Track & Trace Logistics API

Advanced Shipping Notice (Inbound)

Supports BarTender Track & Trace 12.1 and Later Versions

Contents

Overview

The Advanced Shipping Notice (ASN) API enables external systems to create, manage, monitor, and reconcile inbound shipments in BarTender Track & Trace. This API is intended for integration with warehouse management systems (WMS), enterprise resource planning (ERP) systems, EDI providers, and other external logistics platforms.

In BarTender Track & Trace, an ASN represents the expected content of an inbound shipment. Once created, the ASN becomes available in the Receiving workflow in the Track & Trace platform and mobile application, where operators can validate incoming goods against the expected shipment content. The system then updates ASN status, records received content and supports discrepancy analysis between expected and received items.

Typical use cases

This API supports scenarios such as:

- Creating expected inbound shipments from a WMS or ERP
- Monitoring the progress of receiving operations
- Retrieving received quantities or serialized items
- Comparing expected shipment content with actual received content
- Searching for ASNs by status, destination, or date range
- Deleting or updating ASNs before receiving is completed

1. Base URL

Use the tenant-specific base URL for all requests:

```
https://api.bartender-tt.com
```

2. Authentication

All requests require authentication headers.

Required headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
Content-Type: application/json
```

Header descriptions

Header	Required	Description
`ApiKey`	Yes	API key associated with the tenant and authenticated user or integration.

Header	Required	Description
`Authorization`	Yes	Basic authorization and the x-tenant are used for authorization.
`x-tenant`	Yes	Basic authorization and the x-tenant are used for authorization.
`Content-Type`	For requests with a body	Must be `application/json`.

Example

```
ApiKey: abc123examplekey
Authorization: Basic
x-tenant: eyJhbGciOi
Content-Type: application/json
```

3. Core concepts

3.1 What is an ASN

An Advanced Shipping Notice is an electronic notification that describes the expected contents of an inbound shipment before the shipment is received. In BarTender Track & Trace, the ASN drives the receiving workflow used by operators in the mobile application.

An ASN typically includes:

- A transaction or shipment reference
- A source location
- A destination location
- Expected shipment content
- Optional metadata such as carrier information, tracking number, or estimated arrival date

3.2 ASN lifecycle status

The ASN status indicates the current stage of processing.

Status	Description
`available`	The ASN has been created and is available for receiving.
`in_progress`	Receiving has started but is not complete.
`done`	Receiving has been completed.
`canceled`	The ASN has been canceled and is no longer available for operational processing.

This status model is described in the source model section.

3.3 Content formats

The ASN content can be represented in different formats.

Content format	Description
`tag`	Serialized item-level tracking.
`quantity`	GTIN or PID-level aggregated quantity.
`sku-quantity`	SKU-level aggregated quantity.

This structure is described in the ASN model from the source document.

4. Typical integration workflow

A common integration flow is:

Step 1: Create an ASN

The external system sends expected shipment data to Track & Trace.

```
PUT /logistics/asn
```

Step 2: Search for an ASN

The external system searches for ASNs.

```
POST /logistics/asn/searches
```

Step 3: Monitor ASN status

The external system checks whether receiving is still in progress or completed.

```
GET /logistics/asn/status/{asnId}
```

Step 4: Retrieve an Advanced Shipping Notice

The external system retrieves a specific ASN.

```
GET /logistics/asn/{asnID}
```

Step 5: Retrieve receiving results

Once receiving is complete, the external system retrieves the received content.

```
GET /logistics/asn/result/{asnId}
```

Step 6: Compare expected and received content

The external system retrieves discrepancy data for reconciliation.

```
GET /logistics/asn/compare/{asnId}
```

This workflow is described in the introduction of the source document.

5. Best practices

- Synchronize product master data before creating or updating ASNs.
- Use a stable `transactionId` from the source system for cross-system traceability.
- Use the status endpoint for polling instead of repeatedly retrieving the full ASN payload.
- Use result and comparison endpoints after the ASN reaches `done` for final reconciliation.
- Do not update or delete ASNs that are already being processed unless business rules explicitly allow it.
- Use pagination when searching for ASNs.
- Use a consistent result format across integrations to reduce downstream mapping complexity. These recommendations are reflected across the source endpoint descriptions.

6. Endpoints

6.1 Create an Advanced Shipping Notice

Method and path

```
PUT /logistics/asn
```

Description

Creates a new Advanced Shipping Notice representing an expected inbound shipment in BarTender Track & Trace. Once created, the ASN becomes available in the Receiving workflow of the Track & Trace mobile application. Operators can then validate received goods against the expected shipment content.

When to use

Use this endpoint when:

- A WMS or ERP confirms that a shipment is in transit
- An EDI 856 message has been received and parsed
- A receiving workflow must be prepared before shipment arrival
- An inbound ASN must be created before goods reach the destination site

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
Content-Type: application/json
```

Request body

The request body defines the ASN content and metadata.

