← required, cannot be null
← required
← required
← required
← required
← cannot be null
← required
← required
← required
← cannot be null
← cannot be null
← required unique label
← required unique label

GET/pos/transfers/v1/incoming

Returns posted Incoming transfers.

```

{
  "id": 3,
  "ManifestNumber": "0000000001",
  "ShipperFacilityLicenseNumber": "Cultivator_ABC",
  "ShipperFacilityName": "Fancy Cultivator",
  "Name": null,
  "TransporterFacilityLicenseNumber": "Distributor_DEF",
  "TransporterFacilityName": "Fancy Distributor",
  "DriverName": "Bill Smith",
  "DriverOccupationalLicenseNumber": null,
  "DriverVehicleLicenseNumber": "CADL 1234",
  "VehicleMake": "Honda",
  "VehicleModel": "CRV",
  "VehicleLicensePlateNumber": "XYZ 987",
  "DeliveryCount": 0,
  "ReceivedDeliveryCount": 0,
  "PackageCount": 1,
  "ReceivedPackageCount": 1,
  "CreatedDateTime": "2019-04-10",
  "CreatedByUserName": null,

```

```

"LastModified": null,
"DeliveryId": null,
"RecipientFacilityLicenseNumber": "Retailer_GHI",
"RecipientFacilityName": "Fancy Retailer",
"ShipmentTypeName": "Transfer",
"ShipmentTransactionType": "Standard",
"EstimatedDepartureDateTime": null,
"ActualDepartureDateTime": null,
"EstimatedArrivalDateTime": null,
"ActualArrivalDateTime": "2019-04-11T16:49:00.000",
"DeliveryPackageCount": 0,
"DeliveryReceivedPackageCount": 0,
"ReceivedDateTime": "2019-04-11T16:49:19-07:00",
"Packages": [
  {
    "Label": "packagelabel_12"
  },
]

```

GET/pos/transfers/v1/outgoing

Returns previously posted outgoing transfers.

```

{
  "id": 4,
  "ManifestNumber": "0000000002",
  "ShipperFacilityLicenseNumber": "Cultivator_ABC",
  "ShipperFacilityName": "Fancy Cultivator",
  "Name": null,
  "TransporterFacilityLicenseNumber": "Distributor_DEF",
  "TransporterFacilityName": "Fancy Distributor",
  "DriverName": "Bill Smith",
  "DriverOccupationalLicenseNumber": null,
  "DriverVehicleLicenseNumber": "CADL 1234",
  "VehicleMake": "Honda",
  "VehicleModel": "CRV",
  "VehicleLicensePlateNumber": "XYZ 987",
  "DeliveryCount": 0,
  "ReceivedDeliveryCount": 0,
  "PackageCount": 1,
  "ReceivedPackageCount": 1,
  "CreatedDateTime": "2019-04-10",
  "CreatedByUserName": null,
  "LastModified": null,
  "DeliveryId": 0,
  "RecipientFacilityLicenseNumber": "Retailer_GHI",
  "RecipientFacilityName": "Fancy Retailer",
  "ShipmentTypeName": "Sale",
  "ShipmentTransactionType": "Standard",
  "EstimatedDepartureDateTime": null,
  "ActualDepartureDateTime": "2019-04-11T16:49:00.000",
  "EstimatedArrivalDateTime": "2019-04-11T17:49:00.000",
  "ActualArrivalDateTime": null,
  "DeliveryPackageCount": 0,
  "DeliveryReceivedPackageCount": 0,
  "ReceivedDateTime": null,
  "Packages": [
    {
      "Label": "packagelabel_12"
    },
  ],
}

