Title: Specification: Lane JSON
ID: Validaide_QMS_SPEC-2
Version: 1.1
Effective: Mar 13, 2024 12:23:27 (UTC)

LANE JSON
VERSION 1.3
SPECIFICATION

	Name	Role	Digitally signed
Author	Serena Toto	Product Owner	Mar 13, 2024 12:19:57 (UTC)
Reviewer	Mark Bijl	CTO	Mar 13, 2024 12:20:53 (UTC)
Approver	Serena Toto	Product Owner	Mar 13, 2024 12:22:13 (UTC)
QA	Anastasia Themistocleous	Head of Capabilities Platform	Mar 13, 2024 12:22:55 (UTC)

Table of Contents

1. Introduction

1.1. Purpose

This document specifies the Lane JSON format as used in the Lane download / export functionality of Validaide. The same Lane JSON format is applicable for both your own Lanes, as well as Lanes that are shared with you - so-called 'Shared Lanes'.

1.2. Scope

The scope of this document is restricted to the details of the Lane JSON format.

1.3. Related Documents

1.4. Definitions

DG	Dangerous Goods
JSON	Is a shorthand that stands for Javascript Object Notation. It is a common and light-weight format used in web development.
PAT	Predicted Ambient Temperature, a feature from Validaide that – based on a Lane Configuration and historical climate / weather data – predicts the expected Ambient Temperature
UUID	Universal Unique Identifier

2. Obtaining JSON Lane Files

There are two ways of obtaining Lane JSON files:

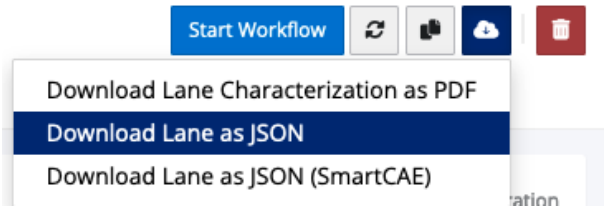
- You can download a JSON file of a Lane
- You can download a Lane Data Export in which Lane JSON files reside of all your Lanes

2.1. Download a single Lane JSON file

Users can download a JSON file of a Lane. A common use case for this type of download would be to share the Lane information with e.g. a packaging provider who can then use the Predicted Ambient Temperature (PAT) to recommend a packaging solution and possible simulate the product temperature.

You can obtain a single Lane JSON as follows:

1. Log in to Validaide
2. Navigate to an individual Lane
3. Press the download button in the upper right corner
4. Select 'Download Lane as JSON' (see below)



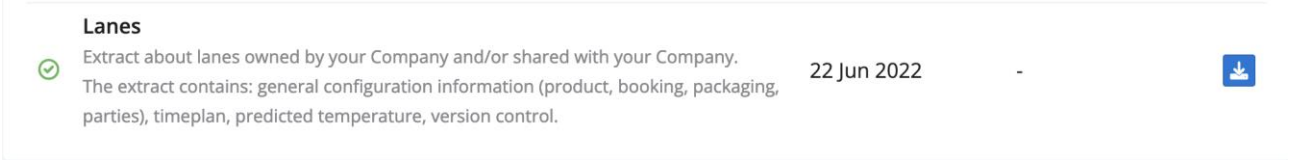
2.2. Download a Lane Data Export containing all Lane JSON files

(Only available if you have the Validaide Premium Membership, which includes Data Exports)
(Only available as of Validaide 5.2.0)

Administrators also have the possibility to obtain a Lane Data Export from the administration console in Validaide. This is a ZIP file that is generated periodically and contains a Lane JSON file for every Lane in your environment.

You can obtain a Lane Data Export as follows:

1. Log in to Validaide
2. Navigate to the Administration section
3. Navigate to the 'Data Exports' section
4. In the Lane Export section, press the download button



3. Concepts

To better interpret the content of the Lane JSON file, it is important to understand several concepts.

- General Information
- Timeplan
- Predicted Ambient Temperature
- Version Control

3.1. General Information

The Lane JSON contains general configuration information about the Lane, such as its name, information about the product, booking, and packaging used, but also which parties and physical locations are involved in the actual transport.