Sample request

```
{
  "transactionId": "RECV-002-251009",
  "contentFormat": "quantity",
  "source": "urn:mjx:site:loc:DEMOTT.00004.0",
  "destination": "urn:mjx:site:loc:DEMOTT.00002.0",
  "extensions": {
    "ext1": "val1",
    "ext2": "val2"
  },
  "containers": [
    {
      "content": [
        {
          "format": "quantity",
          "quantity": 2,
          "pid": "03663328100103"
        }
      ]
    }
  ]
}
```

This sample is based on the example present in the source export, normalized for readability.

Request field summary

Field	Type	Required	Description
`transactionId`	string	Recommended	External reference used to associate the ASN with a record in another system.
`contentFormat`	string	Yes	Defines how shipment content is represented: `tag`, `quantity`, or `sku-quantity`.
`source`	string	Yes	Source business location of the shipment.
`destination`	string	Yes	Destination business location where the shipment will be received.
`extensions`	object	No	Optional additional metadata.
`containers`	array	Yes	Defines the expected shipment content and packaging structure.

Success response

201 Created

```
{
  "asnId": 8133384547531811,
  "status": "available"
}
```

The source export indicates a 201 response with the ID of the new ASN.

Possible responses

Code	Meaning
201	ASN created successfully.
204	Request processed successfully with no body returned.
400	Invalid request body or malformed input.
401	Authentication failed or credentials are missing.
404	Related resource not found.
502	Upstream or gateway error.

Notes:

- *Keep* `transactionId` *stable between systems.*
- *Avoid changing ASN content after receiving has started.*
- *Validate product and location references before creating the ASN. These practices are stated in the source endpoint description.*
- *The x-tenant header is required when using Basic authorization.*

6.2 Retrieve an Advanced Shipping Notice

Method and path

```
GET /logistics/asn/{asnId}
```

Description

Retrieves a single ASN by its unique identifier. This endpoint returns ASN header details such as transaction ID, status, source, destination, and expected content structure as stored in BarTender Track & Trace.

When to use

Use this endpoint when:

- Displaying ASN details in an integration or operational application
- Verifying the contents of an ASN after creation or update

- Reviewing source, destination, timestamps, or expected content before receiving begins

Path parameters

Parameter	Type	Required	Description
`asnId`	integer or string	Yes	Unique ASN identifier generated by Track & Trace.

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
```

Sample request

```
GET /logistics/asn/8133384547531811
```

Success response

200 OK


```
{
  "asnId": 8133384547531811,
  "transactionId": "RECV-002-251009",
  "contentFormat": "quantity",
  "creationTime": "2025-10-09T15:20:54.735Z",
  "updateTime": "2025-10-09T15:20:54.798Z",
  "expirationTime": null,
  "lastStatusChange": "2025-10-09T15:20:54.735Z",
  "status": "available",
  "destination": "urn:mjx:site:loc:DEMOTT.00002.0",
  "source": "urn:mjx:site:loc:DEMOTT.00004.0",
  "extensions": {
    "ext1": "val1",
    "ext2": "val2"
  },
  "containers": [
    {
      "content": [
        {
          "format": "quantity",
          "quantity": 2,
          "pid": "03663328100103"
        }
      ]
    }
  ]
}
```


This payload is reconstructed from the example and model information in the source export. The export itself did not provide a clean response sample.

Possible responses

Code	Meaning
200	ASN returned successfully.
400	Invalid request.
401	Authentication failed or credentials are missing.
404	ASN not found.
502	Upstream or gateway error.

Notes:

- *If only the ASN status is needed, use the status endpoint instead of retrieving the full ASN payload. That recommendation appears in the source content.*
- *The x-tenant header is required when using Basic authorization.*

6.3 Retrieve the ASN Status

Method and path

