DX Despatch Manager

API Integration Guide

despatch-api.dxdelivery.com

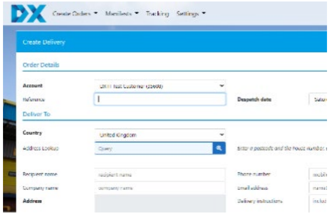
v2.2407.12.1120

Table of contents

1 Executive summary

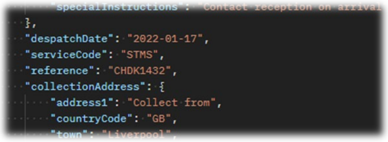
The DX Despatch Manager application is used to:

- create delivery and collection orders
- print labels
- view manifested orders
- view item tracking
- view proof of delivery



Account: UK1 Test Customer (35608)
Reference:
Despatch date: Satur
Country: United Kingdom
Address Lookup: Query
Enter a postcode and the house number
Recipient name: recipient name
Phone number: mobil
Company name: company name
Email address: name
Address
Delivery instructions: includ

The DX Despatch Manager API provides access to all the above features via a RESTful JSON web service that can be effectively integrated for high performance despatching and responsive customer service queries.



```
        "specialInstructions": "Contact reception on arrival
    },
    "despatchDate": "2022-01-17",
    "serviceCode": "STMS",
    "reference": "CHDK1432",
    "collectionAddress": {
        "address1": "Collect from",
        "countryCode": "GB",
        "town": "Liverpool",
```

This document provides the basic detail for developers to get started using the API immediately, understand its use for order creation in detail, and gives an overview of other features.

2 Document control

2.1 Document version

Version	Revision date	Author	Summary
1.2201.14.1242	15 Jan 2022	R Brame	First release
2.2203.14.833	31 Mar 2022	R Brame	Updated customs integration release
2.2206.21.1815	30 Jun 2022	Y Granger R Brame	Updated tracking release
2.2208.10.1209	12 Aug 2022	R Brame	Add location to tracking event
2.2406.13.953	13 Jun 2024	R Brame	Changes to support Windsor Framework Agreement, added NICustomsRequest
2.2407.12.1120	12 Jul 2024	R Brame	Add UKInternalMarketCompliant property to NICustomsRequest

2.2 Document authorisation

Document approver	Title	Date of approval
Jonathan Davies	Head of IT Software Development	
Robert Brame	IT Business Solutions Lead	
Mark Langford	IT Customer Integrations Manager	

2.3 Confidentiality

DX reserves all rights in its Confidential Information. The disclosure of Confidential Information by one party does not give the other party or any other person any licence or other right in respect of any Confidential Information beyond the rights expressly set out in this document.

This document is considered Confidential Information and must not be shared to any other party without the written permission of DX Network Services Limited.

3 Scope

This document describes the functionality and definition of the DX Despatch Manager API. The document introduces how the API can be used and may suggest the approach to be taken. It cannot instruct you on software development practices; how to design and build an integration; or how to configure your system(s) to work with the API.

4 Prerequisites

The DX Customer Integration team will supply alongside this document:

1. Credentials to access the integration test platform
2. The URL of the integration test platform, this document refers only to {{baseUrl}}

Developers may choose to use any suitable technology or framework that supports processing JSON and calling REST web services to access the DX Despatch Manager API. It is possible to call the API with any HTTP client but the additional benefits of using a mature framework cannot be overstated.

It is recommended that developers have established technical knowledge and experience of integrating to REST based APIs, and good familiarity with their choice of technology.

5 Integration test platform

DX require all API integrations to be implemented and proven against the test platform before a connection is permitted to the live API instance.

The test platform is a non-production environment that features:

- a fully working instance of the latest available API implementation
- available all day, every day – on a best endeavours basis
  (support / monitoring during office hours only)
- a suitably sized environment for testing; it is only indicative of the live environment performance and is not intended to support performance or stress testing
- customer configurations can be applied to mirror anticipated live services
- tracking events can be generated on request

Contact the DX Customer Integration Team for more details.

⚠ **Test environment data protection**
DX strongly recommends using only anonymised test data in line with your organisation's information security policy. Access to the test integration environment may be shared. The DX data retention policies do not apply to test environments. DX can on request remove customer test data post integration if required.

6 OpenAPI specification and sample Postman collection

An OpenAPI specification of the API is available to import into API tooling for creating mock requests and speed up development.

A Postman collection has also been included. This was created by importing the OpenAPI specification of the API and then further developing it to allow a simple walk though of authentication and creating of consignments. For more information on Postman visit https://www.postman.com/.

See the Integration Artifacts section 16 on how to access these supporting documents.

7 Integration overview

A typical integration will include the following high-level processes / steps to ensure that the design and implementation can be completed and validated by the developer, and then proven **as required by DX** before any go-live transition can take place.

1. **Evaluate** what the API offers and understand how to integrate to it; potentially using the Getting started section to create sample requests and prove the concepts required. Identify the appropriate technology that will be used, and the resources required to complete the integration.

2. **Consider** the integration approach to be taken from the perspective of the calling system; identify the appropriate point in the processing of orders to create a consignment and when to print a label; if and how consignments will need to be cancelled if not required; or re-created should the original input change (i.e. despatch date). How will errors be handled, what errors should result in the request being automatically reattempted and what must be manually corrected. Consider the quality of the contact details and address data available / collected for the end recipient.

3. **Determine** the DX services that will be offered and then understand what details are necessary for international services (if required) or based on the delivery method (i.e. contact details for notifications, delivery instructions for items that can be left, Transit Liability etc.). More details on the services available and what they offer are contained in the DX Express Service Guide.

4. **Develop** the integration against the DX test platform, aligned to the important guidelines in the Key concepts section, confirming the services expected are available. Ensure all the expected scenarios are covered; both those that might arise in the processing of orders in the calling system and the responses from the API (including potential errors).

5. **Validate** the integration in the test environment through applicable real world test scenarios, prove the integration using the same expected methods of processing. If the production system will process in batches, then test in batches. A formal test approach may cover test phases such as unit testing, system testing and integration testing – but remember the DX test platform is not intended for stress or performance testing. Ensure the DX labels print in high quality, to the correct size and aspect ratio, positioned correctly on the label media.

6. **Evidence** the testing carried out to supply the DX Customer Integration Team with sample data used, the resulting labels, and a summary of the methods called. The DX team have some visibility over the calls made to the API and will cross-reference what is provided against the test platform usage.

7. **Request** access to the API production instance and technical go-live approval. The DX Customer Integration Team will validate the supplied evidence and confirm if the integration is approved for use in production. Provide your IP address (list) to DX if you have not already done so.

8. **Confirm** access to the API in production and validate the integration from the calling system's production environment to the API in production, remembering that any test consignments must be cancelled, or they risk being charged.

9. **Go-live**! This is not purely a technical decision, the go-live check should ensure operational readiness on both sides to start processing, despatching, and collecting. Once confirmed by all parties the hard work put into this integration can now start to pay off!

10. **Monitoring**. The DX Customer Integration Team will monitor the integration in the production environment and work with you to investigate and resolve any potential issues that may arise.

8 Getting started

This section is a quick start guide to get you up and running quickly, covering how to:

- Connect to the test API
- Create an access token
- Get a list of available services
- Create a consignment
- Request a label

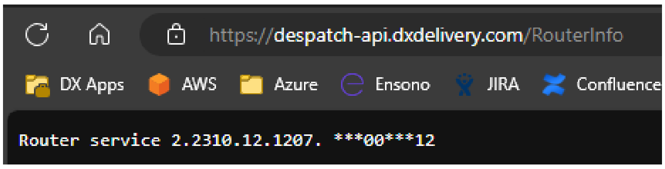
Examples are formatted as multi-line curl requests. Curl is a simple HTTP client that is freely available. Requests written in curl notation can be imported into other applications. Tools like Postman can import a curl request and output code samples in all common languages (C#, Java, Node.js, PHP, Python, Ruby etc.).