3.2. Timeplan

The timeplan is the transport 'blueprint' and represents the kinds of transport steps that take place and for each step what the process looks like. We refer to the Lane Timeplan consisting of zero or more Steps, and each Step consisting of zero or more Phases.

Depending on the type of Step additional information might be present for that Step. For example, a Step of Type `transit_air` is a Step representing transportation by plane (or RFS) and will therefore contain information about the Flight.

A Step in its turn can consist of smaller stages. We refer to these stages as the 'process' taking place during that Step, and each of those stages as a Phase. For example, a Step of type `storage_warehouse` will have a process at the selected warehouse consisting of (1) the offloading of cargo (i.e. from a truck), (2) the storage of the cargo, and (3) the loading of the cargo (i.e. into a truck).

One Lane can have zero or more Steps, and one Step can have zero or more Phases as part of its process. The sequence of these Steps and Phases is relevant, as they represent the business logic, for example, you cannot have a 'road' step followed immediately by an 'ocean' step without some sort of handling in between. It is especially at these moments during the transportation where risks of temperature deviation are the highest.

3.3. Predicted Ambient Temperature (PAT)

This section contains the *Predicted Ambient Temperature (PAT)* as calculated by Validaide for the Lane for a given month of the year. The PAT is calculated based on the lane configuration and where – geographically speaking – the process is taking place.

Depending on the geographical coordinate, climate data is used to determine the outside temperature as a starting point of the calculation. Following that, the Lane configuration is then used to transform the outside temperature to a prediction, incorporating information such as the duration of a process, whether a process is taking place indoor or outdoor, whether the process takes place in a temperature-controlled environment, and so forth.

3.4. Version Control

In case a Lane has its Version Control module enabled, the JSON also contains a section about the workflow history, detailing who performed what action as part of the Lane workflow.

4. Specification

The tables below provide an overview of the information that is available in the Lane JSON file and how it should be interpreted. The table headers used are as follows:

- **Field** – The name of the field in the JSON entity
- **R/O** – Indicates if the field is Required or Optional
- **N** – Indicates if the field can be Null or not
- **Type** – Indicates the type of the field, either a basic type (e.g. string) or another entity
- **Description** – Additional relevant information about the field
- **V** – Indicates in which version the field was added if not the main version
- **D** – Indicates the field is deprecated and the version where the field will be removed

- NOTE: Validaide currently supports only English, which means that all fields such as country names, questions, labels, etc. will be displayed in English.

4.1. Metrics

With regards to metrics, the following applies:

- **Temperature**
  Is defined in degrees Celsius, unless indicated otherwise
- **Distance**
  Is defined in meters, unless indicated otherwise
- **Duration**
  Is defined in minutes, unless indicated otherwise

4.2. Coordinate

The `coordinate` is an array containing two floats representing the latitude and longitude of a geographical coordinate respectively. For example:

```
[47.433037, 19.2616]
```

4.3. String Representations

4.3.1. Country Code

The `country_code` is a string representing a ISO 3166-1 2-character country code. For example:

```
NL
```

4.3.2. Date

The `date` is a string representing a date. It is in the (European) format of “Y-m-d”. For example:

```
2022-01-13
```

4.3.3. DateTime

The `datetime` is a string representing a datetime. It is in the ISO 8601 format. For example:

```
2004-02-12T15:19:21+00:00
```

4.3.4. Dangerous Goods (DG) Classification

The `dg_classification` is a string that indicates a Dangerous Goods (DG) classification. The format is:

```
{category}[.{subcategory}]
```

where the *category* and *subcategory* are numbers referencing the standard DG classes and sub-classes. The subcategory is optional. For example:

`1.1` – means Category 1, Sub Category 1

`2` – means Category 2

See: https://www.hse.gov.uk/cdg/manual/classification.htm

4.3.5. Days of the Week

The `days_of_the_week` is a 7-character string to indicate days of the week that some activity is taking place. An example would be the days of operation of a flight. The format is:

```
MTWTFSS
```

Here M stands for Monday, T for Tuesday, etc. They are all numbers of either “1” or “0”. We use “1” to indicate that an activity *is* taking place and “0” to indicate that an activity *is not* taking place.