```
GET /logistics/asn/status/{asnId}
```

Description

Retrieves the current lifecycle status of an ASN and the timestamp of the most recent status change. This endpoint provides a lightweight way to monitor receiving progress without retrieving the full ASN payload.

When to use

Use this endpoint when:

- Polling ASN progress during receiving
- Monitoring state transitions
- Checking whether the ASN is complete before retrieving final result or comparison data

Path parameters

Parameter	Type	Required	Description
asnId	integer or string	Yes	Unique ASN identifier generated by Track & Trace.

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
```

Sample request

```
GET /logistics/asn/status/8133384547531811
```

Success response

200 OK

```
{
  "asnId": 8133384547531811,
  "status": "in_progress",
  "lastStatusChange": "2025-10-09T16:14:22.100Z"
}
```

The source identifies this endpoint as returning the current status and `lastStatusChange`, but the export did not provide a readable body example. This sample is therefore illustrative.

Possible responses

Code	Meaning
200	Current ASN status returned successfully.
400	Invalid request.
401	Authentication failed or credentials are missing.
404	ASN not found.
502	Upstream or gateway error.

Notes:

- *Use this endpoint for polling instead of repeatedly retrieving the full ASN payload. The source explicitly recommends this approach.*
- *The x-tenant header is required when using Basic authorization.*

6.4 Retrieve Received Items

Method and path

```
GET /logistics/asn/result/{asnId}
```

Description

Retrieves the items that have been scanned so far for a specific ASN. This endpoint can be used while receiving is still in progress or after receiving has been completed. It supports different result formats, including serialized tag-level output and aggregated quantity output.

When to use

Use this endpoint when:

- Monitoring receiving progress
- Retrieving received items after completion
- Exporting received quantities back to a WMS or ERP
- Retrieving serialized items when tag-level detail is required

Path parameters

Parameter	Type	Required	Description
`asnId`	integer or string	Yes	Unique ASN identifier generated by Track & Trace.

Query parameters

Parameter	Type	Required	Default	Description
`result_format`	string	No	`tag`	Defines whether the response returns serialized tags or aggregated quantity content.

Supported `result_format` values

Value	Description
`tag`	Returns tag-level or serialized item results.
`quantity`	Returns aggregated GTIN or PID quantity results.
`sku-quantity`	Returns aggregated SKU quantity results.

The source states that `result_format` controls tag-level or aggregated product/quantity output. It also recommends using `quantity` or `sku-quantity` for WMS/ERP integrations and `tag` for serialized receiving.

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
```

Sample request

```
GET /logistics/asn/result/8133384547531811?result_format=quantity
```

Success response example: quantity

200 OK

```
{
  "asnId": 8133384547531811,
```

```
  "resultFormat": "quantity",
  "results": [
    {
      "pid": "03663328100103",
      "quantity": 2
    }
  ]
}
```

Success response example: tag

200 OK

```
{
  "asnId": 8133384547531811,
  "resultFormat": "tag",
  "results": [
    {
      "epc": "urn:epc:id:sgtin:0614141.112345.400",
      "hexa": "3034257BF7194E4000000190"
    }
  ]
}
```

These examples are illustrative because the export did not include usable response samples.

Possible responses

Code	Meaning
200	Received items returned successfully.
400	Invalid request.
401	Authentication failed or credentials are missing.
404	ASN not found.
502	Upstream or gateway error.

Note:

- *The x-tenant header is required when using Basic authorization.*

6.5 Compare Received vs Expected

Method and path

```
GET /logistics/asn/compare/{asnId}
```

Description

Compares the received items for an ASN with the original expected ASN content. This endpoint supports reconciliation by identifying matches, missing content, and unexpected content. The source describes these outcomes as matches, unders, and overs.

When to use

Use this endpoint when:

- Receiving has been completed and discrepancies must be identified
- Reconciliation results must be exported to a WMS or ERP
- Receiving accuracy must be validated
- Operational performance must be reviewed at the item or quantity level

Path parameters

Parameter	Type	Required	Description
`asnId`	integer or string	Yes	Unique ASN identifier generated by Track & Trace.

Query parameters

Parameter	Type	Required	Default	Description
`as_quantity`	boolean	No	`false`	Forces comparison at aggregated PID or GTIN quantity level.
`as_sku_quantity`	boolean	No	`false`	Forces comparison at aggregated SKU quantity level.

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
```

Sample request

```
GET /logistics/asn/compare/8133384547531811?as_quantity=true
```

Success response example

200 OK

```
{
  "asnId": 8133384547531811,
  "comparisonFormat": "quantity",
  "matches": [
    {
      "pid": "03663328100103",
      "expected": 2,
      "received": 2
    }
  ],
  "unders": [],
  "overs": []
}
```

This sample is illustrative. The export provided the functional meaning of the endpoint and its query parameters, but not a clean response body.