8.1 Connect to the test API

The DX Despatch Manager API supports plain GET requests that you can use to confirm you are able to connect to the servers hosting the test API. You can also open these links direct in a browser!

```
https://{{baseUrl}}/RouterInfo
```

```
https://{{baseUrl}}/Version/Index
```



```
Router service 2.2310.12.1207. ***00***12
```

In the examples we will use {{baseUrl}} to represent the location of the test API. The domain name (**and port**) must be configurable. The scheme can be fixed to https://

And requesting the API version from a REST client:

Curl request

```
curl -X 'GET' \
  'https://{{baseUrl}}/Version/Index \
  -H 'accept: */*'
```

To confirm the current version of the API always use **/Version/Index** not **/RouterInfo**

Response body (version number will vary)

```
2.2310.12.1207
```

If you see the version number being returned in plain text to your browser or REST client then you can access the test API from your current location.

Use these GET requests from any environment as a simple check that the relevant API is accessible on the appropriate domain name and port in use

Now you have hopefully connected OK and started to follow the documentation, getting used to the format of the notes, ideas, and important notices. The next section will deal with getting authenticated so we can start to make actual calls to the test API.

8.2 Create an access token

To access the API the calling client must first authenticate to obtain a valid access token. Tokens are valid for a fixed period after which a new token must be requested.

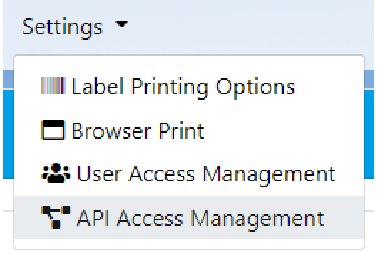
`https://{{baseUrl}}/Session/CreateToken`

The CreateToken method is called with header values:

- AccountCode
- Username
- Password

All authenticated API requests require the HTTP header: `Token`. The only unauthenticated requests are the version GET requests detailed in the last section and the CreateToken call used to create a new access token

In the test environment the DX Customer Integration team will provide these credentials.



In the production environment the DX Despatch Manager application allows super users to manage API credentials.

API credentials do not expire.
DX recommends rotating authentication credentials regularly in line with your organisation's information security policy

Curl request

```
curl -X 'POST' \
  'https://{{baseUrl}}/Session/CreateToken' \
  -H 'accountCode: {{AccountCode}}' \
  -H 'username: {{Username}}' \
  -H 'password: {{Password}}' \
  -d ''
```

Requests are not case-sensitive, URL paths, header names and JSON properties are shown in camelCase for ease of reading

Response body

```
{
  "accessToken": {
    "key": "FR8ofnWDJfKDfCq18fvah9Ab4XZzPhB2",
    "expiry": "2022-01-15T22:10:30.1838423+00:00"
  },
  "refreshToken": {
    "key": "AOPGS7fck9OnVTzp9CoamaRIptR6ojNm",
    "expiry": "2022-01-16T00:10:30.1838597+00:00"
  }
}
```

The token that must be provided as the token HTTP header is contained in the `accessToken.key` value. The token is valid until the time specified by the `accessToken.expiry` value.

The `refreshToken.key` can be used to request a new accessToken, more detail can be found in the Detailed API guide section of this document.

The calling system **must not** continually request a new or refreshed token, this is wasteful as the cryptographic processes involved consume additional resources and the API is optimised to support token reuse over re-authentication. Integrations will not be permitted to go-live if this is not respected.

8.3 Get a list of available services

You should now have connected to the API successfully and obtained a valid access token. Now let's move onto an authenticated request that will return the list of available courier services that you have access to:

- Availability is controlled by factors like delivery country, address (postcode in the UK)
- Access is based on your commercial agreement with DX

A call to the GetServices method requires a valid `accessToken.key` supplied as the `Token` HTTP header, and a JSON request to describe what services are required based on:

- Account code
- Delivery postcode
- Delivery country (described by its ISO Alpha-2 code)
- Service type (Delivery, Transfer or Return)

`https://{{baseUrl}}/Services/GetServices`

Curl request

```
curl -X 'POST' \
  'https://{{baseUrl}}/Services/GetServices' \
  -H 'Token: {{accessToken.key}}' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountCode": "{{AccountCode}}",
  "deliveryPostcode": "NN3 6FL",
  "deliveryCountryCode": "GB",
  "serviceType": "Delivery"
}'
```

In this example simple values have been included that can be used to try the example out. The sample data in this section is only the most straightforward example for a UK mainland customer. For other localised examples see the detailed documentation sections

Response body

```
[
  {
    "serviceCode": "NS",
    "name": "Secure Flex",
    "premium": false,
    "saturday": false,
    "sunday": false,
    "air": false,
    "timed": false,
    "slaTime": null,
    "sla": 2400,
    "deliveryMethod": "Signature",
    "signatureRequired": true,
    "neighbourSignature": true,
    "allowedCover": true,
    "maximumCover": 250,
    "standard": true,
    "supplement": false,
    "serviceLevel": "Next day",
    "days": 1,
    "b2BDeclarationType": "NotRequired",
    "b2CDeclarationType": "NotRequired",
    "serviceType": 1
  }
]
```

The service entity describes all the features of the service using both codes and descriptions.

Service codes do not change, but service availability does vary depending on the delivery address.

For an explanation of all the properties of a service see the API reference section.

The response body will contain all the services that are available based on the request body, and that the account has access to. To use the service only the `serviceCode` is required.

8.4 Create a consignment

Based on the available service code(s) obtained from the last section you can now create your first consignment using another simple example request.

The CreateConsignment method requires a valid `accessToken.key` supplied as the `Token` HTTP header, and a JSON request to describe the consignment, including these mandatory properties:

- Account code
- Despatch date
- Delivery address: recipient or company, address1, town, postcode (for GB), country code
- Service code
- Item details: quantity, package type, weight (in kg)

`https://{{baseUrl}}/Consignment/CreateConsignment`

Curl request

```
curl -X 'POST' \
  'https://{{baseUrl}}/Consignment/CreateConsignment' \
  -H 'Token: {{accessToken.key}}' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountCode": "{{AccountCode}}",
  "despatchDate": "2022-01-17",
  "deliveryAddress": {
    "company": "DX",
    "address1": "Test address",
    "town": "Northampton",
    "postcode": "NN3 6FL",
    "countryCode": "GB"
  },
  "serviceCode": "NS",
  "items": [
    {
      "quantity": 1,
      "packageTypeCode": "BX",
      "weight": 1.25
    }
  ]
}'
```

Despatch date will need to be set to today or a future date if past today's cut off

Package type codes and other codes accepted by the API can be requested from the API. `BX` is for a Box. See the Options section in the Detailed API documentation

Response body

```
{
  "service": {
    "serviceCode": "NS",
    "name": "Secure Flex",
    "deliveryMethod": "Signature",
    /* the whole service entity is returned from the API
       the full details have been omitted for clarity */
  },
  "trackingNumbers": [
    "7000000000"
  ]
}
```

A successful response will be a HTTP 200 status and contain a valid JSON object containing a `trackingNumbers` array of the tracking numbers assigned to each item label.

8.5 Request a label

A successful call to the CreateConsignment method returns the tracking number(s) assigned but it does not return the label(s) required to physically despatch the items.

The labels must be requested via a separate call to one of the API methods:

`https://{{baseUrl}}/Label/PDF`

`https://{{baseUrl}}/Label/ZPL`

The choice to use PDF or ZPL (Zebra Printer Language) will depend on the type of printers in use.

A ZPL label request only needs a `trackingNumber` and whether to print the whole consignment (true/false).

A PDF label request also supports specific paper sizes (inch4x4, inch4x6, A5, A4) and on larger sizes (A4) the starting location for the first label to be printed.

An example of both is below:

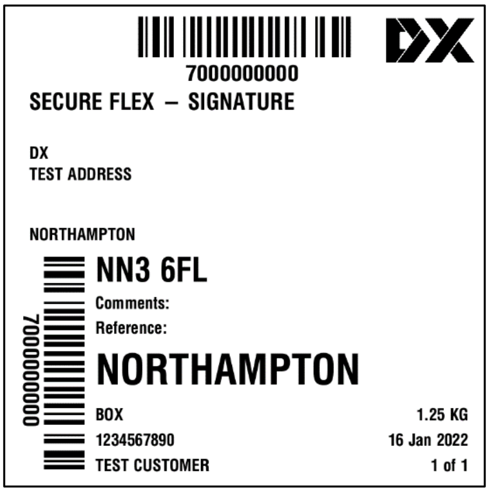
Curl PDF label request

```
curl -X 'POST' \
  'https://{{baseUrl}}/Label/PDF' \
  -H 'Token: {{accessToken.key}}' \
  -H 'Content-Type: application/json' \
  -d '{
  "trackingNumber": "7000000000",
  "printWholeConsignment": true,
  "pdfPaperSize": "inch4x4",
  "pdfA4StartingLocation": 1
}'
```

`inch4x4` is a 4-inch square thermal printer label and the size that the DX label is designed for.

