

Table of Contents

Version History.....	2
Overview	3
Upgrading to myPortal.....	3
API Key	4
URLs.....	4
UAT.....	4
Considerations for users already using the Customer Portal	4
Functions.....	5
HTTP Response codes	5
Consignments.....	6
Creating and Updating	6
Deleting	7
Requesting Labels	7
Requesting ETAs.....	7
Requesting Tracking Statuses	8
Requesting Image Data	9
Class Definitions.....	10
Consignment	10
Properties.....	10
Validation and Considerations	12
Example JSON.....	18
ConsignmentUploadResponse.....	20
Example of an un-successful consignment creation:.....	20
Example of a successful consignment creation:	21
List of potential warning messages.....	22
List of potential error messages.....	22
ConsignmentDeletionResponse.....	23
Success	23
Failure	23
LabelRequestResponse	23
Success	23
Failure	23
ETA Request Response.....	24

Success	24
Failure	26
Tracking Status Request Response	27
Success	27
Failure	31
Image Data Information Request Response	32
Success	32
Failure	34
GET Configuration	35
GET Consignment	35
POST Customer Paperwork	36
Example JSON	36
Properties	36
Failure Responses	36
API Key Request Form	38
Appendices	39
Customer Implant	39
Common Configuration Information	40

Version History

Date	Author	Comments
25/03/2021	Andrew Walchester	Initial draft documentation for consignment creation / updates / deletions and label creation
02/04/2021	Andrew Walchester	Updates based on initial feedback
13/04/2021	Andrew Walchester	Adding consignment response examples and typical warnings/errors
21/04/2021	Andrew Walchester	Added in "IsResidential" to the Address properties
21/04/2021	Andrew Walchester	Added in information about ADR information node
09/07/2021	Andrew Walchester	Modified the 'base URL' to include the API version
15/07/2021	Andrew Walchester	Added a bit of clarification around the format of the Base64 label string
04/03/2022	Aaron Parkes	Updated information on CubicMetres property
25/09/2021	Adrian Smith	Initial draft documentation for ETA, Tracking & Image Data
30/09/2021	Adrian Smith	Amendment to POD request to allow for request of specific image data
14/10/2022	Adrian Smith	Change of request parameters to "camelCase". Change of response when no data can be found Slight change to the response JSON Additional API Error responses included.

18/10/2022	Adrian Smith	Small change in the Status Code response to include “podname”
14/11/2022	Carlo Fiorletta	V1 & V2 API Functionality Documents Merged
08/02/2023	Carlo Fiorletta	Amendment made to request image data base64 encoded image string (uid amended to uld)
19/07/23	Carlo Fiorletta	Added embedded api key request form
11/03/25	Adrian Smith	Addition of 2 new Endpoints. Get Configuration & Delete Consignments
25/06/25	Adrian Smith	Addition of additional endpoints to be able to get Consignment Details and now post Customer Paperwork Images
01/07/25	Adrian Smith	Information for upgrading from Client Portal to myPortal

Overview

This document will detail how to use the Customer API to perform the following:

- Enter/Edit Consignments on the Vigo Customer Portal
- Request labels to affix to the products
- Request Estimated Time of Arrival Information
- Request Tracking Status Update Information
- Request Image Data (Proof of Delivery, Electronic Signatures & Delivery Photographs)

The API is written as a resource for developers performing integration on behalf of a ‘customer’ of a ‘depot’ and is not intended to be used by any third parties to integrate at ‘depot’ level and enter for multiple different accounts codes.

The REST API functions use JSON notation for their content and any consumers of the API must have a valid authentication key.

To use the API in a live environment you must first perform UAT with Vigo and your Depot, to ensure the data is as expected and that the GET requests are being successfully registered by the Vigo API.

Upgrading to myPortal

Please be aware that when you are upgrading the API connection from Vigo’s previous Client Portal to myPortal, there are some small differences between the two system that could affect a direct swap of the API URL.

Firstly, the field names will now be in Pascal Case. While this won’t affect posting data to the API. Some capitalisations on field names, especially in responses could be different.

Image formats that are returned when using GET Images endpoint could now be formats others than jpeg (for example .tiff, .pdf)

API Key

All calls to the API must have a valid API Key. An API key can be requested from Vigo Software ([form](#))

The API Key will be linked to your account on the customer portal and your permissions i.e. what functions you have available to you. As the API is expanded to provide additional functionality, it might be that there are multiple software providers within a business who require access to different functions and would therefore have an API key configured just for the relevant functions they require.

NB. The API Key will be different for use in the UAT environment and LIVE environment.

If you have any reason to enter orders as different Account Codes you will need more than one API Key.

There may be cause to re-issue API Keys, so integrators are advised to ensure they can change this key in their software easily in case of a need to re-issue.

URLs

The 'live' root URL for the API is

```
(LiveURL) [to be defined]
```

The 'UAT' root URL for the API is

```
https://api-uat.vigocloud.com/api/v1
```

UAT

Before a 'live' key is issued to an integrator UAT must be performed and signed off by all parties i.e. Vigo, the Depot and Customer.

The exact items that make up UAT may differ depending on customer but typically will ensure that all expected types of consignments (combinations of sizes of product and service/surcharge/time combinations) that you expect to ship can be entered.

Considerations for users already using the Customer Portal

When entering consignments via the API they ultimately end up as consignments on the Vigo Customer Portal in the same way as if they had been entered manually or uploaded via the [customer implant](#), with this in mind the API will use a lot of the options setup against the customer on the portal when importing consignments. This mostly relates to the validation performed by the API e.g. what services and surcharges are allowed, what unit types your user is allowed to use etc.

The customer portal also has a lot of 'defaults' which will pre-populate sections of an order based on their most common order scenario e.g. "Next Day" service and "Pre 12:00 delivery" surcharge. These defaults work well when paired with a user-interface because they can be changed by the end user if needed, they are after all just defaults. When using an API where Vigo are not in control of the user interface these defaults don't work very well. Take for example surcharges. If there are no surcharges passed as part of the JSON, should the system assume this means "no surcharges are required" or does it mean "use the defaults". With this in mind the API will not support most of the 'defaults'. Where any default data will be used by the API if they are not populated, this will be noted in this document.

Functions

The following sections detail the functions available in the URL.

Where the URL to a function contain values in curly brackets {} this implies a value is used in its place. A consistent example across all of the URLs is the addition of the query parameter of {apiKey} which should be the API key you have been given for the environment (i.e. live/UAT) that you are trying to access.

As part of the data request, ETAs, Tracking Statuses & Images can be requested for 1 individual consignment, or all consignments with events between a date range.

Where data is requested via a “from” & “to” Timestamp, this is the created data time and not the data capture time. Time requested should be in UTC.

It is the responsibility of the consumer to record how many items there are per request.

Requests will be limited using {timestamp} to 1 request per API key per hour.

HTTP Response codes

Requests to the API use standard HTTP status codes to determine success or failure.