Possible responses

Code	Meaning
200	Comparison returned successfully.
400	Invalid request.
401	Authentication failed or credentials are missing.
404	ASN not found.
502	Upstream or gateway error.

Notes:

- *Use tag-level comparison for serialized receiving.*
- *Use quantity or SKU-level comparison for ERP or WMS integration.*
- *Do not mix `as_quantity` and `as_sku_quantity` in the same request.*
- *Perform final comparison after ASN status is `done`. These behaviors are described in the source endpoint guidance.*
- *The x-tenant header is required when using Basic authorization.*

6.6 Search for ASNs

Method and path

`POST /logistics/asn/searches`

Description

Searches for ASNs using filter criteria. This endpoint supports retrieval of ASN lists based on status, destination, dates, or other searchable properties. It also supports pagination and result ordering.

When to use

Use this endpoint when:

- Retrieving ASNs available for receiving
- Monitoring ASNs in progress
- Filtering by destination site
- Retrieving completed or canceled ASNs
- Building dashboards or operational views

Query parameters

Parameter	Type	Required	Description
`from`	integer	No	Starting position of the result set.
`size`	integer	No	Maximum number of ASNs returned.

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
Content-Type: application/json
```

Request body

The request body contains search criteria, filters, and optional ordering.

Sample request

```
{
  "filters": [
    {
      "property": "status",
      "operator": "EQ",
      "values": ["available", "in_progress"]
    },
    {
      "property": "destination",
      "operator": "EQ",
      "values": ["urn:mjx:site:loc:DEMOTT.00002.0"]
    }
  ]
}
```

This sample is based on the search example present in the source export, normalized for JSON format.

Success response example

200 OK

```
{
  "from": 0,
  "size": 2,
  "results": [
    {
      "asnId": 8133384547531811,
      "transactionId": "RECV-002-251009",
      "status": "available",
      "destination": "urn:mjx:site:loc:DEMOTT.00002.0"
    },
    {
      "asnId": 8133384547531812,
      "transactionId": "RECV-002-251010",
      "status": "in_progress",
      "destination": "urn:mjx:site:loc:DEMOTT.00002.0"
    }
  ]
}
```

Possible responses

Code	Meaning
200	Matching ASNs returned successfully.
206	Partial content returned because more results are available.
400	Invalid search criteria or malformed body.
401	Authentication failed or credentials are missing.
404	No matching resource or invalid path.
502	Upstream or gateway error.

Notes:

- *Always use* `from` *and* `size` *for pagination in production integrations. The source recommends filtering by destination and status to reduce response size.*
- *The x-tenant header is required when using Basic authorization.*

6.7 Update an ASN

Method and path

```
PUT /logistics/asn/{asnId}
```

Description

Updates an existing ASN in BarTender Track & Trace. This endpoint can be used to modify shipment content, metadata, or status before receiving is completed. It is typically used to correct expected quantities, source or destination details, expiration data, or shipment status before operational processing finishes.

When to use

Use this endpoint when:

- Shipment content changes before receiving begins
- Expected quantities must be adjusted
- Source or destination information must be corrected
- ASN status must be updated, for example to `canceled`
- Expiration time must be set or changed

Path parameters

Parameter	Type	Required	Description
`asnId`	integer or string	Yes	Unique ASN identifier generated by Track & Trace.

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
Content-Type: application/json
```

Request body

Only the fields included in the request body are updated, according to the source description.

Sample request

```
{
  "asnId": "8133384547531811"
  "contentFormat": "quantity",
  "transactionId": "RECV-002-251009",
  "status": "available",
  "destination": "urn:mjx:site:loc:DEMOTT.00002.0",
  "source": "urn:mjx:site:loc:DEMOTT.00004.0",
  "extensions": {
    "ext1": "val1",
    "ext2": "val2"
  },
  "containers": [
    {
      "content": [
        {
          "format": "quantity",
          "quantity": 2,
          "pid": "03663328100103"
        }
      ]
    }
  ]
}
```

This sample is based on the example included in the source export.