Curl ZPL label request

```
curl -X 'POST' \
  'https://{{baseUrl}}/Label/ZPL' \
  -H 'Token: {{accessToken.key}}' \
  -H 'Content-Type: application/json' \
  -d '{
  "trackingNumber": "7000000000",
  "printWholeConsignment": true
}'
```



7000000000

SECURE FLEX – SIGNATURE

DX
TEST ADDRESS

NORTHAMPTON

NN3 6FL

Comments:

Reference:

NORTHAMPTON

BOX 1.25 KG

1234567890 16 Jan 2022

TEST CUSTOMER 1 of 1

This is an example of the rendered label that these sample requests would have produced, the tracking number, account number and customer name will be different on your version.

You have now completed the Getting started section of this document, and you have covered how to:

- ✓ Connect to the API
- ✓ Create an access token
- ✓ Get a list of available services
- ✓ Create a consignment
- ✓ Request a label

9 API overview

This section introduces all the API functions at a high level, as an overview and a guide to locating the key functionality the API offers.

The controllers of the API group the available methods into logical areas based on the entities they deal with, but the table below puts them in the context of the functionality offered by the API.

Functional area	Controllers	Methods
Authentication	Session	Session/CreateToken Session/CreateTokenWithRefreshToken Session/Session_Logout
Consigning / despatching	Consignment Label ReturnAddress Services	Consignment/CreateConsignment Label/PDF Label/ZPL ReturnAddress/GetAll Services/ConfirmService Services/GetServices
Customs declaration	Document	Document/Add Document/Attach
Cancellation	Consignment	Consignment/CancelItem
Data retrieval / Tracking	Consignment Manifest Tracking	Consignment/GetConsignment Consignment/GetItem Manifest/Get Tracking/GetItem Tracking/GetConsignment Tracking/GetAttachment Tracking/Search
Reference data	Options	Options/GetCountryCodes Options/GetPackageTypes Options/GetCurrencies Options/GetIncoTerms
Version	Monitor RouterInfo Version	/Monitor /RouterInfo /Version/Index

10 Key concepts

These are the key concepts that a developer or an integration must consider and abide by when integrating to the API. These important guidelines apply to all integrations and ensure that the availability of the API for all users is not compromised by any one user.

10.1 Connectivity

Check you can communicate with the API from your development system. As the test API instance is accessible via the public internet this should be straight forward, however client / IT management policies, proxy servers, firewalls and other IT controls may block the connection process and if so would need to be updated. Note that the production API instance is currently access controlled on the DX perimeter by an IP whitelist and you will need to communicate your IP address (list) to the DX Customer Integration Team in advance of go live.

DX normally makes two instances available to customers, each exists within its own segregated environment and has a specific endpoint with a different domain name and potentially a non-standard port number. The domain name (and port) should therefore be configurable based on the environment the calling system is connecting to.

The API is only accessible over HTTPS and supports both TLS 1.3 (recommended) and TLS 1.2 with ECDHE (Elliptic Curve Diffie-Hellman Key Exchange in ephemeral mode). The ciphers supported should be mirrored between the test environment and production, but if you believe your system may possibly not support the recommended TLS 1.3 configuration or TLS 1.2 ECHDE then contact the DX Customer Integration Team ASAP to ask how best to confirm this before go-live. The ciphers supported may need to change over time as security recommendations are updated, but DX will endeavour to give customers advance warning of any change.

> ⚠ Network devices between the calling system and the API may return a HTTP 200 response without the request reaching the API. The response should always be checked to confirm it contains the expected payload, even if that payload is an empty response.

10.2 Authentication

The API is secured to prevent unauthorised access to your data and to record the calling application on every request. This is done by requiring an authentication token to be obtained and used on all requests which is validated before the request is processed.

API credentials are required to create an authentication token, they are managed in the main DX Despatch Manager application by users with the superuser role. Note that the API credentials do not expire - DX recommends rotating credentials regularly in line with your organisation’s information security policy.

> ⚠ The calling system **must not** continually request a new or refreshed token, this is wasteful as the cryptographic processes involved consume additional resources and the API is optimised to support token reuse over re-authentication. Integrations will not be permitted to go-live if this is not respected.

The authentication token is returned with its expiry time, and a refresh token that can be used to obtain a new token. The refresh token has a longer expiry time and when used to refresh the authentication token another new refresh token is returned extending the expiry time further. This means in practice the calling system does not need to constantly provide the API credentials while a valid token is available, further securing the credentials.

10.3 Error handling

Error handling is a necessary part of any integration and will inevitably occur in practice if not during development or implementation. DX recommends appropriate error handling be implemented up front in the design and development process, and error conditions are incorporated into the test phases to ensure they are handled gracefully, and that the system resumes once transient errors have resolved.

Error categories

Errors fall into one of a few categories below and can represent a permanent failure due to the specifics of the request, or a transient issue that will be short lived and so the request should be retried. Requests should be retried with a suitable delay (e.g. 20 seconds) between attempts. If a transient error does not resolve within a few minutes, then an alert or notification should be raised within the calling system for the error to be investigated, and potentially raised with DX.

Category	Description	Action
Security	Likely an authentication or an authorisation issue. If the token is missing or has expired. Not having the correct permissions to make the request.	Request a new authentication token if the token has expired. Check the level of access granted to the API credentials access if permission is denied.
Syntax	If the body of the request does not contain the expected data, or field lengths do not confirm to the schema.	Repeating the same request will only return the same error. The calling system should continue to process other requests and raise an alert for the problem data to be investigated before being re-submitted.
Semantic	The request contains data that is formatted correctly but the data was not accepted by the API because it is not valid or not it is not possible to process it. Examples might be a code (or combination of codes) that is not recognised (service code, postcode, country code).	Repeating the same request will only return the same error. The calling system should continue to process other requests and raise an alert for the problem data to be investigated before being re-submitted. If creating a consignment, it may be that the parameters provide cannot be resolved to an available service, and the calling system should perhaps call for the available services based on the same parameters.
Transient	An unknown problem in the DX environment that may be caused by an outage or planned maintenance.	Transient errors should be retried after a suitable delay, and if not resolved in a few minutes, then raised for investigation.
Timeout	If the request to the API does not return a success or an error response. It may not be possible to reach the API, or the API took too long to respond.	Do not automatically retry a transaction in case the first request was processed. If possible, check a connection can be established though a simple GET request to the API. If there is an issue connecting, raise an alert for investigation.

HTTP error codes

The API will return a HTTP status code to indicate if the request was successful or not, as well as any payload. The HTTP codes returned are standardised and can be returned by other intermediate components (like proxy servers or client-side libraries). They are grouped into high level categories based on the first number of the code:

- 1xx – information
- 2xx – success
- 3xx – redirection
- 4xx – client errors
- 5xx – server errors

The following is a list of the common HTTP status codes that may be returned by the API itself:

Code	Message	Category	Description
204	No content	Semantic	The requested consignment was not found, either the tracking number is incorrect or that consignment has been archived.
400	Bad request	Syntax	The syntax of the request was not valid. This could be a missing or invalid parameter. The text response may give more information.
401	Unauthorised	Security	Returned when authentication is required, but the token was invalid or not provided. The token may also not have access to the correct account for the requested data. The text response may give more information (i.e. AccountCode, Invalid UserCredentials).
403	Forbidden	Security	The API credential that the provided token represents, does not have access to the requested method. Also returned if a user credential is used against the API (this is not allowed).
404	Not found	Syntax	The requested API method does not exist. This could be an incorrect method / URL, or it could be due to a server problem.
422	Un-processable entity	Semantic	The request was well-formed but could not be processed due to semantic errors. The text response may give more information.
429	Too Many Requests	Transient	The API has received too many requests from this account. The response will also contain these HTTP headers: • **Retry-After**: the number of seconds to wait before retrying • **X-Max-Allowed-RPM**: maximum requests/minute allowed
500	Internal server error	Transient	There has been an internal error processing the request. The request should be retried after a suitable delay, and if not resolved in a few minutes, then raised for investigation.