Code	Description
200	If a request is successful it will return a HTTP status code of
400	Invalid data. This could many things but typically means you are authorised to make this request, but it has not passed of validation checks. The specific problem will be part of the json content returned
401	If your API key is invalid or not for the URL you are trying to access i.e. live / UAT
403	If your API key is valid, but you either do not permissions to that function or the resource you are trying to access via that function e.g. the consignment you have requested is not for the customer your API key refers to
404	If you request an consignment update, label or any other function specifically for a single consignment and that consignment cannot be found. <i>No content will be returned.</i>
5XX	An error has occurred on the server.

All responses will contain an array of Warnings and Errors in the JSON content

Errors will only be returned on any requests that have not been successful i.e. those returning a HTTP status in the 400 range.

Warnings can even occur on successful responses.

Consignments

Consumers of the API can Create, Update and Delete orders. When Creating and Updating orders you must be aware that the expected values and validation performed by any function could differ depending on how the customer has been setup by their depot (and could change over time). This is an important factor if you are a software provider and are integrating with the Vigo API on behalf of more than one customer/depot.

Some examples of the differences are what unit types, services, surcharges are available and what lines of the manifest notes are available for population by the users.

It is important you understand how your customer has been setup when using this API. To aid in this we provided a form in this appendix with [common questions you will need to know the answers to before consuming the API](#).

Creating and Updating

Create

Users can create a consignment via the following URL using the POST method:

```
/consignments?code={apiKey}
```

The body of the request should be a single [Consignment](#) and the response will be in the format of [ConsignmentUploadResponse](#). An important part of the response to a successful import is the 'TrackingCode'. This is a unique reference used to identify the job and in most cases is also part of the barcode on the label. It is an important element of other requests such as updating consignment and requesting labels, so it is important this is logged in the system consuming the API.

A HTTP status code of 200 indicates a new consignment was successfully entered.

The response will include various items including the TrackingCode allocated to the new consignment. The TrackingCode is a unique reference to the consignment that will be generated automatically. It is important that if you want to update consignments or request labels (and future information like PODs, Tracking, ETA's etc) that this is recorded because it forms part of the requests for that information.

Please read the [Consignment](#), [ConsignmentUploadResponse](#) and [Validation of Creation/Updating sections](#) for information about the content of the request.

Please note that depots have the option to manage 'cut off' in the system, which are used to define when consignments must be entered by to ensure collection on the requested date. There is potential for this to reject the consignment or automatically carry over the consignment to the new next day. See '[Cut Offs](#)' for more information.

Updating

Updating of an order can be performed using the following URL using the PUT method:

```
/consignments/{trackingcode}?code={apiKey}
```

The body and the response of this request is identical to that of the 'Create' function i.e. the request should be a single [Consignment](#) and the response will be in the format of [ConsignmentUploadResponse](#). The key difference is that request must include a valid TrackingCode for the customer that your API Key relates to. This TrackingCode would be the result of a previous successful consignment creation.

Deleting

Deleting of an order can be performed using the following URL using the DELETE method:

```
/consignments/{trackingcode}?code={apiKey}
```

This request does not require a body

Please check the HTTP status code to determine if the request was successful.

The response body will be as per [ConsignmentDeletionResponse](#)

Requesting Labels

Labels for each item on the consignment can be produced by making a call to the following URL using the GET method.

```
/consignments/{trackingcode}/labels?code={apiKey}
```

Please check the HTTP status code to determine if the request was successful.

The response will be as per [LabelRequestResponse](#)

The label document will be returned as a PDF that has been base 64 encoded. The consumer of the API should decode the content to generate the PDF document.

Requesting ETAs

ETA data for each consignment can be produced by making a call to the following URL using the GET method. The consumer can request ETAs using only one of the following references. This GET request will return the latest Estimated Time of Arrival for the consignment in question ONLY.

```
/consignments/eta?code={ApiKey}&trackingCode={trackingCode}
/consignments/eta?code={ApiKey}&consignmentNumber={consignmentNumber}
/consignments/eta?code={ApiKey}&customerReference1={customerReference1}
/consignments/eta?code={ApiKey}&customerReference2={customerReference2}
```

ETA Data can also be requested using the same reference values but with an additional criteria of “includeall”. When including this in the GET request to the API, all ETAs available on a given consignment will be returned. This will include previously outdated ETAs which have been superseded by a more recently calculated Time of Arrival.

```
/consignments/eta?code={ApiKey}&trackingCode={trackingCode}&includeall=true
/consignments/eta?code={ApiKey}&consignmentNumber={consignmentNumber}&includeAll=true
/consignments/eta?code={ApiKey}&customerReference1={customerReference1}&includeAll=true
/consignments/eta?code={ApiKey}&customerReference2={customerReference2}&includeAll=true
```

ETA data for consignments between a date range can be produced by making a call to the following URL using the GET method. This GET request will return only the latest Estimated Time of Arrival for all consignments with events created between the date and time range requested.

```
/consignments/eta?code={apiKey}&from={fromTimestamp}&to={toTimestamp}
```

Time stamp should be in the format yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (2021-07-01T14:59:55.711Z) using UTC.

The response will be as per [Requesting ETAs](#)

Requesting Tracking Statuses

Tracking data for each consignment can be produced by making a call to the following URL using the GET method. The consumer can request Tracking Statuses using only one of the following references. This GET request will return all tracking data for the consignment number requested.

```
/consignments/status?code={ApiKey}&trackingCode={trackingCode}

/consignments/status?code={ApiKey}&consignmentNumber={consignmentNumber}

/consignments/status?code={ApiKey}&customerReference1={customerReference1}

/consignments/status?code={ApiKey}&customerReference2={customerReference2}
```

Tracking data for consignments between a date range can be produced by making a call to the following URL using the GET method. This GET request will only return tracking data for events created between the date and time range requested.

```
/consignments/status?code={apiKey}&from={fromTimestamp}&to={toTimestamp}
```

Time stamp should be in the format yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (2021-07-01T14:59:55.711Z) using UTC.

The response will be as per [Tracking Status Request Response](#)

Requesting Image Data

Image data for each consignment can be produced by making 2 separate API call to the following URLs using the GET method. The reason for this is that the images returned by the API will be base64 encoded images. If a response contains multiple base64 encoded images, the length of the response may be too long to be practical. Firstly, an “Image ID” can be requested from the API and then this ID will be required to gain the base64 encoded image.

Images returned using the Image Data Request will be a combination of Electronic Signature Data, Scanned Proof of Delivery Paperwork and Photographs captured at the Delivery Location.