For example:

`0000000` – never

`1111111` – every day

`1000001` – only Monday and Sunday

4.3.6. Encoded Polyline

The `encoded_polyline` is a string that contains an encoded polyline. The encoding is based on Google’s encoding standard, as defined here:

https://developers.google.com/maps/documentation/utilities/polylinealgorithm

4.3.7. IATA2

The `iata2` is a 2-character string representing the IATA code of an Airline. For example:

```
EY
```

4.3.8. IATA3

The `iata3` is a 3-character string representing the IATA code of an Airport. For example:

```
AMS
```

4.3.9. Temperature Range

The `temperature range` is a string that indicates a temperature range. The format is:

`{minimum}*{maximum}`

Here *minimum* and *maximum* are (signed, when negative) numbers indicating a degree Celsius. A special character is the keyword 'inf' meaning infinite meaning that no minimum or maximum is defined. For example:

`2*8` – means between +2 and +8

`inf*-20` – means -20 or lower

`15*25` – means between +15 and +25

4.3.10. First shipment date

The `first shipment date` is a date that indicates a proposed first shipment. It is in the (European) format of "Y-m-d". For example:

`2024-02-13`

4.3.11. Temperature monitors

The `temperature monitors` is a Boolean value that represents temperature monitor usage. The following values are possible:

- true
- false

4.4. Enumerations

4.4.1. CommodityType

The `CommodityType` is a string enumeration representing the type of Commodity. The following values are possible:

- pharameuticals
- perishables
- dangerous_goods
- live_animals
- high_value

> - NOTE: Validaide currently only supports the Commodity 'pharmaceuticals' for the Lane Configuration.

4.4.2. PackagingType

The `PackagingType` is a string enumeration representing the type of packaging used for transportation. The following values are possible:

- none

- thermal_cover
- thermal_system_passive
- container_passive
- liquid_nitrogen container_active
- container_active:electro
- container_active:dry_ice

4.4.3.ContainerType

The `ContainerType` is a string enumeration representing the type of container used for transportation. The following values are possible:

- reefer
- dry

4.4.4. TransportMode

The `TransportMode` is a string enumeration representing the transport mode used for transportation. The following values are possible:

- unknown
- air
- barge
- ocean
- rail
- road

4.4.5.StepType

The `StepType` is a string enumeration representing the type of Step the Lane Configuration. The Step Type indicates the sort of stage the transportation is in. Based on the type of Step, certain information can be included or excluded. There is a distinction between 'transit' Steps that indicate that transport is taking place (e.g., by air, road, etc.) and non-transit steps. The non-transit steps in turn represent a physical location where cargo is picked up, stored, or dropped off. The following values are possible:

- transit_air
- transit_ocean
- transit_road
- storage_warehouse
- storage_handling_origin
- storage_handling_destination
- storage_handling_transfer
- port_handling_origin
- port_handling_destination
- port_handling_transfer
- point_start
- point_finish

4.4.6.PhaseType

The `PhaseType` is a string enumeration representing the type of (Process) Phase taking place at or during a particular Step of the Lane Configuration. The Phase Type indicates the sort of activity that is taking place. The following values are possible:

- transit_air
- transit_barge
- transit_ocean
- transit_rail
- transit_road
- transit_border_crossing
- transit_ferry_crossing
- transit_ferry_waiting
- transit_parking_stop
- container_storage
- container_truck_to_plugins
- container_veseel_to_plugins
- container_plugins_to_vessel
- container_plugins_to_truck
- loose_cargo_offloading
- loose_cargo_loading
- loose_cargo_storage
- warehouse_activity
- uld_storage
- uld_loading
- uld_offloading
- uld_buildup
- uld_breakdown
- tarmac_loading
- tarmac_offloading
- tarmac_cargo_loaded

4.4.7. ExposureType

The `ExposureType` is a string enumeration indicating whether a product is exposed to ambient temperature. The following values are possible:

- no_exposure
- exposed
- optional_exposure
- unknown_exposure

4.4.8. LaneStatus

The `LaneStatus` is a string enumeration indicating the workflow status of a Lane. The following values are possible:

- disabled
- in_progress

- under_review_internal
- under_review_external
- approved
- about_to_expire
- expired

4.4.9. ProductCategory

The `ProductCategory` is a string enumeration indicating the category a product. The following values are possible:

- biopharmaceuticals
- vaccines
- api
- diagnostics
- medical_devices
- consumer_health
- animal_health
- cell_and_gene_therapy
- standard pharmaceuticals
- other

4.4.10. TemperaturePredictionType

The `TemperaturePredictionType` is a string enumeration indicating the type a temperature prediction. The following values are possible:

- climate
- historical_weather

4.4.11. ReferenceType

The `ReferenceType` is a string enumeration indicating the type of Company Reference. The following values are possible:

- customer
- supplier

4.5. Entities

4.5.1. Lane

The Lane entity encapsulates all the Lane configuration information, including the temperature predictions, as well as information regarding the version control / workflow of the Lane.

Field	R	N	Type	Description
id	R	–	`string`	Unique identifier
name	R	–	`string`	Name of the Lane
active	R	–	`boolean`	Lane is actively being used for transport
reference	R	N	`string`	Reference of the Lane
incoterms	R	N	`string`	Incoterms of the Lane
comment	R	N	`string`	A plain text comment
containerType	R	N	`enum<ContainerType>`	The Container Type
transportMode	R	–	`enum<TransportMode>`	The Transport Mode
booking	R	–	`BookingInfo`	Lane Booking Information
product	R	–	`ProductInfo`	Lane Product Information
packaging	R	–	`PackagingInfo`	Lane Packaging Information
parties	R	–	`PartiesInfo`	Lane Parties Information
locations	R	–	`LocationsInfo`	Lane Locations Information
timeplan	R	–	`Step[]`	Lane Timeplan
temperaturePrediction	R	–	`TemperaturePrediction[]`	Lane PAT
versionControl	O	–	`VersionControl`	Lane Version Control

4.5.2. Address

The **`Address`** entity encapsulates address information. The Validaide software will attempt to locate the latitude and longitude corresponding to the address information using geocoding services, but this information is not guaranteed to be available.

Field	R	N	Type	Description
street	R	N	`string`	Street name
streetNumber	R	N	`string`	Street number
building	R	N	`string`	Name of the building
office	R	N	`string`	Name of the office inside a building
city	R	–	`string`	Name of the city
postcode	R	N	`string`	Postal code
countryCode	R	–	`country_code`	The ISO 3166-1 country code
country	R	–	`string`	Name of the country
latitude	R	N	`float`	Latitude related to this address
longitude	R	N	`float`	Longitude related to this address

4.5.3. Airport

The **`Airport`** entity encapsulates information about an airport.

Field	R	N	Type	Description
name	R	–	`string`	Name of the Airport
iata	R	N	`iata3`	IATA code of the Airport
icao	R	N	`string`	ICAO code of the Airport
faa	R	N	`string`	FAA code of the Airport
latitude	R	–	`float`	Latitude of the Airport
longitude	R	–	`float`	Longitude of the Airport
elevation	R	N	`int`	Elevation of the Airport (in feet)

timezone	R	-	`string`	Timezone of the Airport (IANA timezone)

4.5.4. Airline

The **`Airline`** entity defines information about an Airline that is handled by this Location. It contains the information as listed below.

Field	R	N	Type	Description
name	R	-	`string`	Name of the Airline
iata	R	-	`iata2[]`	IATA2 codes linked to this Airline

4.5.5. BookingInfo

The **`BookingInfo`** entity defines the Booking details of a Lane.

Field	R	N	Type	Description
temperatureRange	R	N	`temperature_range`	The booked temperature range
shippingDays	R	-	`days_of_the_week`	The days of the week this Lane is executed

4.5.6. PackagingInfo

The **`PackagingInfo`** entity defines the packaging details of a Lane.