A HTTP error status will often be accompanied with additional information in the response body, this will usually be plain text and should be retained by the calling system, with details of the request, to help in any diagnostics later.

API error messages

There is a detailed list of API error messages in the appendices at the end of this document.

10.4 Timeouts

Use a suitable timeout when calling the API. Too short a timeout could lead to duplicates if calls to the API timeout while being processed, (especially if there is network congestion or high server load). We recommend setting the timeout to at least 90 seconds.

10.5 Multiple requests

The API will happily support multiple requests in parallel from the same calling system, but you must ensure you do not overload the API by initiating too many parallel requests. We recommend you use an appropriate method to control resources so no more than 5 parallel requests are sent to the API from the calling system.

The API contains rate limiting protection to prevent excessive numbers of requests being sent and potentially causing performance problems for other users. If the rate limit is breached the API will respond with a HTTP 429 status and HTTP headers that describe the maximum number per minute and how long to wait in seconds before retrying:

- **Retry-After**: the number of seconds to wait before retrying
- **X-Max-Allowed-RPM**: maximum requests/minute allowed

The default rate limit on the API is 20 requests per minute, which would suffice for most customers when despatching over the course of an entire day. The rate can however be increased to accommodate spikes in manifesting and for customers with higher-than-average volumes. Please contact the DX Customer Integration Team if you feel that the default rate will be too low for your integration.

10.6 Logging

Requests to the API are logged to ensure DX can support and investigate issues you may experience when integrating to the API. Only requests that are received by the API are logged, so any issues that prevent the request reaching the API will not be recorded i.e. network connection via proxies and firewalls.

DX recommends that all calls to the API are logged by the calling system to support investigation, with the minimum level of detail being the time (to the nearest millisecond), the URL requested, and the result returned. Error responses should be logged in full. Due to the loa din a production environment it may not be possible to record every detail, but ideally the logs in a test or development environment would include the entire request and response.

DX recommends appropriate logging be implemented up front in the design and development process, and validation of the logging is incorporated into the test phases alongside ensuring error scenarios are handled gracefully.

10.7 Data quality

The quality of data provided is critical to DX being able to provide the best possible service to you and the recipient(s).

Ensuring the following details are valid and correctly represented will contribute to a good experience for the recipient:

- ✓ Mobile phone number
- ✓ Email address
- ✓ Collection and/or delivery address(es)

The correctness of the data you provide via the API will also ensure that no delays, additional surcharges, or customs fines are incurred for under declaration:

- ✓ Despatch date
- ✓ Package type (identifying size in particular)
- ✓ Weight
- ✓ Customs details (if required):
  - ✓ Commodity code
  - ✓ Commercial value
  - ✓ Country of origin
  - ✓ Recipient EORI
  - ✓ Supporting documentation

Address data quality

Creating orders with valid, correctly formatted addresses is essential. To ensure the collection or delivery is performed at the correct location, and supporting route planning, which in turn allows DX to generate ETAs and give visibility of the courier enroute to the address.

Address validation starts at the point of receiving the order from the end customer, or the setup of a customer account if taking repeated orders. DX recommends all address entry is done using an address validation service to locate the address and format it correctly.

A valid UK address should always contain the following elements:

- ✓ **Postcode**. Correctly formatted with a single space between the first section and the last 3 characters i.e. “NN3 6FL”
- ✓ **Town**. The town or city that the address is in, or most local to in the case of villages.
- ✓ **House or premise details**. This can vary considerably between residential addresses where a house number and street are most common, to residential flats and apartments, industrial business ‘units’, and commercial properties. For addresses with a unit, flat or house number, the correct number is clearly very important.

Example address formats (not real addresses):

18 Acacia Avenue
Milton Keynes
MK14 3ER

Unit 1
Mill Close
Intercity Trading Estate
Wolverhampton
WV1 3DA

Flat 1
St. James House
10-12 High Street
Ryton-In-Arden
Coventry
CV11 2ET

A UK address **must not** contain:

- ✖ **County**. This must not be included; the postcode is used instead of it. Modern datasets do not contain county information, and it is a potential source of confusion for automated routing software. The UK postcode database can contain post county information, but it has not been required as part of the address on postal services since 1996. The inclusion of a county name is now considered 'vanity addressing' as a recipient's preferred county did not always align to the actual postal county.

All addresses **must not** contain:

- ✖ **Personal information** or contact details. These details are required, but in other data fields not as part of the address. To include personal details in the address or any incorrect field is a data protection issue, as the details may be printed on labels and manifests (which are less controlled when printed), and they are not intended to be present there.
- ✖ **Surplus data**. As outlined in some of the cases below, the presence of surplus data is too often not an improvement on an address and instead causes confusion for automated routing software; impacting on DX's ability to deliver a good service.
- ✖ **Vanity addressing**. This might be a house name, or a local area name, sometimes the name of the original housing development. Recipients attempt to include this information because they prefer how the address appears, or the belief that it improves the address in some way. This is only vanity the actual address is defined by the required details above.
- ✖ **Instructions**. Collection or delivery instructions if available must be provided as the 'SpecialInstructions' data field, which is then highlighted to the courier on their handheld device when an instruction exists, as well as on the label and manifest as required.

A Republic of Ireland address can contain:

- ✓ **Eircode**. Correctly formatted with a single space between the first 3 and last 4 characters i.e. "A65 D4T2"
- ✓ **County**. This is required for addresses that are otherwise ambiguous without the county name.

10.8 Tracking

DX endeavours to make detailed tracking information available to the sender and recipient to keep them informed of the progress of their item or request. Different methods of accessing tracking are provided to suit the relevant party and the specific use cases including:

- For the sender **despatch.dxdelivery.com** can be used within an organisation to provide tracking to customer service agents
- For the recipient, DX will send notifications via email and/or text depending on the service chosen, to keep the recipient informed and direct them to the DX website for more information
- For the recipient, the DX website **my.dxdelivery.com** provides up to the minute information on the progress of their item, with the ETA and location of the courier displayed in relation to their address (when available, depending on the service chosen)

The DX Despatch Manager API provides tracking events in two ways:

- **Interactively**, that is to support someone (or something) needing to obtain information about a specific item, at
- **Continuously**, to provide updated information that can be regularly requested, downloaded, and made available from a customer's own system

⚠ The correct approach to obtain tracking **must** be used. DX will monitor and act to prevent misuse of the API being called repeatedly for updated tracking information on individual items when the updates should be loaded continuously. The maximum number of requests per minute (RPM) allowed will **not** be increased to allow higher volumes of tracking requests to be made.

Interactive tracking

The methods that support interactive requests for tracking also assist with locating the correct items to view tracking for. In practice it may not be necessary in a live integration to use the search methods that locate items, as all items created via the API return the unique DX tracking number assigned. The tracking number is then used to obtain tracking events directly, potentially on all items in the same consignment.

Tracking method	Use
Search	Search for items and consignments based on reference number, recipient, contact name, company, collection or delivery address, and collection or delivery postcode
GetConsignment	Returns all the details of the consignment as well as all tracking events for each item
GetItem	Returns tracking events for a single item
GetAttachment	Returns the photo or POD signature image associated with an event

Consider using the above methods during development to obtain example events, with the support of the DX Customer Integration Team tracking can be made available in the test environment.
The best approach for most integrations is however to not use these methods in production, instead using the continuous update method to always have the latest events available across all items.

Continuous tracking updates

The Tracking **GetUpdates** method returns new events based on the parameters supplied.

If tracking events are required by a calling system, then updates should be downloaded from the API continuously to benefit from 'live' updated information on each item being delivered or collected by DX, made available to the caller at the same time as all other DX tracking facilities.

Events may be delayed from any mobile device or routine process; they may also arrive out of order of when they occurred. Events are only generated from the point an item is received, there will be no event returned from a consignment being created.

The process of requesting updates allows the calling system to control; how many updates are provided in each response; and the next update to be returned based on the previous response. The calling system must provide the identifier 'FromTrackingID' of the last update that was successfully processed, and the API will then respond with the next updates in the chronological order they were made available.

⚠ The FromTrackingID is quite obviously a sequential value, but the format may change over time. Calling systems **must not** attempt to infer any meaning from the value and should allow for the value to be up to 50 characters in length.