```

GET/pos/transfers/v1/rejected

Returns previously rejected transfers associated to a location.

```
{
  "ManifestNumber": "0000000003",
  "ShipperFacilityLicenseNumber": "Cultivator_ABC",
  "ShipperFacilityName": "Fancy Cultivator",
  "Name": null,
  "TransporterFacilityLicenseNumber": "Distributor_DEF",
  "TransporterFacilityName": "Fancy Distributor",
  "DriverName": "Bill Smith",
  "DriverOccupationalLicenseNumber": null,
  "DriverVehicleLicenseNumber": "CADL 1234",
  "VehicleMake": "Honda",
  "VehicleModel": "CRV",
  "VehicleLicensePlateNumber": "XYZ 987",
  "DeliveryCount": 0,
  "ReceivedDeliveryCount": 0,
  "PackageCount": 1,
  "ReceivedPackageCount": null,
  "CreatedDateTime": "2019-04-10",
  "CreatedByUserName": null,
  "LastModified": null,
  "DeliveryId": 0,
  "RecipientFacilityLicenseNumber": "Retailer_GHI",
  "RecipientFacilityName": "Fancy Retailer",
  "ShipmentTypeName": "Sale",
  "ShipmentTransactionType": "Standard",
  "EstimatedDepartureDateTime": null,
  "ActualDepartureDateTime": null,
  "EstimatedArrivalDateTime": null,
  "ActualArrivalDateTime": null,
  "DeliveryPackageCount": 0,
  "DeliveryReceivedPackageCount": 0,
  "ReceivedDateTime": null,
  "Packages": [
    {
      "Label": "packagelabel_12"
    },
    {
      "Label": "packagelabel_13"
    }
  ]
}
```

Appendix A - Definitions

Rooms - Create					
	Example Response	Required	Can be Null	True/ False	Type
"Name"	"Plants Room"	Yes	No	x	string
Rooms - Update					
	Example Response	Required	Can be Null	True/ False	Type
"id"	1	Yes	x	x	integer
"Name"	"Harvest Room"	Yes	No	x	string
Strains - Create					
	Example Response	Required	Can be Null	True/ False	Type
"Name"	"Strain1"	Yes	No	x	string
"TestingStatus"	"None"	No	Yes	x	string
"ThcLevel"	0.5	No	Yes	x	number
"CbdLevel"	0.5	No	Yes	x	number
"IndicaPercentage"	50	No	Yes	x	number
"SativaPercentage"	50	No	Yes	x	number
Strains - Update					
	Example Response	Required	Can be Null	True/ False	Type
"id"	51	Yes	x	x	integer
"Name"	"Strain1"	No	Yes	x	string
"TestingStatus"	"None"	No	Yes	x	string
"ThcLevel"	0.5	No	Yes	x	number
"CbdLevel"	0.5	No	Yes	x	number
"IndicaPercentage"	50	No	Yes	x	number
"SativaPercentage"	50	No	Yes	x	number
Item Category - Create					
	Example Response	Required	Can Be Null	True/ False	Type
"Name"	"Flower"	Yes	x	x	string
"ProductCategoryType"	"Flower"	Yes	x	x	string
"QuantityType"	"WeightBased"	Yes	x	x	string
"DefaultLabTestingState"	"NotSubmitted"	No	Yes	x	string
"RequiresApproval"	"false"	No	x	Yes	boolean
"RequiresAdministrationMethod"	"false"	No	x	Yes	boolean
"RequiresStrain"	"true"	No	x	Yes	boolean
"RequiresCbdPercent"	"true"	No	x	Yes	boolean
"RequiresUnitCbdContent"	"true"	No	x	Yes	boolean
"RequiresThcPercent"	"true"	No	x	Yes	boolean
"RequiresThcContent"	"true"	No	x	Yes	boolean
"RequiresUnitVolume"	"false"	No	x	Yes	boolean
"RequiresUnitWeight"	"false"	No	x	Yes	boolean
"RequiresServingSize"	"false"	No	x	Yes	boolean
"RequiresSupplyDurationDays"	"false"	No	x	Yes	boolean
"UnitQualityMultiplier"	null	No	Yes	x	number (\$double)
"UnitQuantityUnitofMeasureName"	"Grams"	No	Yes	x	string
"RequiresIngredients"	"false"	No	x	Yes	boolean
"RequiresProductPhoto"	"false"	No	x	Yes	boolean
"CanContainSeeds"	"false"	No	x	Yes	boolean
"CanBeRemediated"	"false"	No	x	Yes	boolean