The consumer can request Image IDs using only one of the following references.

```
/consignments/images?code={ApiKey}&trackingCode={trackingCode}

/consignments/images?code={ApiKey}&consignmentNumber={consignmentNumber}

/consignments/images?code={ApiKey}&customerReference1={customerReference1}

/consignments/images?code={ApiKey}&customerReference2={customerReference2}
```

As part of the Image ID request, we will allow the consumer to request to only receive a specific type of images, be this POD data (including Electronic Signature Data) or Delivery Photograph data. The consumer will have the ability to set a parameter to include the data they wish to receive. The following calls can be made to receive only specific data.

```
/consignments/images?code={ApiKey}&trackingCode={trackingCode}&includePod=true

/consignments/images?code={ApiKey}&trackingCode={trackingCode}&includePhoto=true
```

Should the consumer wish to receive both image types, the following call can be made.

```
/consignments/images?code={ApiKey}&trackingCode={trackingCode}&includePod=true&includePhoto=true
```

Image ID data for consignments between a date range can be produced by making a call to the following URL using the GET method. This GET request will only return image data for events created between the date and time range requested.

```
/consignments/images?code={apiKey}&from={fromTimestamp}&to={toTimestamp}
```

Time stamp should be in the format yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (2021-07-01T14:59:55.711Z) using UTC.

Once the Image ID has been returned successfully, then the following call can be made to receive the base64 encoded image string

```
/consignments/images?code={apiKey}&imageType={imageType}&uId={UID}
```

The request to receive the base64 encoded image string will require both the image type and the uid value.

Both parameters are required and the image cannot be requested by using just an ID or an ImageType.

The responses will be as per [Image Data Information Request Response](#)

Class Definitions

Consignment

Properties

Property	Type	Comments
References		
ConsignmentNumber	String (7)	Please see Validation – Consignment Number . Depending on how the customer has been setup by their depot, this could be mandatory, auto-generated or auto-prefixed and it’s length could differ depending on the preference of the depot. This is typically the reference that a customer will see on the consignments documentation such as labels and invoices
CustomerReference	String (16)	Optional additional reference to the order
CustomerReference2	String (16)	Optional additional reference to the order
TrackingCode	String	Will be removed from the spec. Only relevant on updates, but it’s in the URL so not needed in here.
ConsignmentType	Integer	1 = Collection, 2 = Delivery, 3 = Collection And Delivery See Validation - Job Types
Collection		
Address (NB. The rules noted here are affected by the ConsignmentType)		
Name	String (30)	Mandatory when <i>References</i> . ConsignmentType is 1 (Col) or 3 (Col and Del)
Address1	String (30)	Mandatory when <i>References</i> . ConsignmentType is 1 (Col) or 3 (Col and Del)
Address2	String (30)	
Address3	String (30)	Town/City. Mandatory when <i>References</i> . ConsignmentType is 1 (Col) or 3 (Col and Del)
Address4	String (30)	County. Mandatory when <i>References</i> . ConsignmentType is 1 (Col) or 3 (Col and Del)
Postcode	String (10)	Mandatory when <i>References</i> . ConsignmentType is 1 (Col) or 3 (Col and Del) Outbound section must be valid
Country	String (3)	ISO 3 digit country code
IsResidential	Boolean	True/false depending on the type of property. Please note some depots may default additional surcharges to the consignment in this situation. If additional surcharges are added, a ‘warning’ message will be returned.
Date	String (10)	Should be no earlier than today. Format of yyyy-mm-dd
Time	String (5)	Format of HH:MM – only valid on ConsignmentType of 1 or 3
Contacts		
Name	String (30)	See Validation and Considerations - Contacts . Only appropriate on job types 1 and 3
Phone	String (16)	
SMS	String (25)	
Email	String (50)	
Notes		
Line1	String (48)	See Validation and Considerations - Notes
Line2	String (48)	
Line3	String (48)	
Line4	String (48)	
Delivery		
Address (NB. The rules noted here are affected by the ConsignmentType)		
Name	String (30)	Mandatory when ConsignmentType is 2 (Del) or 3 (Col and Del)
Address1	String (30)	Mandatory when ConsignmentType is 2 (Del) or 3 (Col and Del)
Address2	String (30)	
Address3	String (30)	Town/City. Mandatory when ConsignmentType is 2 (Del) or 3 (Col and Del)
Address4	String (30)	County. Mandatory when ConsignmentType is 2 (Del) or 3 (Col and Del)
Postcode		Mandatory when ConsignmentType is 2 (Del) or 3 (Col and Del) Outbound section must be valid
Country	String (3)	ISO 3 digit country code
IsResidential	Boolean	True/false depending on the type of property. Please note some depots may default additional surcharges to the consignment in this situation. If additional surcharges are added, a ‘warning’ message will be returned.
Date	String (10)	be no earlier than today. Format of yyyy-mm-dd
Time	String (5)	Format of HH:MM – only valid on ConsignmentType of 1 or 3
Contacts		
Name	String (30)	See Validation and Considerations - Contacts . Only appropriate on job types 2 and 3
Phone	String (16)	
SMS	String (25)	
Email	String (50)	
CustomerPaperworkRequired	Boolean	See Validation and Considerations – Customer Paperwork
Notes		

	Line1	String (48)	See Validation and Considerations - Notes
	Line2	String (48)	
	Line3	String (48)	
	Line4	String (48)	
GoodsDefinition			
	TotalSpaces	Integer	Total whole pallet spaces the consignment takes up. <i>This may be over-ridden depending on a customer level setting. If enabled, it will be set to the sum of the number of items on the consignment.</i>
	TotalKilos	Integer	Total whole kilos for the consignment
	UnitCode	String (2)	Mandatory – Must be a valid unit code for your customer
	CubicMetres	“Decimal”	Optional – cubic metres for the consignment. This must be written in quotation.
	Items	Array	See Item Types
	Item Type	String (10)	A valid Item Type for the customer
	Quantity	Integer	A positive integer value
	OversizeCharging		See Oversize Charging Information
	Items	Array	
	Item Type	String (10)	A valid Item Type for the customer
	Quantity	Integer	A positive integer value
	TotalSpaces	Integer	A positive integer, which is the sum of the quantities entered for each item in the items array
ServiceDefinition – See Services and Surcharges for more information			
	Service	String	A valid service code for this customer
	Surcharges	Array of string	A list of valid surcharges available for this customer, note that some surcharges are incompatible with other surcharges and the usage of certain surcharges may make other fields like time or contact information mandatory. <i>NB. All surcharges required must be specified, none will be defaulted based on customer configuration on the customer portal.</i>
Amazon			
	AdvancedShippingNotification		Future Development.
	FulfilmentByAmazon		Currently this information should be entered into the delivery notes
	BillOfLading		
	CartonCount		
	UnitCount		
	PurchaseOrderNumber		
	HazardousInformation	Array	See ADR Information
	ItemNo	Integer	Integer value for the item no this ADR information applies to. NB. Not all customers are required to enter this and the ADR is linked only to the consignment, not an individual item
	UNNo	Integer	A valid UN No for the ADR product
	NoOfPackages	Integer	Total number of packages this product is in
	PackageType	String	A valid product type
	PackageMeasure	String	A valid package measure
	QtyPerPackage	Integer	A positive integer for how much is in each package
	PackingGroup	String	Where appropriate a valid Packing Group for this UN No
	ProductDescription	String	Description/Product name of the ADR Product

Validation and Considerations

Consignment Number

The length of the consignment number will depend on the settings used by the depot you are a customer of.

It could be prefixed with one or more characters.