Update frequency

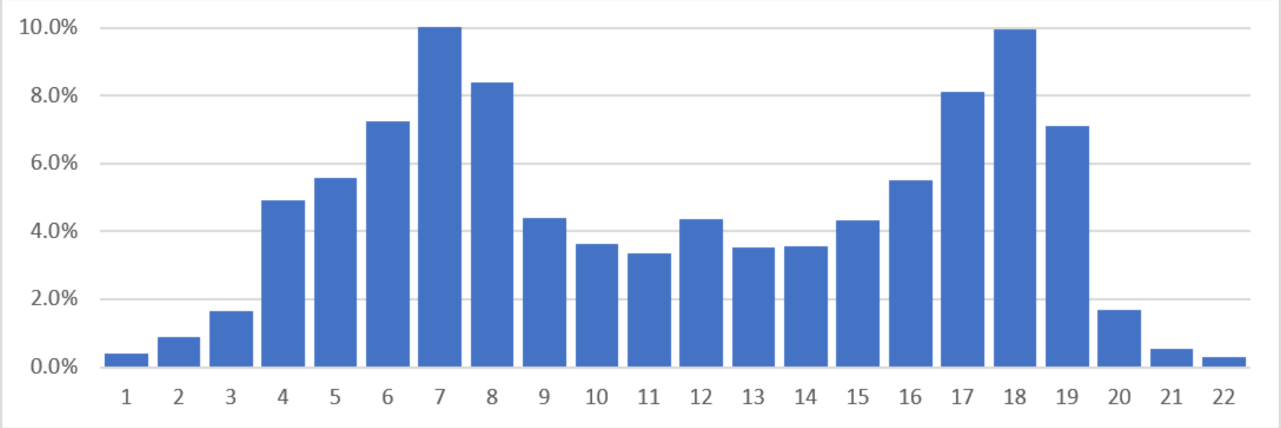
To keep up with the available updates, it is necessary to request updates periodically, throughout the day. Numerous events are generated overnight, so the calling system would ideally not stop or pause outside of business hours (though it would be acceptable to reduce the frequency based on the time of day). The interval between calls for updates may depend on the capability of the calling system, the volume of consignments created, and how the updates are planned to be used. DX recommend leaving an interval of 1 to 5 minutes between calls once no more results are returned. If any updates are returned, it is recommended to call again for further updates. Even if the number of updates is less than the maximum requested – the maximum is an upper limit only – the API may respond with slightly less than the maximum due to the methods of filtering used internally.

⚠ If no new updates are returned, the calling system must not call for updates again within 1 minute of the last call made. DX will monitor and act to prevent misuse of the API in this regard.

Performance considerations

The typical profile of available updates throughout the day would be a high number of events occurring at the beginning of the working day when couriers are departing from depots, and then later in the day at the point of newly collected items being received into the network (based on the collection time and the time taken to return to the depot).

The number of events per item is on average 5 – since those usually occur within a 24-hour period – we would suggest when sizing the number of events per item per day to expect; 5 is a reasonable amount. This will of course vary based on the service chosen and the success ratio of delivery/collection attempts.



Typical profile of events returned over a day

In the morning peak an event should be returned for every item being delivered that day, typically between 06:00 and 09:00 you could expect to receive 25% of the total volume of events for that day. The later peak is very much dependant on the collection time(s) agreed with DX but expect a received event for each item as they are processed at the collection depot.

Legacy tracking details

The API also returns legacy event codes and descriptions, which we recommend the calling system should **ignore** unless there is a specific legacy support requirement identified.

DX provides these legacy details with no guarantees of the ongoing support available and may in the future change or remove codes as required based on changing business practices.

For more information on how DX can support any bespoke needs for coded tracking events please contact the DX Customer Integration Team.

11 Consignment creation process

The process to create a consignment is as follows:

1. **Authenticate with the API**. To access the API an access token must be obtained by calling the `CreateToken` method. All the authenticated requests require the `AccessToken` returned by this method to be provided in the request header. The token should be cached and re-used until it expires, when it can either be refreshed or a new token created by re-authenticating.
2. **Determine the service code**. This is done by calling the `GetServices` method with a service type and the delivery or collection location. It can either be done every time a consignment is created or as and when needed to ascertain the services available. Service codes can also be validated using the `ConfirmService` method by providing the same details. Service entities also return any customs declaration requirements that may exist based on the collection and delivery location.
3. **Create a consignment**. Each consignment is created by calling the `CreateConsignment` method with all the details required based on the service type. The API will validate the request and create tracking number(s) for the quantity of items contained in the request.
4. **Print labels**. Call the appropriate Label method to request the correct labels for the printer in use. PDF labels for plain paper printers using `Label/PDF` or ZPL labels for Zebra thermal printers using `Label/ZPL`. A single label can be returned or the whole consignment if required. If a label is requested more than once (reprinted) then the original is cancelled.
5. **Submit customs documents**. This step (if required) can happen before or after the consignment is created. Call `Document/Add` to upload each document and then `Document/Attach` if the consignment has already been created. The exact documents required to support the customs regulations in force are outside the scope of this document, see section 13.1 for more detail.
6. **Cancellations**. When labels or items are created are not actually required, a call to `CancelItem` will request that label/item be cancelled. It may not be possible to cancel the item if it has been processed or received. All items in a consignment must be cancelled individually. Requests to cancel deliveries must be made on or before the despatch date, requests to cancel collections must be made before the requested collection date.

The **Returns** service type requires that a return address key be provided to the `CreateConsignment` method, the return addresses allowed can be retrieved by calling the `ReturnAddress/GetAll` method.

There are also supporting methods that provide reference codes to be used when creating a consignment. The following data entities are described by the API which returns a list of available codes from the Options controller using the appropriate method for each:

- Countries
- Currencies
- Incoterms (International Commercial Terms)
- Package types

12 Viewing and tracking consignments created

Consignment details submitted can be retrieved from the API. As each item is processed by DX tracking events are generated and can also be requested:

1. **Manifested consignments**. Use the `Manifest/Get` method to return summaries of consignments based on their despatch date.
2. **Consignment details**. Call the `Consignment/GetConsignment` method to return the consignment details and the details of all the items within a consignment.
3. **Item details**. Call `Consignment/GetItem` to return the consignment details and the details of just one item within a consignment.
4. **Search**. Using the `Tracking/Search` method to search for items based on tracking number, reference, address, recipient, or company name.
5. **Consignment Tracking**. The `Tracking/GetConsignment` method returns the same details as the `Consignment/GetConsignment` method with tracking events for all items.
6. **Item tracking**. The `Tracking/GetItem` method just returns tracking events for the item requested.
7. **Tracking attachments**. Depending on the service requested, tracking events may have attachments that are requested from the API by calling the `Tracking/GetAttachment` method.

13 Customs declarations

A customs declaration may be required when a consignment is being delivered outside of the account's registered customs territory, based on the regulations in force. The DX Despatch Manager API describes the level of customs declaration required based on the account and the delivery country / address. Consignments that do not include declaration details when expected will be rejected by the API as invalid.

> ⚠ **Customs declarations are a legal obligation**
> A customs declaration is a legal document required for security and compliance with the rules enforced by customs authorities. Information provided must be accurate, otherwise delays and penalties may be applied.

This guide does not cover in detail all the customs processes and obligations required; it details the API support available for creating consignments with DX. For more details or support on customs handling and how DX can help, please contact the DX Customer Integration Team or your DX Sales / Account Manager in the first instance.

A call to `GetServices` will return the type of declaration expected, based on the delivery being either a B2B or B2C movement. Only one declaration type can be provided for each consignment:

- **Basic**
  Contents description and value are required for deliveries to the Channels Islands (Jersey / Guernsey)

- **Full**
  Additional recipient and invoice details with commodity level information (quantity, origin, and classification / tariff code) and supporting documentation (commercial invoice / packing list)

- **NI**
  Based on the Windsor Framework Agreement, coming into force on Monday 30th September 2024; additional information is required for deliveries to Northern Ireland. Although this is not a full

customs declaration, it requires similar information for a movement to NI, has a similar structure as a Full declaration, but some fields are not required, and some are optional. Although `CommodityCode` is not a mandatory field, it is a requirement for DX to move the goods, and not supplying this may delay the goods being delivered.