Item Category - Update					
	Example Response	Required	Can be Null	True/ False	Type
"id"	1	Yes	x	x	integer (int32)
"Name"	"Flower"	Yes	x	x	string
"ProductCategoryType"	"Flower"	Yes	x	x	string
"QuantityType"	"WeightBased"	Yes	x	x	string
"DefaultLabTestingState"	"NotSubmitted"	No	Yes	x	string
"RequiresApproval"	"false"	No	x	Yes	boolean
"RequiresAdministrationMethod"	"false"	No	x	Yes	boolean
"RequiresStrain"	"true"	No	x	Yes	boolean
"RequiresCbdPercent"	"true"	No	x	Yes	boolean
"RequiresUnitCbdContent"	"true"	No	x	Yes	boolean
"RequiresThcPercent"	"true"	No	x	Yes	boolean
"RequiresThcContent"	"true"	No	x	Yes	boolean
"RequiresUnitVolume"	"false"	No	x	Yes	boolean
"RequiresUnitWeight"	"false"	No	x	Yes	boolean
"RequiresServingSize"	"false"	No	x	Yes	boolean
"RequiresSupplyDurationDays"	"false"	No	x	Yes	boolean
"UnitQualityMultiplier"	null	No	Yes	x	number
"UnitQuantityUnitofMeasureName"	"Grams"	No	Yes	x	string
"RequiresIngredients"	"false"	No	x	Yes	boolean
"RequiresProductPhoto"	"false"	No	x	Yes	boolean
"CanContainSeeds"	"false"	No	x	Yes	boolean
"CanBeRemediated"	"false"	No	x	Yes	boolean
Items - Create					
	Example Response	Required	Can be Null	True/ False	Type
"ProductCategoryName"	"Other Concentrate (weight)"	Yes	x	x	string
"Name"	"Cookie"	Yes	x	x	string
"UnitofMeasureName"	"Grams"	Yes	x	x	string
"AdministrationMethod"	null	No	Yes	x	string
"StrainName"	"OG Kush"	Yes	x	x	string
"IntermediaryProductUsed"	"true"	Yes	x	Yes	boolean
"UnitCbdPercent"	25	No	Yes	x	number
"UnitCBDContent"	0.25	No	Yes	x	number
"UnitCBDContentUnitOfMeasureName"	null	No	Yes	x	string
"UnitThcPercent"	75	No	Yes	x	number
"UnitThcContent"	0.75	No	Yes	x	number
"UnitThcContentUnifofMeasureName"	null	No	Yes	x	string
"UnitVolume"	null	No	Yes	x	number
"UnitVolumeUnitofMeasureName"	null	No	Yes	x	string
"UnitWeight"	4	No	Yes	x	number
"UnitWeightUnitofMeasureName"	"Grams"	No	Yes	x	string
"ServingSize"	null	No	Yes	x	string
"SupplyDurationDays"	null	No	Yes	x	integer
"Ingredients"	null	No	Yes	x	string
"Intermediaries"	x	Yes (f "Intermediary ProductUsed" is TRUE)	x	x	
"Name"	"Butter"	Yes		x	string
"Type"	"Other Concentrate (weight)"	Yes		x	string
Items - Update					
	Example Response	Required	Can be Null	True/ False	Type
"id"	1	Yes	x	x	integer
"ProductCategoryName"	"Flower"	Yes	x	x	string
"Name"	"Pre-roll"	Yes	x	x	string
"UnitofMeasureName"	"Grams"	Yes	x	x	string
"AdministrationMethod"	null	No	Yes	x	string
"StrainName"	"OG Kush"	Yes	x	x	string
"IntermediaryProductUsed"	"true"	No	x	Yes	boolean
"UnitCbdPercent"	25	No	Yes	x	number
"UnitCBDContent"	0.25	No	Yes	x	number
"UnitCBDConentUnitOfMeasureName"	null	No	Yes	x	string
"UnitThcPercent"	75	No	Yes	x	number
"UnitThcContent"	0.75	No	Yes	x	number
"UnitThcContentUnifofMeasureName"	null	No	Yes	x	string