Depending on the options that have been set against your customer on the customer portal this field may be; Auto-calculated or available for entry

Auto Calculation

When Auto-calculation is enabled the consignment number will be generated by the system. This auto-generated value is a string made up of an incremental number that has been padded with leading zeroes to fill the max length of the consignment number. It may or not be prefixed with additional characters.

If a consignment number is included in the JSON when auto-generation is enabled, the system will ignore it.

The response to a successful consignment insert will return the consignment number that was generated as part of the response.

Supplying a Consignment Number

If auto-generation is not enabled, then you must supply a consignment number. The value can be any combination of characters between A-Z and 0-9. It must be less than or equal to the maximum allowed consignment number length NB. When a prefix is configured the length of consignment number must factor in these additional characters.

If the value supplied is less than that allowed, it will be padded with leading zeroes and if appropriate have any prefix characters added.

The response to a successful consignment insert will return the consignment number that was generated.

Job Types

A consignment can be one of the following:

Value	Type	Description
1	Collection	A consignment you would like to be collected and brought back to one of your site addresses. You must supply a collection address for this job type. Delivery address is optional. If not populated it will use your default site address. If populated it should match one of your site addresses on the portal.
2	Delivery	A consignment you are despatching from one of your site addresses to be delivered elsewhere. If not populated, collection address will populate with your default site address. If populated it should match one the addresses setup in the customer portal.
3	Collection and Delivery	A consignment that is neither collected from one of your own site addresses or to be delivered to one of your site addresses. Both the collection and delivery address should be populated.

You may not have access to every job type. Please check if all job types are available to the customer.

Contacts

All four of the properties within the Contacts property can be enabled/disabled by the depot on a per customer basis depending on the options offered, so not all contact lines are needed. If you provide values that are not required, you will receive a 'warning' message in the response to the function call.

Property	Type	Description
Name	String (30)	A single name
Phone	String (16)	A single phone number
SMS	String (25)	A single mobile phone number, which if enabled can be used for notifications if this is offered by the depot. MUST ONLY BE A SINGLE NUMBER.

Email	String (50)	A single e-mail address If populated it should be a valid format i.e. contain an @ and at least a single . after the @
-------	-------------	--

Notes

There are up to 4 lines of notes available for both collection notes and delivery notes.

Please note, the amount of lines available for entry by the customer is configurable by depot. This is because some depots like to reserve certain lines of notes for their own purposes e.g. drivers notes or loaders notes.

With this in mind, depending on how the customer has been setup you may get errors/warnings if you populate lines you are not allowed to access.

For *Delivery.Notes*, there is an additional consideration which is that some depots like the notes to contain the descriptions of any surcharges e.g. "Timed Delivery, Tail Lift Required" and/or any dates or times. If this is enabled it can further limit the amount of notes available for input via the API.

In the case that any additional notes have been added onto the order based on the surcharges or date/times they will prefix any notes specified in the API. This may cause some notes to move from property to the next i.e. line1 to line2 and so on. Typically, if a depot wants to enable the auto-population of notes, they will also disable manual entry into Line1.

Customer Paperwork

Depending on the kind of orders being sent and the service being offered by the depot, a customer may be able to use the customer portal to specify that they require delivery paperwork to be included with their deliveries. Depending on the method used to deliver the orders this could be physical paperwork that is sent with the product or could be electronic scans of the paperwork that are sent electronically to the party performing the delivery.

Customer paperwork options can be enabled or disabled. If it is disabled and this option is populated with true, the order will enter, but a 'warning' will be returned to say that this feature is not available

If Customer paperwork is enabled it can either be set by the depot to be:

- Always true
- Always false
- Available for the user to choose.

If the depot is forcing a value of either true/false, any value entered here will be ignored and a warning will be included in the response.

Item Types

The items array is an array of objects with the following properties:

- ItemType
- Quantity

The types of item available to a customer will differ depending on the type of product they transact and the commercial agreement they have with their depot.

One item should be added to the array for each distinct Item Type and the quantity should be total number of units of that type. The available item types for your customer will be available by talking to the Depot and Vigo.

So for example if there were unit types X,Y, Z available and you had one type X, 2 type Y and 3 type Z you would generate JSON like this:

```
"Items": [
  {
    "ItemType": "X",
    "Quantity": 1
  },
  {
    "ItemType": "Y",
    "Quantity": 2
  },
  {
    "ItemType": "Z",
    "Quantity": 3
  }
],
```

Quantity is a positive integer value

****Please be aware that item type descriptions could differ. Please use the GET Configuration endpoint to receive the correct data to use****

Oversize Charging Types

This is a specific property that can be enabled at customer level for over-sized pallet products to define what equivalent 'normal' item types it can be compared to from a pricing point of view.

Some depots using the Vigo system charge simply a certain amount per oversize item, but depending on the depots configuration they might charge based on the equivalent number of other units that it takes up the rules for this can be complex, so it is best to talk to the depot directly to understand this.

Let's imagine we have a consignment made up of two over-size items, but one is equivalent to two units of type X and the other is equivalent to three of type Y, we would enter JSON for this as:

```
"OversizeCharging": {
  "Items": [
    {
      "ItemType": "X",
      "Quantity": 2
    },
    {
      "ItemType": "Y",
      "Quantity": 3
    }
  ],
  "TotalSpaces": 5
}
```

Total Spaces should be the sum of the total quantity property for all items in the array

Services and Surcharges

Services

Typically, the service relates to the day that the goods are required to be delivered by. E.g. Next Working Day, 2 Working Days, or a specific day.

There are options stored on a service to denote if the delivery date needs to be populated or if it will be calculated based on service. The date is only *needed* if the service is a specific day service, otherwise it will be calculated. If the data is populated but the service does not require it, it will be ignored. A warning will be returned in the JSON response.

Some services may not be available for all destinations. This can be configured by the depot and is set against the postcode. If you request a service that is not applicable for the postcode

Surcharges

Surcharges can either relate to an additional time constraint required when delivering the consignment or they could simply be an additional instruction or flag on the job.

Examples of timed surcharges could be:

- required before 12:00
- required before a specific time

Timed surcharges can be configured to allow times to be entered rather than a default of e.g. before 12:00 and these can also be set to have min and max ranges e.g. any time between 10:00 and 12:00

Where there are any constraints setup like this, the upload of consignments will validate to ensure the delivery time is within the constraints. If it isn't then the consignment entry will fail and an error will be returned.

You must only specify one timed surcharge or the consignment will fail. For instance, if a consignment is required at 11:00 do not enter both a specific time and a pre-12:00 surcharge.

Examples of additional non-timed surcharges could be:

- Tail Lift Required
- Booking In Required
- FORs
- It's destined for an AMAZON warehouse
- Pre-booked i.e. required on a specific delivery date

The exact list of surcharges will depend on the configuration of the customer

Combinations

Note that not all services allow the same list of surcharges. They are often very similar, but there is potential for there to be differences. Please liaise with the depot of the customer to understand this fully.

If there is an invalid combination of services and surcharges and creation or updating of an order will fail with an error.