Northern Ireland B2C restrictions
A consignment can only be classed as B2C if the goods are being sent by a business in Great Britain to a private individual residing in Northern Ireland for personal use only. It must also comply with weight limits; the package must not exceed 31.5kg if it contains more than one item, or 100kg if it contains a single item. Any consignment that does not confirm to this must be declared as B2B.

- **Consumer**
  If allowed by the relevant regulations in force, a (B2C) consumer transaction may not require a detailed customs declaration. This type of declaration is to confirm that the recipient of the goods is a consumer and therefore no detailed declaration is required.
- **Not required**
  Where there are no customs requirements in force between the collection and delivery address then the customs declaration is not required.

13.1 Supporting documentation

A full customs declaration requires the following additional documentation to be provided when available:

- Commercial invoice (mandatory)
- Packing list (optional)
- Or a combined document containing both

Documents can be uploaded in JPG, PNG image or PDF format. Scanned copies are accepted but ideally original electronic versions should be supplied.

Ideally documents would be uploaded when a consignment is created, but in some cases, this may not be possible, or the documentation might reference the tracking number, in which case it can be uploaded after the consignment has been created.

Either upload the document(s) first:

1. Call `Document/Add` to upload each document
2. Then call `CreateConsignment` with the document number(s) returned by the first call

Or if uploading the document(s) afterwards:

1. Call `Document/Add` to upload each document
2. Then call `Document/Attach` to associate the document(s) with the consignment that has already been created

14 API order reference

This section lists all the API methods related to ordering and their required parameters, with notes on usage of each parameter. Shared schema types are listed first and referenced from the methods that require them. Customs specific types are listed in their own section after the main methods.

> Requests are not case-sensitive, URL paths, header names and JSON properties are shown in CamelCase for ease of reading

14.1 Shared order schema types

Address

Delivery or collection address for a consignment.

Address type			
Property	**Type**	**Required**	**Use**
Recipient	String(30)	Yes, if no company name	The name of the recipient or contact for this address. There is no restriction placed on the service by the recipient's name being specified
Company	String(40)	Yes, if no recipient name	The name of the company for a business address
Address1	String(50)	Yes	First line of the address, i.e. building name or house number and street
Address2	String(50)	No	Second line of the address
Address3	String(50)	No	Third line of the address
Town	String(30)	Yes	The town or city of the address
County	String(30)	No	**Ireland only**. Do not supply the county on a UK address it is not required.
Postcode	String(8)	Yes, if not IE	The valid postcode assigned, required for all addresses outside of Ireland. Eircode should be provided on an Irish address.
CountryCode	String(2)	Yes	ISO 3166-1 alpha-2 country code. The full list can be obtained by calling `Options/GetCountryCodes`
Phone	String(15)	No	Phone number with full STD code. International formats are also accepted: 0044 +44 etc. To potentially receive SMS notifications a UK mobile number must be provided.
Email	String(50)	No	Recipient Email address. To potentially receive email notifications this must be provided.
SpecialInstructions	String(75)	No	Instructions to the courier, to locate the address or if the service permits such a delivery - to also advise a suitable leave safe location or neighbour to sign.

ConsignmentResponse

Details of a consignment and one or more items depending on how it was requested.

ConsignmentResponse type			
Property	**Type**	**Required**	**Contains**
AccountCode	String(20)	Yes	The account code that the consignment was booked against
Reference	String(30)	No	Reference number supplied when created.
DespatchDate	Datetime	Yes	Date provided for despatch (or collection)
CollectionAddress	Address	No	Collection address
DeliveryAddress	Address	Yes	Delivery address
Service	Service	Yes	Service type describing service attributes including the service level the consignment was booked under
CoverRequired	Boolean	No	True if additional Transit Liability cover was requested
CoverValue	Integer	No	The value of the additional Transit Liability cover requested in the local currency (UK £, IE €). Whole number only
Items	ItemResponse[]	Yes	Array of type ItemResponse describing either **all** the items in the consignment, or the **single** item requested depending on the method used
Customs	CustomsResponse	No	If required based on the service available to the collection and delivery location. See section 13 on Customs declarations

CoverValueCurrency	String	No	One of the allowed ISO 4217 currency codes:
			EUR = Euro
			GBP = Pound sterling

ItemResponse

Detail of an item in a consignment.

Item type			
Property	**Type**	**Required**	**Contains**
TrackingNumber	String(20)	Yes	Item tracking number
Quantity	Integer	Yes	Fixed value of 1
PackageTypeName	String(50)	Yes	Type of the package e.g. Box, Pallet, Length. The full list can be obtained by calling `Options/GetPackageTypes`
PackageTypeCode	String(3)	Yes	Code representing the package type. The full list can be obtained by calling `Options/GetPackageTypes`
Weight	Decimal	Yes	Total weight in kilogrammes (kg) of these items to 2 decimal places
Cancelled	Boolean	Yes	True if this item has been requested to be cancelled

ItemRequest

Quantity of each package type specified when creating a new consignment.

ItemRequest type			
Property	**Type**	**Required**	**Use**
Quantity	Integer	Yes	The number of items that are this type of package
PackageTypeCode	String(3)	Yes	PackageType code describing the type of the package as agreed with DX e.g. Boxes, Pallets, Lengths etc. The full list can be obtained by calling `Options/GetPackageTypes`
Weight	Decimal	Yes	Total weight in kilogrammes (kg) of these items to 2 decimal places

Service

Describes all the attributes of a DX service.

Service type			
Property	**Type**	**Required**	**Contains**
ServiceCode	String(20)	Yes	Unique code for this service (e.g. LS, MS, MSPre1)
Name	String(30)	Yes	Secure (Flex, Business, Transfer, Return)
Premium	Boolean	Yes	True if this service may incur a premium
Saturday	Boolean	Yes	True if deliveries will be attempted on a Saturday
Sunday	Boolean	Yes	Reserved for future use.
Air	Boolean	Yes	True if using an air route for a faster service
Timed	Boolean	Yes	True if this service has a timed SLA (e.g. Pre 1)
SLATime	String(5)	No	Timed SLA in HH:mm (i.e. 13:00)
SLA	Integer	Yes	Numeric form of the SLA (e.g. 1300, 2400, 4800)
DeliveryMethod	String(30)	Yes	DX delivery method (see DX Express Service Guide)
SignatureRequired	Boolean	Yes	True if this item must be signed for
NeighbourSignature	Boolean	Yes	True if a neighbour can receive and sign for the item
AllowedCover	Boolean	Yes	True if additional Transit Liability cover can be specified
MaximumCover	Integer	No	Maximum amount of additional Transit Liability cover allowed
Standard	Boolean	Yes	Reserved for future use.
Supplement	Boolean	Yes	True if the location may incur supplementary cost
ServiceLevel	String(30)	Yes	DX service level (e.g. Next Day, Pre 1)
Days	Integer	Yes	Number of days for delivery to this location
B2BDeclarationType	String(30)	Yes	Customs requirement for business orders
B2CDeclarationType	String(30)	Yes	Customs requirement for consumer orders
ServiceType	ServiceType	Yes	Enumeration containing: 1 = Delivery 2 = Return 3 = Transfer

14.2 Consignment controller

CreateConsignment

Creates a new delivery or collection consignment.

Method			
/Consignment/CreateConsignment			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
AccountCode	String(20)	Yes	Supply the account code to be used to obtain services and that the consignment will be booked against
Reference	String(30)	No	Your reference number for this consignment / order number. Can be used to search for this consignment and may be included in invoices, reports and notifications from DX
DespatchDate	Datetime	Yes	For a delivery: the date this item will be handed over to DX. If a collection has already taken place today, then the next collection day. For a collection: the date the collection should take place. If the cut off has already passed today, then the next working day
CollectionAddress	Address	No	For a Transfer or Return service type the address the consignment is to be collected from
DeliveryAddress	Address	Yes	The delivery address
ServiceCode	String(20)	Yes	A valid service code that is available to the delivery (and collection) address
CoverRequired	Boolean	No	True if additional Transit Liability cover is required on this consignment
CoverValue	Integer	No	The value of the additional Transit Liability cover required in the local currency (UK £, IE €)
Items	ItemRequest[]	Yes	Array of type ItemRequest describing the items in the consignment; the quantity of each package type, and the total weight
Customs	CustomsRequest	No	If required based on the service available to the collection and delivery location. See section 13 on Customs declarations
DontDowngradeService	Boolean	No	If True then requests with premium services that are not available will be rejected as invalid, rather than being downgraded to the next available service level
ReturnAddressKey	String(36)	No	Returns service type, the identifier for the returns address

/Consignment/CreateConsignment Response		
Name	**Type**	**Contains**
Service	Service	The service requested, updated to reflect the service level possible if the request was downgraded
TrackingNumbers	String(20)[]	Array of tracking numbers created

GetConsignment

Returns the details entered for a consignment including **all** the items in that consignment.