"UnitVolume"	null	No	Yes	x	number
"UnitVolumeUnitofMeasureName"	null	No	Yes	x	string
"UnitWeight"	null	No	Yes	x	number
"UnitWeightUnitofMeasureName"	"Grams"	No	Yes	x	string
"ServingSize"	null	No	Yes	x	string
"SupplyDurationDays"	null	No	Yes	x	integer
"Ingredients"	null	No	Yes	x	string
"Intermediaries"	x	Yes (If "IntermediaryProductUsed" is TRUE)	x	x	
"Name"	"Flower"	Yes		x	string
"Type"	"Concentrates (Other)"	Yes		x	string
Plant Batches - Create Plantings					
	Example Response	Required	Can be Null	True/ False	Type
"Name"	"Plant Batch 1"	Yes	x	x	string
"Type"	"Clone"	No	x	x	string
"Count"	30	Yes	x	x	integer
"StrainName"	"OG Kush"	Yes	x	x	string
"RoomName"	"Plants Room"	No	Yes	x	string
"PlantedDate"	"2019-03-12"	No	x	x	datetime2
Plant Batches - Change Growth Phase					
	Example Response	Required	Can be Null	True/ False	Type
"Name"	"Plant Batch 1"	Yes	x	x	string
"Count"	15	Yes	x	x	integer
"NewTag"	null	No	Yes	x	string
"GrowthPhase"	"Vegetative"	Yes	x	x	string
"RoomName"	"Plants Room 2"	No	Yes	x	string
"GrowthDate"	"2019-03-30"	No	Yes	x	datetime2
Plant Batches - Create Packages					
	Example Response	Required	Can be Null	True/ False	Type
"PlantBatchID"	1	Yes	x	x	integer
"PlantBatchName"	"Plant Batch 1"	Yes	x	x	string
"Quantity"	15	Yes	x	x	number
"RoomName"	"Packaging Room"	No	Yes	x	string
"ProductName"	"Immature Plants"	Yes	x	x	string
"Label"	"98111101"	Yes	x	x	string
"PackagedDate"	"2019-04-02"	Yes	Yes	x	datetime2
Plant Batches - Destroy					
	Example Response	Required	Can be Null	True/ False	Type
"Name"	"Plant Batch 1"	Yes	x	x	string
"DestroyedCount"	25	Yes	x	x	integer
"DestroyedNote"	"Bugs"	No	Yes	x	string
"DestroyedDate"	"2019-03-14"	Yes	Yes	x	string
Plants - Create Plantings					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"PACKAGELABEL1"	Yes	x	x	string
"PlantBatchName"	"Plant Batch 1"	Yes	x	x	string
"PlantBatchTypeName"	"Manicure"	No	Yes	x	string
"PlantCount"	1	Yes	x	x	integer
"RoomName"	"Harvest Room"	No	Yes	x	string
"StrainName"	"OG Kush"	Yes	x	x	string
"PlantedDate"	"2019-04-15"	No	Yes	x	string
Plants - Move Plants					
	Example Response	Required	Can be Null	True/ False	Type
"Id"	12	Yes	x	x	integer
"Label"	"ABC4524k"	Yes	Yes	x	string
"RoomName"	"Harvest Room"	No	Yes	x	string

Plants - Manicure Plants					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"ABC004432"	Yes	x	x	string
"ManicuredWeight"	102.26	Yes	x	x	number
"ManicuredUnitofWeightName"	"Grams"	Yes	x	x	string
"RoomName"	"Plants Room"	No	Yes	x	string
"HarvestName"	"ManicureHarvest1"	Yes	Yes	x	string
"ManicuredDate"	"2019-04-11"	No	x	x	string
Plants - Change Plant Growth Phase					
	Example Response	Required	Can be Null	True/ False	Type
"Id"	5	Yes	x	x	integer
"Label"	"ACGE558693"	Yes	Yes	x	string
"NewTag"	"KO89332"	Yes	Yes	x	string
"GrowthPhase"	"Vegetative"	No	Yes	x	string
"NewRoom"	"Plants Room"	No	Yes	x	string
"GrowthDate"	"2019-04-09"	No	Yes	x	string
Plants - Harvested Plants					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"AAE0035550"	Yes	x	x	string
"HarvestedWeight"	86.27	Yes	x	x	number
"HarvestedUnitofWeightName"	"Grams"	Yes	x	x	string
"RoomName"	"Harvest Room"	No	Yes	x	string
"HarvestName"	"Harvest_32"	Yes	x	x	string
"HarvestedDate"	"2019-04-16"	No	Yes	x	string
Plants - Destroy Plants					
	Example Response	Required	Can be Null	True/ False	Type
"Id"	4	Yes	x	x	integer
"Label"	"JIE119283000"	No	Yes	x	string
"DestroyedNote"	"Bugs"	Yes	Yes	x	string
"DestroyedDate"	"2018-05-12"	Yes	Yes	x	string
Harvests - Create Packages					
	Example Response	Required	Can be Null	True/ False	Type
"HarvestID"	6	Yes	x	x	integer
"Label"	"RHK43982982001231"	Yes	x	x	string
"RoomName"	"Harvest Room"	No	Yes	x	string
"ProductName"	"Buds"	Yes	x	x	string
"Weight"	114.89	Yes	x	x	number
"UnitofMeasure"	"Grams"	Yes	x	x	string
"IsProductionBatch"	"true"	No	x	Yes	boolean
"ProductionBatchNumber"	null	No	Yes	x	string
"ProductRequiresRemediation"	"false"	No	x	Yes	boolean
"RemediationMethodId"	null	No	Yes	x	integer
"RemediationDate"	"2019-04-22"	No	Yes	x	string
"RemediationSteps"	null	No	Yes	x	string
Harvests - Create Harvest					
	Example Response	Required	Can be Null	True/ False	Type
"Name"	"Harvest 12"	Yes	x	x	string
"HarvestType"	"Product"	Yes	Yes	x	string
"DryingRoomId"	2	No	x	x	integer
"DryingRoomName"	"Harvest Room"	No	Yes	x	string
"UnitofWeightName"	"Grams"	Yes	x	x	string
"IsOnHold"	"false"	No	x	Yes	boolean
"HarvestStartDate"	"2019-02-09"	Yes	x	x	string
"FinishedDate"	null	No	Yes	x	string
"ArchivedDate"	null	No	Yes	x	string
"LastModified"	null	No	Yes	x	string