ADR Information

This information is only needed if defined by the depot.

Property	Type	Comments
ItemNo	Integer	Integer value for the item no this ADR information applies to. NB. Not all customers are required to enter this and the ADR is linked only to the consignment, not an individual item
UNNo	Integer	A valid UN No for the ADR product
NoOfPackages	Integer	Total number of packages this product is in
PackageType	String	A valid package type. This list can be configured per depot, so please liaise with your depot for valid values. Typical values are: BAG, BOX, CAN, CAR, CYL for (Bags, Boxes, Cans, Cartons and Cylinders respectively) but please confirm with your depot
PackageMeasure	String	This list can be configured per depot, so please liaise with your depot to get the valid package measures. Typical values are: KGS or LTR
QtyPerPackage	Integer	A positive integer for how much is in each package
PackingGroup	String	Where appropriate a valid Packing Group for this UN No. Some UN No's are only allowed to be defined as specific packing groups, so it must be appropriate for the UN No. Packing groups are entered as I, II or III
ProductDescription	String	Description/Product name of the ADR Product

Cut-Offs

For each type of job i.e. 1,2,3 for Collection, Delivery and Collection & Delivery respectively, the depot can optionally create a cut-off time after which consignments cannot be entered or amended.

If there is a cut-off configured and a consignment is entered after this time, it will automatically have it's collection date and delivery date rolled onto the next day. The delivery date will not be altered if the service is a 'specific day' service.

A Warning with a 'type' of CUTOFF will generated if this occurs.

Example JSON

```
{
  "References": {
    "ConsignmentNumber": "ABC1234",
    "CustomerReference": "CUSTREF1",
    "CustomerReference2": "CUSTREF2",
    "TrackingCode": "1655551234567",
    "ConsignmentType": "2"
  },
  "Collection": {
    "Address": {
      "Name": "Collection Name",
      "Address1": "Collection Property",
      "Address2": "Collection Road",
      "Address3": "Collection Town",
      "Address4": "Collection County",
      "Postcode": "B97 6QL",
      "Country": "GBR",
      "IsResidential": false
    },
    "Date": "2021-04-07",
    "Time": null,
    "Contacts": [
      {
        "Name": "Mr Smith ",
        "Phone": "01527 123 456",
        "SMS": "07711 123456",
        "Email": "mrsmith@testemail.com"
      }
    ]
  },
  "Delivery": {
    "Address": {
      "Name": "Delivery Name",
      "Address1": "Delivery Property",
      "Address2": "Delivery Road",
      "Address3": "Delivery Town",
      "Address4": "Delivery County",
      "Postcode": "B6 6QE",
      "Country": "GBR",
      "IsResidential": true
    },
    "Date": "2021-04-08",
    "Time": "10:00",
    "CustomerPaperworkRequired": true,
    "Contacts": [
      {
        "Name": "Mr Jones",
        "Phone": "123456789",
        "SMS": "07890 1235",
        "Email": "mrjones@testcompany.com"
      }
    ],
    "Notes": {
      "Line1": "Please deliver on small vehicle",
      "Line2": "get paperwork signed",
      "Line3": null,
      "Line4": ""
    }
  }
},
```

```

"GoodsDefinition": {
  "TotalSpaces": 5,
  "TotalKilos": 2500,
  "UnitCode": "K",
  "CubicMetres": 6,
  "Items": [
    {
      "ItemType": "FULL",
      "Quantity": 2
    },
    {
      "ItemType": "OVERSIZE",
      "Quantity": 2
    }
  ],
  "OversizeCharging": {
    "Items": [
      {
        "ItemType": "FULL",
        "Quantity": 3
      }
    ]
  },
  "TotalSpaces": 3
},
"ServiceDefinition": {
  "Service": "ND",
  "Surcharges": [
    "AM",
    "LF"
  ]
},
"Amazon": {
  "AdvancedShippingNotification": "ASN NUMBER",
  "FulfilmentByAmazon": "FBA REFERENCE",
  "BillOfLading": "BOL 1234",
  "CartonCount": "1 ",
  "UnitCount": "5",
  "PurchaseOrderNumber": "PO No 1234"
},
"HazardousInformation": [
  {
    "ItemNo": null,
    "UNNo": "2010",
    "NoOfPackages": 10,
    "PackageType": "BAG",
    "PackageMeasure": "KILOS",
    "PackingGroup": "II",
    "ProductDescription": "Supergas Garden Gas"
  }
]
}

```

ConsignmentUploadResponse

Property	Type	Comments
Success	Boolean	True/False
TrackingCode	String (13)	The tracking code of the consignment as generated by the system. Capture this for use on subsequent requests like updates or label requests. NB. On any updates this will return the original tracking code, it will not change
ConsignmentNumber	String (16)	The consignment number either auto-generated by the system or as uploaded (with any leading zeroes or prefixes added)
Errors	Array	
Type	String	
Description	String	
Warnings	Array	
Type	String	
Description	String	

A successful order will never have any errors, but it could have warnings.

A unsuccessful order will have errors provided. It might also have warnings too.

The arrays of Errors and Warnings are the same structure i.e. they have a Type and a Description. The type could be used by the consuming application to provide context to the end user or could be used as the basis of a filter to only show warnings that are relevant. E.g. You may not worry if the notes have truncated, but do worry if the job has been entered after the cut-off and as such as rolled onto another day and will not be delivered when it is required.

Example of an un-successful consignment creation:

```
{
  "Success": false,
  "TrackingCode": "",
  "ConsignmentNumber": "",
  "Errors": [
    {
      "Type": "SERVICE",
      "Description": "Service ND is not valid for the specified postcode (AB10)"
    },
    {
      "Type": "SURCHARGES",
      "Description": "Surcharge PB is not available for Service ND"
    }
  ],
  "Warnings": [
    {
      "Type": "CUTOFF",
      "Description": "The cut-off (10:00) has passed. The consignment has been moved to the following working day (12/04/2021)"
    },
    {
      "Type": "NOTES",
      "Description": "Some of the notes entered have been truncated"
    }
  ]
}
```

Example of a successful consignment creation:

```
{
  "Success": true,
  "TrackingCode": "0015551234567",
  "ConsignmentNumber": "A123456",
  "Warnings": [
    {
      "Type": "CUTOFF",
      "Description": "The cut-off (10:00) has passed. The consignment has been moved to the following working day (12/04/2021)"
    },
    {
      "Type": "NOTES",
      "Description": "Some of the notes entered have been truncated"
    }
  ]
}
```

List of potential warning messages

Type	Description	Comments
CUTOFF	The cut-off of (xx:xx) has passed. The consignment has been moved to the following working day (dd/mm/yyyy)	
NOTES	Some of the notes entered have been truncated	
CONTACTS	Delivery Contact Name is not required and has been ignored	A customer can be configured to only have certain contact fields required. Name, Phone, SMS and Email are all available to be enabled/disabled Typically, Phone and Name are enabled, but SMS and Email will depend on services offered by the depot
CONTACTS	Delivery Contact Phone is not required and has been ignored	
CONTACTS	Delivery Contact SMS is not required and has been ignored	
CONTACTS	Delivery Contact E-Mail is not required and has been ignored	
SERVICE	The service specified does not require a delivery date to be specified, it is calculated from the collection date and service chosen	When the API has specified a delivery date on the order, but it is not <i>required</i> for this specific service.
TIME	You have entered a delivery time, but not selected a timed surcharge	
SURCHARGES	The surcharge(s) of "HD" have been added to this order because the address is a Residential Address.	