Field	R	N	Type	Description
type	R	N	`enum<PackagingType>`	The type of packing used
innerDimension	R	N	`string`	Inner dimensions of the Packaging
refrigerantUsed	R	N	`string`	Type of Refrigerant used
description	R	N	`string`	Instructions or comments
shipmentRequirements	R	N	`string`	Any additional requirements when shipping
viability	R	N	`int`	Packaging Viability (in hours)

4.5.7. ProductInfo

The **`ProductInfo`** entity defines the product details of a Lane.

Field	R	N	Type	Description
temperatureRange	R	N	`temperature_range`	The Product Temperature Range
dgClassification	R	N	`enum<DGClassification>`	The DG classification
dgDescription	R	N	`string`	The DG description
commodity	R	-	`enum<CommodityType>`	The Commodity Type
category	R	N	`enum<ProductCategory>`	The Product Category

4.5.8. CompanyReference

The **`CompanyReference`** entity defines summary information about a Company. It contains the information as listed below.

A `CompanyReference` contains the fields ‘reference’ and ‘referenceType’ that are intended to be used to link this Company to your organization’s systems. Depending on the context, we might be referencing one of your Suppliers, or one of your Customers. Most of the time, it will be a supplier reference, but, for example, for the Lane Parties configuration, the ‘customer’ party will be a ‘customer’ reference.

Field	R	N	Type	Description	V
id	R	-	`string`	Unique identifier of the Company	
name	R	-	`string`	Name of the Company	
reference	R	N	`string`	Reference of your organization for this Company	

referenceType	R	–	enum<ReferenceType>	The type of reference	1.1

4.5.9. LocationReference

The **LocationReference** entity defines summary information about a Location. It contains the information as listed below.

Field	R	N	Type	Description
id	R	–	string	Unique identifier of the Location
name	R	–	string	Name of the Location
reference	R	N	string	Reference of the Location
owningCompany	R	–	CompanyReference	The Company who owns this Location

4.5.10. Parties

The **Parties** entity defines information about Companies used with Parties

Field	R	N	Type	Description
forwarder	R	N	CompanyReference	The Company responsible for this role
customer	R	N	CompanyReference	""
primaryCarrier	R	N	CompanyReference	""
packaging	R	N	CompanyReference	""
fourPL	R	N	CompanyReference	""

4.5.11. UserReference

The **UserReference** entity defines summary information about a User.

Field	R	N	Type	Description
name	R	–	string	Name of the User
email	R	–	string	Email of the User
jobTitle	R	N	string	The Job Title defined by the User
owningCompany	R	–	CompanyReference	The Company who owns this Location

4.5.12. Locations

The **Locations** entity defines information about relevant physical locations (e.g. distribution centers, offices) that have a specific role in the Lane. We distinguish between:

- **Responsible Location**
  The Location who is responsible for maintaining the Lane
- **Origin Location**
  The Location at the origin responsible for export
- **Destination Location**
  The Location at the destination responsible for import
- **Customer Origin Location**
  The Customer Location at the origin responsible for export
- **Customer Destination Location**
  The Customer Location at the destination responsible for import

Field	R	N	Type	Description
responsbileLocation	R	N	LocationReference	The Location responsible for this role
originLocation	R	N	LocationReference	""

destinationLocation	R	N	LocationReference	""
customerOriginLocation	R	N	LocationReference	""
customerDestinationLocation	R	N	LocationReference	""

4.5.13. Step

The **Step** entity encapsulates a specific 'step' of a Lane. Effectively a Lane is configured with zero or more Steps where the sequence is important, as they are sequential. Each Step represents a certain stage during the transportation process, for example one step can represent a road transport stage. However, a road transport step can only be preceded by either a pickup or some other kind of warehouse storage.

Depending on the type of Step, extra step-specific data can be available. For example, a Step representing transport by air will have information about the flight being used, but this information will not be available for other steps.

It is also worth noting that a specific Step can contain additional smaller stages which we refer to as the 'process' and the process consists of zero or more phases. A **Phase** represents a specific activity taking place at a physical location or during the transportation process, for example there is a 'cargo offloading' phase at a Warehouse step. The duration of the step is always equal to the sum of the duration of the phases.