Special note for `tag` content format:

The source states that when contentFormat is tag, each content element must include either a hexadecimal value or an EPC value. If only the hexadecimal value is provided, the EPC is generated automatically from the hexadecimal value, and tags are stored internally as EPCs.

Success response example

204 No Content

No response body is returned.

Possible responses

Code	Meaning
200	ASN updated successfully.
204	Request processed successfully with no body returned.
400	Invalid request body or malformed input.
401	Authentication failed or credentials are missing.
404	ASN not found.
502	Upstream or gateway error.

Notes:

- *Avoid updating ASN content once receiving has started.*
- *Keep WMS and Track & Trace data synchronized.*
- *Cancel and recreate the ASN if major structural changes are required. These recommendations are stated in the source.*
- *The x-tenant header is required when using Basic authorization.*

6.8 Delete an Advanced Shipping Notice

Method and path

```
DELETE /logistics/asn/{asnId}
```

Description

Deletes an existing ASN before it is processed. This endpoint is used to remove an expected inbound shipment that should no longer appear in the Receiving workflow. Once deleted, the ASN is no longer available to operators.

When to use

Use this endpoint when:

- A shipment has been canceled before receiving begins
- An ASN was created in error
- The inbound shipment will not occur and should be removed from operational workflows

Path parameters

Parameter	Type	Required	Description
asnId	integer or string	Yes	Unique ASN identifier generated by Track & Trace.

Request headers

```
ApiKey: <your-api-key>
Authorization: Basic
x-tenant: <Tenant Code>
```

Sample request

```
DELETE /logistics/asn/8133384547531811
```

Success response

204 No Content

No response body is returned.

Possible responses

Code	Meaning
204	ASN deleted successfully.
400	Invalid request.
401	Authentication failed or credentials are missing.
404	ASN not found.
502	Upstream or gateway error.

Notes:

- *Confirm ASN status before deleting it. Do not delete ASNs that are already in progress or completed unless business rules allow it. This guidance is stated in the source.*
- *The x-tenant header is required when using Basic authorization.*

7. Models

The original export contained broken schema mappings such as `oas_any_type_not_mapped`. The following model definitions rewrite that content into a customer-readable format based on the visible model descriptions in the source files.

7.1 ASN

Represents an Advanced Shipping Notice returned by the API.

Field	Type	Description
`asnId`	integer or string	Unique ASN identifier generated by Track & Trace.

Field	Type	Description
`contentFormat`	string	Content representation format: `tag`, `quantity`, or `sku-quantity`.
`transactionId`	string	External reference used to associate the ASN with another business system.
`creationTime`	string (date-time)	Timestamp when the ASN was created.
`updateTime`	string (date-time)	Timestamp when the ASN was last updated.
`expirationTime`	string (date-time) or `null`	Optional time after which the ASN is no longer relevant for processing.
`lastStatusChange`	string (date-time)	Timestamp of the most recent ASN status change.
`status`	string	Current lifecycle status: `available`, `in_progress`, `done`, or `canceled`.
`destination`	string	Destination business location.
`source`	string	Source business location.
`extensions`	object	Optional custom metadata.
`containers`	array	Expected shipment content and packaging structure.

The field descriptions above are based on the model section visible in the Word export.

7.2 ASN Status

Represents the lightweight status response for an ASN.

Field	Type	Description
`asnId`	integer or string	Unique ASN identifier.
`status`	string	Current lifecycle status.
`lastStatusChange`	string (date-time)	Timestamp of the last status transition.

This aligns with the purpose of the status endpoint described in the source.

7.3 Container

Represents a container or grouping of shipment content.

Field	Type	Description
`content`	array	Array of shipment content elements contained within the container.

7.4 Content element

Represents an expected or received content entry.

Field	Type	Description
`format`	string	Representation format of the content element.
`quantity`	number	Quantity for aggregated quantity-based representations.
`pid`	string	Product identifier used for PID or GTIN-based quantity content.
`sku`	string	SKU used for SKU-quantity content.
`epc`	string	EPC value used for tag-based content.
`hexa`	string	Hexadecimal tag representation used for tag-based content.