Method			
/Consignment/GetConsignment			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	Any tracking number contained within the consignment will return the consignment and the details entered

/Consignment/GetConsignment Response		
Name	**Type**	**Contains**
Consignment	ConsignmentResponse	The details of the consignment, including the details of **all** the items in the consignment

GetItem

Returns the details entered for a consignment and only the **single** item requested.

If the consignment requested cannot be found, the HTTP 204 ‘No content’ status code will be returned.

Method			
/Consignment/GetItem			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	The tracking number of the item being requested

/Consignment/GetItem Response		
Name	**Type**	**Contains**
Consignment	ConsignmentResponse	The details of the consignment and only the **single** item requested

CancelItem

Cancels a single item. To cancel a whole consignment every item in the consignment must be cancelled.

Method			
/Consignment/CancelItem			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	The tracking number of the item to cancel

/Consignment/CancelItem Response
HTTP 200 Success

14.3 Document controller

Add

Add a document as part of a customs declaration for a consignment. See section on Customs declarations.

Method			
/Document/Add			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
AccountCode	String(20)	Yes	The account code the consignment has been booked under. Documents cannot be shared between accounts
DocumentFilename	String(50)	Yes	Filename of the document including the file extension
DocumentData	String	Yes	Base64 encoded file content
DocumentType	DocumentTypes	Yes	An enumeration value describing the document: 1 = CommercialInvoice 2 = PackingList 3 = CommericalInvoicePackingList See section on Customs declarations

/Document/Add Response		
Name	**Type**	**Contains**
DXDocumentNumber	String(20)	A unique reference for the document submitted, to use when creating a consignment or to attach it to a consignment already created.

Attach

Attach a document as part of a customs declaration for a consignment. See section on Customs declarations.

Method			
/Document/Attach			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
DXDocumentNumber	String(20)	Yes	The unique reference of the document submitted using the `Documents/Add` method.
TrackingNumber	String(20)	Yes	Any tracking number in the consignment that this document relates to.

/Document/Attach Response
HTTP 200 Success

14.4 Label controller

PDF

Render a PDF label for the requested item or consignment.

Method			
/Label/PDF			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	The tracking number of an item or consignment to print
PrintWholeConsignment	Boolean	No	If True, then labels will be rendered for all items in the consignment
PdfPaperSize	PdfPaperSizes	Yes	Enumeration value for the paper size required: 1 = A4 2 = A6 3 = inch4x4 4 = inch4x6
PdfA4StartingLocation	Integer	Yes	A4 only, where on the page to start printing based on a left to right, top to bottom order: 1 = Top Left 2 = Top Right 3 = Bottom Left 4 = Bottom Right

/Label/PDF Response		
Name	**Type**	**Contains**
TrackingNumbers	String(20)[]	Array of tracking numbers for the labels rendered. When labels are reprinted the tracking number assigned changes, the new tracking number will be returned.
Data	String	Base64 encoded PDF data
LabelType	LabelTypes	Fixed enumeration value for the label type rendered: 1 = PDF
MimeType	String(100)	Fixed value of “application/pdf”
PaperSize	PdfPaperSizes	Enumeration value for the paper size rendered: 1 = A4 2 = A6 3 = inch4x4 4 = inch4x6
StartingLocation	Integer	A4 only, where on the page to start printing based on a left to right, top to bottom order: 1 = Top Left 2 = Top Right 3 = Bottom Left 4 = Bottom Right

ZPL

Render a ZPL (Zebra Printer Language) label for the requested item or consignment.

Method			
/Label/ZPL			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	The tracking number of an item or consignment to print
PrintWholeConsignment	Boolean	No	If True, then labels will be rendered for all items in the consignment

/Label/ZPL Response		
Name	**Type**	**Contains**
TrackingNumbers	String(20)[]	Array of tracking numbers for the labels rendered. When labels are reprinted the tracking number assigned changes, the new tracking number will be returned.
Data	String	Plain text ZPL content
LabelType	LabelTypes	Fixed enumeration value for the label type rendered: 0 = ZPL
MimeType	String(100)	Fixed value of “text/plain”
PaperSize	PdfPaperSizes	Fixed value 0
StartingLocation	Integer	Fixed value 0

14.5 Customs schema types

CustomsRequest

The customs request object contains one of the specific customs request objects.

CustomsRequest type			
Property	**Type**	**Required**	**Contains**
Basic	BasicCustomsRequest	No	Channel Islands declaration
Full	FullCustomsRequest	No	Detailed commodity level declaration
NI	NICustomsRequest	No	Northern Ireland declaration under the Windsor Framework
Consumer	ConsumerCustomsRequest	No	Consumer declaration (exception)

CustomsResponse

The customs response type contains the same fields as the CustomsRequest above with one additional field:

CustomsResponse additional properties (in addition to CustomsRequest)			
Property	**Type**	**Required**	**Contains**
Documents	Document[]	No	Details of the supporting documentation uploaded and associated with this consignment. See section 13.1 for more detail on supporting documents

Document

Details of a supporting customs document uploaded to DX.

ConsumerCustomsRequest type			
Property	**Type**	**Required**	**Contains**
DxDocumentNumber	String(20)	Yes	Assigned DX document identifier
DocumentFilename	String(50)	Yes	Filename including extension
DocumentType	Integer	Yes	Enumeration value for the document type: 1 = CommericalInvoice 2 = PackingList 3 = CommericalInvoicePackingList (Value 3 indicates a combined invoice and packing list). See section 13.1 for more detail on supporting documents

BasicCustomsRequest

Contents description and value as required for deliveries to the Channels Island (Jersey / Guernsey).

ConsumerCustomsRequest type			
Property	**Type**	**Required**	**Contains**
MovementType	String(3)	Yes	Enumeration value for the transaction / movement type: 1 = B2B 2 = B2C
GoodsValue	Decimal	Yes	Commercial value of the goods

GoodsValueCurrency	String(3)	Yes	The ISO 4217 currency code that the GoodsValue property represents:
			EUR = Euro
			GBP = Pound sterling

Contents	String(200)	Yes	A detailed and accurate description of the goods. Vague descriptions are more likely to be delayed clearing customs

BasicCustomsResponse

The BasicCustomsResponse type contains the same fields as the BasicCustomsRequest above with one additional field:

BasicCustomsResponse additional properties (in addition to BasicCustomsRequest)			
Property	**Type**	**Required**	**Contains**
DxCustomsNo	String(20)	No	Assigned DX customs identifier

ConsumerCustomsRequest

If allowed by the relevant regulations in force, a (B2C) consumer transaction may not require a detailed customs declaration. This type of declaration is to confirm that the recipient of the goods is a consumer and therefore no detailed declaration is required.

ConsumerCustomsRequest type			
Property	**Type**	**Required**	**Contains**
MovementType	String(3)	Yes	Enumeration value for the transaction / movement type: 1 = B2B* 2 = B2C *Note that the B2B type being provided on a ConsumerCustomsRequest would be invalid

ConsumerCustomsResponse

The ConsumerCustomsResponse type is identical to the ConsumerCustomsRequest above.

FullCustomsRequest

Required for items moving between international customs territories where no other arrangements exist to simplify trade (i.e. customs union).