Harvests - Update Harvest					
	Example Response	Required	Can be Null	True/ False	Type
"Id"	5	Yes	x	x	integer
"HarvestType"	"Product"	Yes	x	x	string
"DryingRoomId"	3	No	Yes	x	integer
"DryingRoomName"	"Harvest Room"	No	Yes	x	string
"UnitofWeightName"	"grams"	Yes	x	x	string
"IsOnHold"	"false"	No	x	Yes	boolean
"HarvestStartDate"	"2019-11-25"	Yes	x	x	string
"FinishedDate"	null	No	Yes	x	string
"ArchivedDate"	null	No	Yes	x	string
"LastModified"	null	No	Yes	x	string
Harvests - Remove Waste					
	Example Response	Required	Can be Null	True/ False	Type
"Id"	2	Yes	x	x	integer
"UnitofWeightName"	"Grams"	Yes	x	x	string
"TotalWasteWeight"	16.84	Yes	x	x	number
"FinishedDate"	"2019-04-16"	No	Yes	x	string
Harvests - Finish					
	Example Response	Required	Can be Null	True/ False	Type
"Id"	1	Yes	x	x	integer
"FinishedDate"	"2019-04-17"	Yes	x	x	string
Harvests - Unfinish					
	Example Response	Required	Can be Null	True/ False	Type
"Id"	12	Yes	x	x	integer
Packages - Create					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"PKU00443922901"	Yes	x	x	string
"RoomName"	"Plants Room"	No	Yes	x	string
"ProductName"	"Flower"	Yes	x	x	string
"Quantity"	28.01	Yes	x	x	number
"UnitofMeasureName"	"Grams"	Yes	x	x	string
"IsProductionBatch"	"false"	No	x	Yes	boolean
"ProductionBatchNumber"	null	No	Yes	x	string
"ProductRequiresRemediation"	"false"	No	x	Yes	boolean
"PackagedDate"	"2019-05-15"	No	x	x	string
"Ingredients"	[[See below]]				
"Label"	"ACBD4920948"	Yes (if from existing package)	Yes	x	string
"Quantity"	8.07	Yes	Yes	x	number
"UnitofMeasureName"	"Grams"	Yes	Yes	x	string
Packages - Create Plantings					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"HKE4800058214"	Yes	x	x	string
"Quantity"	5.61	Yes	x	x	number
"UnitofMeasureName"	"Grams"	Yes	x	x	string
"PlantBatchName"	"Plant Batch 1"	Yes	Yes	x	string
"PlantBatchTypeName"	"Clone"	No	Yes	x	string
"PlantCount"	1	Yes	x	x	integer
"RoomName"	"Plants Room"	No	Yes	x	string
"StrainName"	"Green River"	Yes	x	x	string
"PlantedDate"	"2019-04-16"	Yes	x	x	string
"FinishedDate"	"2019-05-01T00:00:00Z"	No	Yes	x	string
Packages - Change Item					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"ESHO89002838111"	Yes	x	x	string
"ProductName"	"Tincture"	Yes	x	x	string