Field	R	N	Type	Description
position	R	–	int	The position (sequence) of the Step
type	R	–	enum<stepType>	String Enumuration: StepType
duration	R	–	int	The duration of this Step
process	R	–	Phase[]	Collection of phases taking with this Step
point	O	–	StepPoint	Only present when StepType is: - point_start - point_finish
storage	O	–	StepStorage	Only present when StepType is: - storage_warehouse - storage_handling_origin - storage_handling_destination - storage_handling_transfer
transitAir	O	–	StepTransitAir	Only present when StepType is: - transit_air
transitRoad	O	–	StepTransitRoad	Only present when StepType is: - transit_road
transitOcean	O	–	StepTransitOcean	Only present when StepType is: - transit_ocean

4.5.14. StepPoint

The **StepPoint** entity encapsulates information that is specific to a Point Step.

Field	R	N	Type	Description
location	R	N	LocationReference	The Location where cargo is picked up / dropped off
address	R	–	Address	
coordinate	R	N	coordinate	

4.5.15. StepStorage

The **StepStorage** entity encapsulates information that is specific to a Storage Step.

Field	R	N	Type	Description

location	R	N	LocationReference	The physical location where cargo is stored
address	R	–	Address	
coordinate	R	N	coordinate	

4.5.16. StepTransitAir

The **StepTransitAir** entity encapsulates information that is specific to an Air Step.

An Air Step always has one Flight on which the Lane Timeplan and Assessment are based, but multiple Flights can be selected that are documented as allowed alternatives for this Lane Assessment. For example, you can have flight QR 123 from AMS to DOH that leaves at 10:00 AM on Mondays & Tuesdays, and flight QR 456 from AMS to DOH that leaves at 09:30 AM on Wednesdays. Both flights are approved for the Lane, but only one of them is used as 'reference' on which the Lane Timeplan and Assessment are based.

Field	R	N	Type	Description	V
packagingType	R	N	enum<PackagingType>	The type of Packaging used, if different	
carrier	R	–	CompanyReference	The Airline Carrier	
cargoHoldTemperature	R	–	int	Temperature, when non-temperature controlled	
flight	R	–	Flight	The Reference Flight, i.e. Flight used for the Lane Assessment and Timeplan	
Flights	R	–	Flight[]	Flights that have been selected as possible allowed alternatives for this Air Step	1.1

4.5.17. StepTransitRoad

The **StepTransitRoad** entity encapsulates information that is specific to a Road Step.

Field	R	N	Type	Description
packagingType	R	N	enum<PackagingType>	The type of Packaging used, if different
route	O	N	Route	
carrier	R	N	CompanyReference	
carrierLocation	R	N	LocationReference	
carriersAllowed	R	–	CompanyReference[]	
carrierLocationsAllowed	R	–	LocationReference[]	

4.5.18. StepTransitOcean

The **StepTransitOcean** entity encapsulates information that is specific to an Ocean Step.

Field	R	N	Type	Description
packagingType	R	N	enum<PackagingType>	The type of Packaging used, if different
route	O	N	Route	
carrier	R	N	CompanyReference	

4.5.19. TemperaturePrediction

The **TemperaturePrediction** contains a Predicted Ambient Temperature (PAT) of the Lane for a given month. Typically, only the months January and July are included as they are considered to be the default 'extreme' scenarios.

The content of this field is **empty** by default for the files included in the **Lane Data Export** (2.2). In order to include the temperature data, the specific option *("Include the temperature prediction data?")* should be selected to *Yes* in the *Modify Export Settings*, under *Data Exports* section.

The single files generated in step 2.1 (Download a single Lane JSON file) are not affected by this setting, and the temperature data will always be included.

Field	R	N	Type	Description
monthNumber	R	–	`int`	The month number, where 0 = January and 11 = December
type	R	–	`enum<TemperaturePredictionType>`	The type of prediction
data	R	–	`TemperatureData[]`	

4.5.20. Temperature

The **`Temperature`** contains a timestamped Temperature. Typically, only the months January and July are included.

Field	R	N	Type	Description
timestamp	R	–	`int`	A timestamp in seconds
value	R	–	`float`	A temperature

4.5.21. VersionControl

The **`VersionControl`** entity captures any versioning information and workflow related data of the Lane, if it is enabled for a Lane.