This structure is inferred from the examples and visible model descriptions in the source files.

7.5 Search request

Represents the ASN search request body.

Field	Type	Description
`filters`	array	List of filter objects that define search criteria.
`order`	object or array	Optional ordering criteria for the result set.

7.6 Filter object

Field	Type	Description
`property`	string	Field name to filter on, such as `status` or `destination`.
`operator`	string	Comparison operator, such as `EQ`.
`values`	array	Values used by the filter.

8. Error handling

The source export lists response status codes for each endpoint but does not provide clean error body schemas. The following error structure is recommended for customer-facing documentation and should be validated against the implementation before publication.

Recommended error body

```
{
  "error": "Bad Request",
  "message": "The request body is invalid.",
  "details": [
    {
      "field": "destination",
      "issue": "Destination is required."
    }
  ]
}
```

Common response codes

Code	Meaning
400	The request is invalid or malformed.
401	Authentication failed or required credentials were not provided.
404	The requested ASN or related resource could not be found.
502	An upstream or gateway error occurred while processing the request.

These codes are repeated across the source endpoint definitions.

9. Example end-to-end flow

9.1 Create an ASN

```
PUT /logistics/asn
{
  "transactionId": "RECV-002-251009",
  "contentFormat": "quantity",
  "source": "urn:mjx:site:loc:DEMOTT.00004.0",
  "destination": "urn:mjx:site:loc:DEMOTT.00002.0",
  "containers": [
    {
      "content": [
        {
          "format": "quantity",
          "quantity": 2,
          "pid": "03663328100103"
        }
      ]
    }
  ]
}
```

9.2 Search for ASN

```
POST /logistics/asn/searches
{
  "filters": [
    {
      "property": "status",
      "operator": "EQ",
      "values": ["available", "in_progress"]
    },
    {
      "property": "destination",
      "operator": "EQ",
      "values": ["urn:mjx:site:loc:DEMOTT.00002.0"]
    }
  ]
}
```

9.3 Check status

```
GET /logistics/asn/status/8133384547531811
{
  "asnId": 8133384547531811,
```

```
  "status": "done",
  "lastStatusChange": "2025-10-09T16:25:10.000Z"
}
```

9.4 Retrieve an Advanced Shipping Notice

```
GET /logistics/asn/{asnID}
{
  "asnId": 8133384547531811,
  "contentFormat": "quantity",
  "transactionId": "RECV-002-251009",
  "creationTime": "2025-10-09T15:20:54.735Z",
  "updateTime": "2025-10-09T15:20:54.798Z",
  "expirationTime": null,
  "lastStatusChange": "2025-10-09T15:20:54.735Z",
  "status": "available",
  "destination": "urn:mjx:site:loc:DEMOTT.00002.0",
  "source": "urn:mjx:site:loc:DEMOTT.00004.0",
  "extensions": {
    "ext1": "val1",
    "ext2": "val2"
  },
  "containers": [
    {
      "content": [
        {
          "format": "quantity",
          "quantity": 2,
          "pid": "03663328100103"
        }
      ]
    }
  ]
}
```

9.5 Retrieve results

```
GET /logistics/asn/result/8133384547531811?result_format=quantity
{
  "asnId": 8133384547531811,
  "resultFormat": "quantity",
  "results": [
    {
      "pid": "03663328100103",
      "quantity": 2
    }
  ]
}
```

9.6 Compare expected and received

```
GET /logistics/asn/compare/8133384547531811?as_quantity=true
{
  "asnId": 8133384547531811,
  "comparisonFormat": "quantity",
  "matches": [
    {
      "pid": "03663328100103",
      "expected": 2,
      "received": 2
```

```
    }
  ],
  "unders": [],
  "overs": []
}
```

Related Documentation

Technical Documents

To view and download technical documents, visit:

https://www.bartendersoftware.com/resources/library

User Guides

Support Articles

- Getting Started with Track & Trace
- Getting an Overview with the Track & Trace Homepage
- Setting Up the Track & Trace Mobile App
- Keep Track of Inventory Assets with Track & Trace

Documentation

- Reviewing Assets in Track & Trace
- Managing Track & Trace
- Working with Track & Trace

Other Resources

Please visit the BarTender website at https://www.bartendersoftware.com.