Packages - Adjust					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"JSK92832810002138"	Yes	x	x	string
"Quantity"	4.35	Yes	x	x	number
"UnitOfMeasure"	"Grams"	Yes	x	x	string
"AdjustmentReason"	"Drying"	No	Yes	x	string
"AdjustmentDate"	"2019-05-08"	Yes	Yes	x	string
"AdjustmentNote"	null	No	Yes	x	string
Packages - Finish					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"ABCE58272000481"	Yes	x	x	string
"FinishedDate"	"2019-05-12"	Yes	x	x	string
Packages - Unfinish					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"ANK409211000029"	Yes	x	x	string
Packages - Remediate					
	Example Response	Required	Can be Null	True/ False	Type
"Label"	"EMW8420999201"	Yes	x	x	string
"RemediationMethodName"	"Further Drying"	No	Yes	x	string
"RemediationDate"	"2019-04-26"	No	Yes	x	string
"RemediationSteps"	"Used fans"	No	Yes	x	string
Sales - Receipts					
	Example Response	Required	Can be Null	True/ False	Type
"SalesDateTime"	"2019-04-24T10:30:23.00"	Yes	x	x	string
"SalesCustomerType"	"Consumer"	Yes	x	x	string
"PatientLicenseNumber"	null	No	Yes	x	string
"CaregiverLicenseNumber"	"CGLIC-9876"	No	Yes	x	string
"IdentificationMethod"	null	No	Yes	x	string
"Transactions"	[[See Below]]				
"Package Label"	"packagelabel"	Yes	x	x	integer
"Quantity"	8.01	Yes	x	x	number
"UnitofMeasure"	"Grams"	Yes	x	x	string
"TotalAmount"	34.61	Yes	x	x	number
Transfers - Post Incoming					
	Example Response	Required	Can be Null	True/ False	Type
"ManifestNumber"	"000001"	Yes	x	x	string
"ShipperFacilityLicenseNumber"	"324-aK321"	Yes	Yes	x	string
"ShipperFacilityName"	"Lofty Med-Cultivation B"	Yes	Yes	x	string
"Name"	null	No	Yes	x	string
"TransporterFacilityLicenseNumber"	"324-aK321"	Yes	x	x	string
"TransporterFacilityName"	"Lofty Med-Dispensary"	Yes	x	x	string
"DriverName"	"Tim"	No	Yes	x	string
"DriverOccupationalLicenseNumber"	"C28938103"	No	Yes	x	string
"DriverVehicleLicenseNumber"	"C28CC81ew03"	No	Yes	x	string
"VehicleMake"	"Jeep"	No	Yes	x	string
"VehicleModel"	"Wrangler"	No	Yes	x	string
"VehicleLicensePlateNumber"	"TXA-204"	No	Yes	x	string
"DeliveryCount"	1	Yes	x	x	integer
"ReceivedDeliveryCount"	0	Yes	x	x	integer
"PackageCount"	8	Yes	x	x	integer
"ReceivedPackageCount"	0	Yes	x	x	integer
"CreatedDateTime"	"2019-04-08T08:25:46.00"	Yes	x	x	string
"CreatedByUserName"	null	No	Yes	x	string
"LastModified"	"2019-04-08T08:25:46.00"	No	Yes	x	string
"DeliveryId"	1	Yes	x	x	integer
"RecipientFacilityLicenseNumber"	"132-BAC"	Yes	x	x	string
"RecipientFacilityName"	"Lofty Med-Cultivation A"	Yes	x	x	string
"ShipmentTypeName"	"Transfer"	Yes	x	x	string
"ShipmentTransactionType"	"Standard"	No	Yes	x	string
"ActualDepartureDateTime"	null	No	Yes	x	string
"EstimatedArrivalDateTime"	null	No	Yes	x	string
"ActualArrivalDateTime"	"2019-04-08T010:35:00.00"	Yes	x	x	string
"DeliveryPackageCount"	0	Yes	x	x	integer
"DeliveryReceivedPackageCount"	0	Yes	x	x	integer
"ReceivedDateTime"	null	Yes	x	x	string
"Packages"	[[See Below]]				
"Label"	"ABCE844372091"	Yes	x		