NB. This list is not exhaustive, there will be other warnings/errors not defined here.

List of potential error messages

Type	Description	Comments
NOTES	You have populated line X but this line is not available for your user	Where line X has been reserved for depot use.
SURCHARGE	Invalid surcharge (YY) for the service code XX	
SURCHARGE	Invalid surcharge (xx) for postcode (XXXX)	This surcharge is not available for this postcode. Particularly used for timed deliveries to remote parts of the UK or abroad
SERVICE	Service (XX) is not allowed for this postcode (YYYY)	Similar to the above, but related to the service
SURCHARGE	Invalid combination of surcharge	Some surcharges are not allowed in conjunction with other surcharges. You can have more than one timed surcharge
SURCHARGE	Time is required for surcharge (XX)	
SURCHARGE	Specific date is required for surcharge	
CONTACTS	Email is required for surcharge (XX)	
CONTACTS	Phone is required for surcharge (XX)	
CONTACTS	Name is required for surcharge (XX)	
CONTACTS	SMS is required for the surcharge (XX)	
REFERENCE	Consignment number is required	
TRACKINGCODE	Tracking code cannot be found	Cannot locate this consignment via tracking code when attempting to perform an update
REFERENCE	Consignment number is too long. Max length is X. NB. Your customer has a prefix of Y.	

NB. This list is not exhaustive, there will be other warnings/errors not defined here.

ConsignmentDeletionResponse

Success

```
{
  "Success": true,
  "TrackingCode": "",
  "ConsignmentNumber": "",
  "Errors": [],
  "Warnings": []
}
```

Failure

```
{
  "Success": false,
  "TrackingCode": "123551234567",
  "ConsignmentNumber": "A123456",
  "Errors": [
    {
      "Type": "CUTOFF",
      "Description": "You cannot delete a consignment that is for a date other than today"
    }
  ],
  "Warnings": []
}
```

If the consignment cannot be found via the Tracking Code specified a HTTP status of 404 will be returned.

LabelRequestResponse

The label response will return a similar response to the other functions, but with an additional “Document” property which will contain a base 64 encoded PDF document. The API consumer should de-serialise this content to make a document suitable for consuming in their application.

The Document content will have been URL encoded, so before processing will need to be URL Decoded.

Success

```
{
  "Success": true,
  "TrackingCode": "123551234567",
  "ConsignmentNumber": "A123456",
  "Errors": [],
  "Warnings": [],
  "Document": "TWFuIGl3IGRpc3Rpbmd1aXNoZWQsIG5vdCBvbm....."
}
```

Failure

The json content for a failure where there is no label setup will look like this. The HTTP status will be 400

```
{
  "Success": false,
  "TrackingCode": "123551234567",
  "ConsignmentNumber": "A123456",
  "Errors": [
    {
      "Type": "DESIGN",
      "Description": "There is no label design configured"
    }
  ],
  "Warnings": [],
  "Document": null
}
```

If the consignment cannot be found via the Tracking Code specified a HTTP status of 404 will be returned.

ETA Request Response

The ETA Request response will return “Date & Time Window” property which will contain the latest ETA window available for the trackingCode requested or all ETAs events between a date range.

The ETA window will be dependent on the configuration within the Client Portal. If no ETA window is setup, both DateTime fields will be populated with the same information, which will be the exact Estimate Time of Arrival.

The Start & End Window provided is in the local time of the location of the delivery.

Success

```
{
  "success": true,
  "payload": {
    "consignmentReferences": {
      "trackingCode": "1315550657146",
      "consignmentNumber": "KP57146",
      "customerReference1": "W-2097869",
      "customerReference2": ""
    },
    "etas": [
      {
        "startWindow": "2022-09-29 12:00:00",
        "endWindow": "2022-09-29 14:00:00",
        "uploaded": "2022-09-29 12:33:20"
      }
    ]
  }
}
```

Where the API has been requested to return data between a given date range, the successful response will include all consignment ETA data found since the requested date & time. ***Please note, the data returned will provide all ETA data generated between the date & time ranges provided, not just ETA data that falls between the two provided ranges.***

```
{
  "success": true,
  "payload": {
    "results": [
      {
        "consignmentReferences": {
          "trackingCode": "1315550660178",
          "consignmentNumber": "KP60178",
          "customerReference1": "W-2108145",
          "customerReference2": ""
        },
        "etas": [
          {
            "startWindow": "2022-10-05 09:50:00",
            "endWindow": "2022-10-05 11:50:00",
            "uploaded": "2022-10-05 09:57:23"
          }
        ]
      },
      {
        "consignmentReferences": {
          "trackingCode": "1315550660009",
          "consignmentNumber": "KP60009",
          "customerReference1": "W-2106315",

```

```

        "customerReference2": ""
    },
    "etas": [
        {
            "startWindow": "2022-10-05 09:40:00",
            "endWindow": "2022-10-05 11:40:00",
            "uploaded": "2022-10-05 09:57:21"
        }
    ]
},
{
    "consignmentReferences": {
        "trackingCode": "1315550659868",
        "consignmentNumber": "KP59868",
        "customerReference1": "W-2106359",
        "customerReference2": ""
    },
    "etas": [
        {
            "startWindow": "2022-10-05 11:10:00",
            "endWindow": "2022-10-05 13:10:00",
            "uploaded": "2022-10-05 09:57:23"
        }
    ]
}
]
}
}
}

```

Failure

Should no ETA be found for the requested consignment, using the API request via specific trackingCode, Consignment Number, or Customer References a response will be “204 No Content” with an empty response.

If the consignment cannot be found via the Tracking Code, Consignment Number, Customer Reference 1 or Customer Reference 2 specified a HTTP status of 404 will be returned.

```
{
  "success": false,
  "payload": {
    "errors": [
      "No Consignments Found"
    ]
  }
}
```

Should the request parameters return multiple consignments, the following HTTP status of 404 will be returned

```
{
  "success": false,
  "payload": {
    "errors": [
      "Multiple Consignments Found"
    ]
  }
}
```

Tracking Status Request Response

The Tracking Statuses Request response will contain the latest Tracking data available for the trackingCode requested or all Tracking Statuses since the Date & Time provided.