Field	R	N	Type	Description
currentStatus	R	–	`enum<LaneStatus>`	The workflow status of the Lane
currentVersion	R	–	`int`	The current version
revisionPeriod	R	–	`int`	The period (in months) when the Lane expires
responsible	R	N	`UserReference`	The User assigned to this role
internalContributors	R	N	`UserReference[]`	The Users assigned to this role
internalApprovers	R	N	`UserReference[]`	The Users assigned to this role
externalParties	R	–	`ExternalParty[]`	The External Party the Lane is shared with
versions	R	–	`Version[]`	

4.5.22. ExternalParty

The **`ExternalParty`** entity captures users from outside your organization that play a role in the workflow.

Field	R	N	Type	Description
Company	R	–	`CompanyReference`	The Company of the External Party
approvers	R	–	`UserReference[]`	The Users assigned to this role

4.5.23. Version

The **`Version`** entity captures the current version information, such as the revision, when the document was approved, when it expires, and a list of actions that took place during the workflow.

Field	R	N	Type	Description
version	R	–	`int`	The Revision of this version
approvedOn	R	N	`date`	The date when the Lane was approved
expiresOn	R	N	`date`	The date when the Lane expires

actions	R	–	Action[]

4.5.24. Action

The **Action** entity captures information of a specific action in the workflow, for example a user approved the Lane.

Field	R	N	Type	Description
type	R	–	enum<HistoryType>	String Enumuration: historyType
date	R	–	datetime	The date and time of when this action occurred
user	R	N	UserReference	When null it was a automated action
comment	R	N	string	Any comments for the action taken

4.5.25. Phase

The **Phase** is a representation of a (part of) a process taking place at either a location (could be a distribution center, port, etc) or when in transit (e.g. when transporting by road, waiting at a border crossing is consider a stage in the whole road transport).

The sequence in which Phases are carried out is important, as they represent business logic, i.e. you can only go from 'loose cargo storage' to 'loose cargo loading' and not from 'loose cargo loading' to 'loose cargo loading' as that would be non-sensical.

Field	R	N	Type	Description
position	R	–	int	The position (sequence) of the Phase
type	R	–	enum<PhaseType>	The type of phase
duration	R	–	int	The duration of this phase
exposure	R	–	enum<ExposureType>	The type of ambient temperature exposure
packagingType	R	N	enum<PackagingType>	The type of packaging, if different from the Lane
insideBuilding	R	–	boolean	If the Phase takes place inside a building

4.5.26. Flight

The **Flight** entity encapsulates information of a flight, i.e. transport by air.

Field	R	N	Type	Description
flightCode	R	–	string	The Flight identification
departureLocal	R	–	datetime	The local departure time
arrivalLocal	R	–	datetime	The local arrival time
arrivalDayIndicator	R	–	int	The arrival indicator
operatingDays	R	–	days_of_operation	The days of operation
origin	R	–	Airport	The origin Airport from where the flight took off
destination	R	–	Airport	The destination Airport where the flight landed

4.5.27. Route

The **Route** entity encapsulates geographical routing information and can be used to visualize Lane information.

Field	R	N	Type	Description
from	R	–	coordinate	The 'from' Coordinate
to	R	–	coordinate	The 'to' Coordinate
estimatedDuration	R	–	int	Estimated duration
estimatedDistance	R	–	int	Estimated distance
encodedPolyline	R	N	encoded_polyline	Encoded polyline of the route

5. Change History

Date	Author	Content	Version
20-07-2020	Mark Bijl	Draft specification used for alpha/beta testing and demonstrations and obtaining customer feedback	0.1
24-06-2022	Mark Bijl / Marcel Tuinstra	Enhanced specification with additional information from the Lane configuration, harmonize with Location JSON	1.0
19-08-2022	Mark Bijl	Enhanced specification with the Air Step 'flights' property and the 'referenceType' of a CompanyReference, to be added in release 5.3.0	1.1
12-10-2022	Cristian Năvălici	Updating information related to the Prediction temp data	1.2
13-03-2024	Daumantas Patapas	Updating information related to the Customer locations/ Product categories/ Shipment date and temperature monitors	1.3