Transfers - Post Outgoing					
	Example Response	Required	Can be Null	True/ False	Type
"ManifestNumber"	"000002"	Yes	x	x	string
"ShipperFacilityLicenseNumber"	"324-alk321"	Yes	x	x	string
"ShipperFacilityName"	"Lofty Med-Cultivation B"	Yes	x	x	string
"Name"	null	No	Yes	x	string
"TransporterFacilityLicenseNumber"	"324-alk321"	Yes	x	x	string
"TransporterFacilityName"	"Lofty Med-Dispensary"	Yes	x	x	string
"DriverName"	"Tim"	No	Yes	x	string
"DriverOccupationalLicenseNumber"	"C28938103"	No	Yes	x	string
"DriverVehicleLicenseNumber"	"C28CC81ew03"	No	Yes	x	string
"VehicleMake"	"Jeep"	No	Yes	x	string
"VehicleModel"	"Wrangler"	No	Yes	x	string
"VehicleLicensePlateNumber"	"TXA-204"	No	Yes	x	string
"DeliveryCount"	1	Yes	x	x	integer
"ReceivedDeliveryCount"	0	Yes	x	x	integer
"PackageCount"	8	Yes	x	x	integer
"ReceivedPackageCount"	0	Yes	x	x	integer
"CreatedDateTime"	"2019-05-08T08:25:46.00"	Yes	x	x	string
"CreatedByUserName"	null	No	Yes	x	string
"LastModified"	"2019-05-08T08:25:46.00"	No	Yes	x	string
"DeliveryId"	1	Yes	x	x	integer
"RecipientFacilityLicenseNumber"	"132-BAC"	Yes	x	x	string
"RecipientFacilityName"	"Lofty Med-Cultivation A"	Yes	x	x	string
"ShipmentTypeName"	"Sale"	Yes	x	x	string
"ShipmentTransactionType"	"Standard"	No	Yes	x	string
"ActualDepartureDateTime"	"2019-05-08T010:35:00.00"	Yes	x	x	string
"EstimatedArrivalDateTime"	"2019-05-08T010:35:00.00"	Yes	x	x	string
"ActualArrivalDateTime"	null	No	Yes	x	string
"DeliveryPackageCount"	0	Yes	x	x	integer
"DeliveryReceivedPackageCount"	0	Yes	x	x	integer
"ReceivedDateTime"	null	No	Yes	x	string
"Packages"	{{ See Below }}				
"Label"	"ABCE844372091"	Yes	x		
Transfers - Post Rejected					
	Example Response	Required	Can be Null	True/ False	Type
"ManifestNumber"	"000003"	Yes	x	x	string
"ShipperFacilityLicenseNumber"	"324-alk321"	Yes	x	x	string
"ShipperFacilityName"	"Lofty Med-Cultivation B"	Yes	x	x	string
"Name"	null	No	Yes	x	string
"TransporterFacilityLicenseNumber"	"324-alk321"	Yes	x	x	string
"TransporterFacilityName"	"Lofty Med-Dispensary"	Yes	x	x	string
"DriverName"	"Tim"	No	Yes	x	string
"DriverOccupationalLicenseNumber"	"C28938103"	No	Yes	x	string
"DriverVehicleLicenseNumber"	"C28CC81ew03"	No	Yes	x	string
"VehicleMake"	"Jeep"	No	Yes	x	string
"VehicleModel"	"Wrangler"	No	Yes	x	string
"VehicleLicensePlateNumber"	"TXA-204"	No	Yes	x	string
"DeliveryCount"	1	Yes	x	x	integer
"ReceivedDeliveryCount"	0	Yes	x	x	integer
"PackageCount"	8	Yes	x	x	integer
"ReceivedPackageCount"	6	Yes	x	x	integer
"CreatedDateTime"	"2019-03-08T08:25:46.00"	Yes	x	x	string
"CreatedByUserName"	null	No	Yes	x	string
"LastModified"	"2019-03-08T08:25:46.00"	No	Yes	x	string
"DeliveryId"	2	Yes	x	x	integer
"RecipientFacilityLicenseNumber"	"132-BAC"	Yes	x	x	string
"RecipientFacilityName"	"Lofty Med-Cultivation A"	Yes	x	x	string
"ShipmentTypeName"	"Transfer"	Yes	x	x	string
"ShipmentTransactionType"	"Standard"	No	Yes	x	string
"ActualDepartureDateTime"	null	No	Yes	x	string
"EstimatedArrivalDateTime"	"2019-03-08T10:35:00.00"	No	Yes	x	string
"ActualArrivalDateTime"	null	No	Yes	x	string
"DeliveryPackageCount"	8	Yes	x	x	integer
"DeliveryReceivedPackageCount"	8	Yes	x	x	integer
"ReceivedDateTime"	2019-03-08-T10:38:23.00	Yes	x	x	string
"Packages"	{{ See Below }}	x	x	x	x
"Label"	"ABCE844372091"	Yes	Yes		

Appendix B - Troubleshooting

The NCS team has collected several commonly encountered errors with potential answers and fixes in this section.

If you can't find what you're looking for here, please direct your questions to api@ncsanalytics.com

Can't get access to the API?