Success

```
{
  "success": true,
  "payload": {
    "consignmentReferences": {
      "trackingCode": "1315550660207",
      "consignmentNumber": "KP60207",
      "customerReference1": "W-2108249",
      "customerReference2": ""
    },
    "statuses": [
      {
        "code": "DSO",
        "captured": "2022-10-04 19:25:15",
        "uploaded": "2022-10-04 18:35:10",
        "description": "Depot Scanned OUT",
        "itemNumber": "1",
        "podName": ""
      },
      {
        "code": "DSO",
        "captured": "2022-10-04 19:25:15",
        "uploaded": "2022-10-04 18:41:03",
        "description": "Depot Scanned OUT",
        "itemNumber": "1",
        "podName": ""
      },
      {
        "code": "SCO",
        "captured": "2022-10-05 03:31:54",
        "uploaded": "2022-10-05 04:21:30",
        "description": "Hub Scanned OUT",
        "itemNumber": "1",
        "podName": ""
      },
      {
        "code": "DSI",
        "captured": "2022-10-05 06:52:17",
        "uploaded": "2022-10-05 07:17:38",
        "description": "Depot Scanned IN",
        "itemNumber": "1",
        "podName": ""
      },
      {
        "code": "DSV",
        "captured": "2022-10-05 11:06:12",
        "uploaded": "2022-10-05 11:14:05",
        "description": "Scanned On - Delivery Vehicle",
        "itemNumber": "1",
        "podName": ""
      },
      {
        "code": "UNL",
        "captured": "2022-10-05 12:13:54",
        "uploaded": "2022-10-05 12:19:29",
        "description": "Un Loaded",
        "itemNumber": "1",
        "podName": ""
      }
    ]
  }
}
```

```

    },
    {
      "code": "ARR",
      "captured": "2022-10-05 12:13:53",
      "uploaded": "2022-10-05 12:19:29",
      "description": "Arrived",
      "itemNumber": "1",
      "podName": ""
    }
  ]
}

```

Where the API has been requested to return data from a given date & time, the successful response will include all consignment Tracking Statuses found since between requested date & time.

```

{
  "success": true,
  "payload": {
    "results": [
      {
        "consignmentReferences": {
          "trackingCode": "1315550658796",
          "consignmentNumber": "KP58796",
          "customerReference1": "W-2104180",
          "customerReference2": ""
        },
        "statuses": [
          {
            "code": "EDC",
            "captured": "2022-10-05 09:55:24",
            "uploaded": "2022-10-05 10:03:25",
            "description": "Electronic POD",
            "itemNumber": "2",
            "podName": "salvo"
          },
          {
            "code": "EDC",
            "captured": "2022-10-05 09:55:24",
            "uploaded": "2022-10-05 10:03:25",
            "description": "Electronic POD",
            "itemNumber": "1",
            "podName": "salvo"
          },
          {
            "code": "DEP",
            "captured": "2022-10-05 09:55:33",
            "uploaded": "2022-10-05 10:03:24",
            "description": "Departed",
            "itemNumber": "2",
            "podName": ""
          },
          {
            "code": "DEP",
            "captured": "2022-10-05 09:55:33",
            "uploaded": "2022-10-05 10:03:24",
            "description": "Departed",
            "itemNumber": "1",

```

```

        "podName": ""
    }
]
},
{
    "consignmentReferences": {
        "trackingCode": "1315550658836",
        "consignmentNumber": "KP58836",
        "customerReference1": "W-2104157",
        "customerReference2": ""
    },
    "statuses": [
        {
            "code": "EDC",
            "captured": "2022-10-05 10:01:05",
            "uploaded": "2022-10-05 10:03:11",
            "description": "Electronic POD",
            "itemNumber": "1",
            "podName": "Stiv"
        },
        {
            "code": "ARR",
            "captured": "2022-10-05 09:54:04",
            "uploaded": "2022-10-05 10:03:10",
            "description": "Arrived",
            "itemNumber": "1",
            "podName": ""
        },
        {
            "code": "DEP",
            "captured": "2022-10-05 10:01:14",
            "uploaded": "2022-10-05 10:03:10",
            "description": "Departed",
            "itemNumber": "2",
            "podName": ""
        },
        {
            "code": "DEP",
            "captured": "2022-10-05 10:01:14",
            "uploaded": "2022-10-05 10:03:10",
            "description": "Departed",
            "itemNumber": "1",
            "podName": ""
        },
        {
            "code": "EDC",
            "captured": "2022-10-05 10:01:05",
            "uploaded": "2022-10-05 10:03:10",
            "description": "Electronic POD",
            "itemNumber": "2",
            "podName": "Stiv"
        },
        {
            "code": "UNL",
            "captured": "2022-10-05 09:56:59",
            "uploaded": "2022-10-05 10:03:09",
            "description": "Un Loaded",
            "itemNumber": "2",
            "podName": ""
        }
    ]
}

```

```

    },
    {
      "code": "UNL",
      "captured": "2022-10-05 09:56:59",
      "uploaded": "2022-10-05 10:03:09",
      "description": "Un Loaded",
      "itemNumber": "1",
      "podName": ""
    },
    {
      "code": "ARR",
      "captured": "2022-10-05 09:54:04",
      "uploaded": "2022-10-05 10:03:09",
      "description": "Arrived",
      "itemNumber": "2",
      "podName": ""
    }
  ]
}

```

Failure

Should no Tracking data be found for the requested consignment, using the API request via specific trackingCode, Consignment Number, or Customer References a response will be “204 No Content” with an empty response.

If the consignment cannot be found via the Tracking Code, Consignment Number, Customer Reference 1 or Customer Reference 2 specified a HTTP status of 404 will be returned.

```
{
  "success": false,
  "payload": {
    "errors": [
      "No Consignments Found"
    ]
  }
}
```

Should the request parameters return multiple consignments, the following HTTP status of 404 will be returned

```
{
  "success": false,
  "payload": {
    "errors": [
      "Multiple Consignments Found"
    ]
  }
}
```

Image Data Information Request Response

The Image ID Request response will contain the Image ID Data for the given parameters that have been requested. The response will contain the latest Image Data available for the trackingCode requested or all POD data since the Date & Time provided along with the type of image which relates to the UID provided. The API consumer should retrieve the image data from the URL provided.

The Document Content will contain the Image data for Electronic Signature Data, Scanned Proof of Delivery & Delivery Photographs.

Success

```
{
  "success": true,
  "payload": {
    "consignmentReferences": {
      "trackingCode": "1315550660669",
      "consignmentNumber": "KP60669",
      "customerReference1": "W-2109202",
      "customerReference2": ""
    },
    "images": [
      {
        "imageType": "DELIVERYPHOTO",
        "uId": "70459",
        "captured": "2022-10-06 10:00:00",
        "uploaded": "2022-10-06 16:23:16"
      },
      {
        "imageType": "POD",
        "uId": "1123792",
        "captured": null,
        "uploaded": "2022-10-06 16:19:05"
      }
    ]
  }
}
```

Where the API has been requested to return data between a given date range, the successful response will include all consignment POD data found since the requested date & time.

```
{
  "success": true,
  "payload": {
    "results": [
      {
        "consignmentReferences": {
          "trackingCode": "1315550658796",
          "consignmentNumber": "KP58796",
          "customerReference1": "W-2104180",
          "customerReference2": ""
        },
        "images": [
          {
            "imageType": "POD",
            "uId": "1123557",
            "captured": null,
            "uploaded": "2022-10-05 10:03:45"
          }
        ]
      }
    ]
  }
}
```

```

    ]
  },
  {
    "consignmentReferences": {
      "trackingCode": "1315550658836",
      "consignmentNumber": "KP58836",
      "customerReference1": "W-2104157",
      "customerReference2": ""
    },
    "images": [
      {
        "imageType": "POD",
        "uId": "1123564",
        "captured": null,
        "uploaded": "2022-10-05 10:03:51"
      }
    ]
  }
]
}

```

Requesting the base64 encoded image from the API will return as response that is detailed below. The base64 image will need to be decoded before this can be viewed and used.

```
{
  "success": true,
  "payload": {
    "data": "base64 encoded Image"
    "mimeType": "image/jpg",
    "filename": "8bb745ae723257e4eca2c49dd679ba4146dbec2b.jpg",
    "uniqueReference": "1123792"
  }
}
```

For the purposes of this document, there isn't an example base64 encoded image provided, but this string would be within the "data" key.

Failure

Should no Image data be found for the requested consignment, using the API request via specific trackingCode, Consignment Number, or Customer References a response will be "204 No Content" with an empty response.

If the consignment cannot be found via the Tracking Code, Consignment Number, Customer Reference 1 or Customer Reference 2 specified a HTTP status of 404 will be returned.

```
{
  "success": false,
  "payload": {
    "errors": [
      "No Consignments Found"
    ]
  }
}
```

Should the request parameters return multiple consignments, the following HTTP status of 404 will be returned

```
{
  "success": false,
  "payload": {
    "errors": [
      "Multiple Consignments Found"
    ]
  }
}
```

GET Configuration

The Configuration endpoint enables API users to retrieve permitted Goods Definition information automatically, eliminating the need for manual requests from the logistics provider.

The Configuration can be requested using the following URL:

```
/consignments/configurationsettings?code={apiKey}
```

This will be a GET request.

The GET Configuration endpoint returns a structured JSON response that provides details on available unit codes, associated billing units, allowed services, and applicable surcharges. The response contains a list of UnitCodes, each defined by a unique Code. Each unit code specifies an array of AllowedBillingUnits, indicating valid Pallet Types (e.g. Full Pallets). Additionally, each unit code includes an AllowedServices array, where each service is represented by a unique Service Code. Services also include details such as WorkingDays and MaxWorkingDays, which define the rules in which a service can be used,

Each service may have a list of AllowedSurcharges, with each surcharge containing a unique Surcharge Code. Surcharges, like services, include details and requirements for use.

GET Consignment

The GET Consignment endpoint allows consumers of the API to call a GET endpoint to retrieve complete details of a specific consignment. The Format of the response will match the format of a Consignment POST request to the API.

```
/consignments/{trackingcode}?code={apiKey}
```

This will be a GET request.

If the {trackingcode} used within the request header cannot be found, the following 404 error will be returned

```
{
  "Success": false,
  "Payload": {
    "Errors": [
      "Consignment Not Found"
    ]
  }
}
```

POST Customer Paperwork

This endpoint allows API user to send Customer Paperwork Images to the API, instead of having to manually add COP to each consignment within myPortal.

The Post Images can be used using the following URL:

```
/consignments/images?code={apiKey}
```

This will be a POST request.

Example JSON

Example JSON request to be used

```
{
  "imageType": "CUSTOMERPAPERWORK",
  "fileType": "image/jpeg",
  "imageName": "Example Paperwork Image.jpeg",
  "imageData": "base64 Encoded Image",
  "trackingCode": "1325560000591"
}
```

Properties

Property	Type	Comments
imageType	String	Only valid imageType to be used is CUSTOMERPAPERWORK
fileType	String	Valid types are; image/png, image/jpeg, image/tiff, application/pdf
imageName	String	Name of the Paperwork file being attached
imageData	String	Base64 encoded string of the image data
trackingCode	String	Must be the trackingcode of the consignment the paperwork is being attached to.

The maximum size limit for a single image is 10mb.

Failure Responses

If the {trackingcode} used within the request header cannot be found, the following 404 error will be returned

```
{
  "Success": false,
  "Payload": {
    "Errors": [
      "Sales Order Not Found"
    ]
  }
}
```

If the filetype provided is not valid, a 400 error will be returned with the following response

```
{  
  "Success": false,  
  "Payload": {  
    "Errors": [  
      "File type 'image/bmp' not allowed for attachment type CUSTOMERPAPERWORK. Valid types are image/jpeg,  
image/png, application/pdf, image/tiff"  
    ]  
  }  
}
```

API Key Request Form

Integrators Company Name:		
Software Product name:		
Depot Name:		
Customer Name:		
Account Code:		
Contacts		
Contact Name:		
Job Title:		
E-mail:		
Contact Name:		
Job Title:		
E-mail:		
Contact Name:		
Job Title:		
E-mail:		
Contact Name:		
Job Title:		
E-mail:		
Contact Name:		
Job Title:		
E-mail:		
Additional Contacts:		
Functions Required		
Feature	Required	Expected Daily Usage
Consignment Entry:		
Label Request:		
Additional Comments:		
(Please describe briefly how you intend to use the API)		

Once completed, please email the form to apisupport@vigosoftware.com

Appendices

Customer Implant

As well as the API functions described in this document Vigo also have a product called the Customer Implant that might be suitable for smaller volumes of consignments or situations where development to consume the API is not possible, but a CSV or XLS export of orders is.

The Implant runs in the background on a PC processing input files as they are deposited in a folder. The implant maps the content of the input into a format the Portal can consume and will upload to the portal and optionally produce the labels too.

Any invalid data can be modified in the implant UI.

Common Configuration Information

This table is designed to be filled out according to how the customer portal has been configured for this customer and will aid in understanding how to populate the JSON content used by functions in this API.

Cut Offs	
Is there a cut-off?	
Reference Information	
Is the Consignment Number Auto-generated	
Is there a Consignment Number prefix?	
Maximum Consignment No Length	
Job Types	
What job types are available i.e col/del col and del	
Collection Information	
Is Collection Name required?	
Is Collection Phone required?	
Is Collection SMS required?	
Is Collection Email required	
Delivery Information	
Is Delivery Name required?	
Is Delivery Phone required?	
Is Delivery SMS required?	
Is Delivery Email required	
Collection Notes	
Is Collection Note Line 1 Enabled?	
Is Collection Note Line 2 Enabled?	
Is Collection Note Line 3 Enabled?	
Is Collection Note Line 4 Enabled?	
Delivery Notes	
Is Delivery Note Line 1 Enabled?	
Is Delivery Note Line 2 Enabled?	
Is Delivery Note Line 3 Enabled?	
Is Delivery Note Line 4 Enabled?	
Is Delivery Note auto-prefix enabled?	
Items	
Item Types allowed:	
Is Oversize charging needed:	
Services	
Valid Services	
Valid Surcharges	
Service/Surcharge Combinations	
ADR	
Is ADR information support / required	