It is possible your IP Address is not whitelisted. Please contact your administrator, the NCS team will work with them to whitelist your IP address.

API Keys

Keys to the NCS API automatically expire after 6 months of use. NCS will reach out and notify you when your keys are about to expire.

Error Codes

To review, these are the possible error codes that could be generated.

Server Response	Reason
200 OK	The server was able to process the request successfully
400 Bad Request	An error has occurred while executing your request. Check the payload of the call for errors, stray commas, typos, etc.
401 Unauthorized	The token you provided is not authorized or invalid. Or you have not provided your IP address for whitelisting.
404 Not Found	No vendor or location found for submitted request.
406 Not Found	No related data found. Have you already posted strains, plant batches, labels, or other data that is required by the call.
500 Internal Server Error	An error has occurred while executing your request. Payload may be too large. Break up request and re-submit.

Within these code types there can be a variety of different messages. To help you identify possible solutions to these errors, we will review some of the error messages that you could commonly receive. Please note these are only examples, this is not an exhaustive list.

The Error – 400 Bad Request

In the Rooms Collection

POST Create

“Error: The room with this name _____ already exists”

Try this...

Submit a new room name that you have not previously used. Room names should be unique and can only be used once in room creation.

In the Items Collection

POST Create

“Error: The strain with this name _____ does not exist”

Try this...

Select a strain that already exists in your data set. The strain must already exist.

In the Plant Batches Collection

“Error: The room with this name _____ does not exist”

Try this...

Select a room name that already exists in your data set.

In the Plants Collection

POST Create Plants

“Error: You are only allowed to create one plant at a time; please resubmit a valid request.”

Try this...

Be sure there the “PlantCount” field has the number “1”.

POST Harvested Plants

“Error: You cannot harvest a plant using “each” unit of weight from a plant with “grams unit of weight. Please submit a valid request.”

Try this...

When harvesting Plants into and existing Harvest, you must be sure the Unit of Weight Name correlates to the existing Plant Label being harvested. Unit of Weight Name is required.

POST Move Plants

“Error: Not all received plants belong to the authenticated vendor and location. Please submit a valid request. “

Try this...

In this call the ID, Label, and Room Name must all already exist for the call to be successful. Check to be sure you have these fields entered properly.

In the Harvest Collection

POST Update Harvest

“Error: Not all received harvests belong to the authenticated vendor and location, are you missing any harvest Ids? Please resubmit a valid request.”

Try this...

When updating a harvest, be sure you reference an existing Harvest Id.

In the Packages Collection

POST Create Plantings

“Error: You cannot update a package using “grams” unit of measure from a package with “each” unit of weight. Please resubmit a valid request.

Try this...

Be sure the Unit of Measure Name corresponds to the Unit of Measure used in the Package you are referencing.

POST Adjust

“Error: UnitofMeasureName is required for the package: PKG001, please submit a valid request.”

Try this...

Note the fields of the call that are required. Unit of Measure is a required field and should not be left blank.

In any Collection

“Error: Object reference not set to an instance of an object.”

Try this...

A field that should be used from an existing collection is incorrect. If your call requires you to reference an existing item, label, tag, room, etc. be sure they already exist and are spelled correctly.

The Error – 403

In any Collection

“Message”: “User: anonymous is not authorized to perform: execute-api: Invoke on resource: arn: aws: execute-api:us-west-2*****543”

Try this...

This error means you are trying to access without a whitelisted IP address. Please contact api@ncsanalytics.com to add your IP address.

The Error – 404

In any Collection

“Could not find location for submitted vendor”

Try this...

This error is likely populating because one of your locations is not associated with your Vendor profile in the Platform. Please reach out to the system administrator to verify that all locations are properly assigned.

The Error – 406

In any Collection

“Error: Invalid property data types for submitted _____”

Try this...

Is a comma missing after a bracket separating sets of data?

Are there quotations missing around a response?

Are there quotations where there shouldn't be? (i.e. null)

Are you using the correct data types?

The Error – 500

In any Collection

“Error: An error occurred while updated the entries. See the inner exception for details..

Details: The INSERT statement conflicted with the FOREIGN KEY constraint

“FK_pos_harvest_pos_rooms”. The conflict occurred in Database “ devdb”. The statement has been terminated.”

Try this...

Your request may be too large for the server. Try breaking it up into smaller inquiries and resubmitting.

Contact

For assistance with the NCS JSON API, please reach out to NCS at api@ncsanalytics.com