FullCustomsRequest type			
Property	**Type**	**Required**	**Contains**
MovementType	String(3)	Yes	Enumeration value for the transaction / movement type: 1 = B2B 2 = B2C
RecipientEoriCode	String(20)	No	The EORI (Economic Operators Registration and Identification) number for the organisation or commercial recipient. Not required for (B2C) consumer movements
InvoiceNumber	String(100)	Yes	The commercial invoice number for the transaction associated with this consignment
InvoiceValue	Decimal	Yes	Invoiced value of the goods in the consignment

InvoiceValueCurrency	String(3)	Yes	The ISO 4217 currency code that the InvoiceValue property represents:
			EUR = Euro
			GBP = Pound sterling

IncoTermsCode	String(3)	No	International commercial terms code. The list available can be obtained by calling `Options/GetIncoTerms`, but the account must be setup correctly with DX to use the IncoTerms specified. Contact DX Customs team for more information
CustomsDeclarationRequest	CustomsDeclarationRequest[]	Yes	Detailed lines of commodity information
DxDocumentNumbers	String[]	No	The document number(s) returned by DX when documents were uploaded, to be associated with this consignment

FullCustomsResponse

The FullCustomsResponse type contains the same fields as the FullCustomsRequest above, but the CustomsDeclarationRequest field is renamed CustomsDeclarationResponse and is of type CustomsDeclarationResponse[], as well as one additional field:

FullCustomsResponse additional properties (in addition to FullCustomsRequest)			
Property	**Type**	**Required**	**Contains**
DxCustomsNo	String(20)	No	Assigned DX customs identifier

CustomsDeclarationRequest

Supplied in a FullCustomsRequest to detail each commodity present in the consignment.

CustomsDeclarationRequest type			
Property	**Type**	**Required**	**Contains**
Quantity	Integer	Yes	Quantity of this commodity (number of units). This is not the same as the number of items/packages in the consignment as more than one unit of the commodity could be in each package
Contents	String(200)	Yes	A detailed and accurate description of the commodity. Vague descriptions are more likely to be delayed clearing customs
OriginCountryCode	String(2)	Yes	ISO 3166-1 alpha-2 country code of the country where the commodity was manufactured or produced. The full list can be obtained by calling `Options/GetCountryCodes`
CommodityCode	String(10)	Yes	Commodity code. This should be declared based on the destination customs territory i.e.: • into EU; EU TARIC code • into the UK; UK Trade Tariff code The code is numeric with leading zeros. Do not include any spaces
Weight	Decimal	Yes	Weight of the commodity in kilograms (kg). Can be provided to 3 decimal places (i.e. 1 gram accuracy)
Value	Decimal	Yes	Commercial value of the commodity, or if FOC (free of charge) then the cost price, declared in the currency as supplied below

ValueCurrency	String(3)	Yes	The ISO 4217 currency code that the Value property represents:
			EUR = Euro
			GBP = Pound sterling

CustomsDeclarationResponse

The CustomsDeclarationResponse type contains the same fields as the CustomsDeclarationRequest above with one additional field:

CustomsDeclarationResponse additional properties (in addition to CustomsDeclarationRequest)			
Property	**Type**	**Required**	**Contains**
OriginCountryName	String(255)	Yes	The name of the country that was declared as the commodity's origin

NICustomsRequest

Required for Northern Ireland, under the Windsor Framework to supply the additional information required.

NICustomsRequest type			
Property	**Type**	**Required**	**Contains**
MovementType	String(3)	Yes	Enumeration value for the transaction / movement type: 1 = B2B 2 = B2C
RecipientEoriCode	String(20)	Yes if B2B	The EORI (Economic Operators Registration and Identification) number for the organisation or commercial recipient. Not required for (B2C) consumer movements
NIDeclarationRequest	NIDeclarationRequest[]	Yes	Detailed lines of commodity information
UKInternalMarketCompliant	String(20)	Yes if B2B	Confirmation that the goods declared conform to the UK Internal Market Scheme regulations and are being declared as ‘Not at risk’ under that scheme: CONFIRMED = Fully compliant

NICustomsResponse

The NICustomsResponse type contains the same fields as the NICustomsRequest above, but the NIDeclarationRequest field is renamed NIDeclarationResponse and is of type NIDeclarationResponse[], as well as one additional field:

NICustomsResponse additional properties (in addition to NICustomsRequest)			
Property	**Type**	**Required**	**Contains**
DxCustomsNo	String(20)	No	Assigned DX customs identifier

NIDeclarationRequest

Supplied in a NICustomsRequest to detail each commodity present in the consignment.

NIDeclarationRequest type			
Property	**Type**	**Required**	**Contains**
Quantity	Integer	Yes	Quantity of this commodity (number of units). This is not the same as the number of items/packages in the consignment as more than one unit of the commodity could be in each package
Contents	String(200)	Yes	A detailed and accurate description of the commodity. Vague descriptions are more likely to be delayed clearing customs
OriginCountryCode	String(2)	Yes if B2B	ISO 3166-1 alpha-2 country code of the country where the commodity was manufactured or produced. The full list can be obtained from `Options/GetCountryCodes`. Not required for (B2C) consumer movements
CommodityCode	String(10)	No	Commodity code. This should be declared based on UK Trade Tariff. The code is numeric with leading zeros. Do not include any spaces. **N.B.** DX recommends always providing a valid commodity code to prevent any delays
Weight	Decimal	Yes	Weight of the commodity in kilograms (kg). Can be provided to 3 decimal places (i.e. 1 gram accuracy). DX will use this information to determine the total weight. B2C movements cannot contain a single commodity of over 100kg, or a total weight over 31.5kg if containing multiple commodities
Value	Decimal	Yes	Commercial value of the commodity, or if FOC (free of charge) then the cost price, declared in GBP (£). DX will use this information to determine the total value

NIDeclarationResponse

The NIDeclarationResponse type contains the same fields as the NIDeclarationRequest above with one additional field:

NIDeclarationResponse additional properties (in addition to NIDeclarationRequest)			
Property	**Type**	**Required**	**Contains**
OriginCountryName	String(255)	Yes (if B2B)	The name of the country that was declared as the commodity's origin

15 API tracking reference

This section lists all the API methods related to tracking, with notes on the use of each method. Shared schema types are listed first and referenced from the methods that require them. Shared schema types already listed in the order reference section are not repeated, please refer to section 14.1 as required.

15.1 Shared tracking schema types

AddressSummary

Short form of a delivery or collection address for a consignment.

AddressSummary type			
Property	**Type**	**Required**	**Use**
Company	String(40)	Yes, if no recipient name	The name of the company for a business address
Recipient	String(30)	Yes, if no company name	The name of the recipient or contact for this address
Postcode	String(8)	Yes, if not IE	The postcode or Eircode that was provided

LatLong

GPS coordinates expressed as decimal degrees.

LatLong type			
Property	**Type**	**Required**	**Use**
Lat	Decimal	Yes	The latitude (northing) in decimal degrees from the equator, positive for degrees North, negative for degrees South. Expected range +49.0 to +61.0 for the British Isles.
Long	Decimal	Yes	The longitude (easting) in decimal degrees from the Prime Meridian, positive for degrees East, negative for degrees West. Expected range -11.0 to +2.0 for the British Isles.

ItemSummary

Summary of details of an item.

ItemSummary type			
Property	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	Item tracking number
Reference	String(30)	No	Your reference number for this consignment / order number, as provided when it was created
DeliverTo	AddressSummary	Yes	Delivery address
CollectFrom	AddressSummary	No	Collection address
Service	Service	Yes	Service type describing service attributes including the service level the consignment was booked under
WeightKg	Number	Yes	The weight of the item
Cancelled	Boolean	Yes	True if this item has been requested to be cancelled
Printed	Boolean	Yes	True if this item has had a label printed for it

TrackingAttachmentLink

Attachments linked to an event.

TrackingAttachmentLink type			
Property	**Type**	**Required**	**Use**
AttachedmentID	Int64	Yes	The ID of the attachment associated with a tracking event. Supply this to `Tracking/GetAttachment` to request the attachment
Type	String	Yes	The type of attachment: • Photo • Signature

TrackingEvent

Details of a tracking event.

TrackingEvent type			
Property	**Type**	**Required**	**Use**
TrackingDate	Datetime	Yes	The date and time the event occurred (local time)
StatusText	String(500)	Yes	A description of the event suitable for the sender or recipient, including the ETA if available.
DateForDelivery	Datetime	Yes	Current expected delivery date based on SLA. Ignore the time part of the date time value.
RedeliveryAllowed	Boolean	Yes	True if a redelivery can be organised (via my.dxdelivery.com)
EventCode	String	No	The internal status code for this event, ***legacy support only***
EventGroup	String	No	Event group, one of: - Attempted - Collected - Delivered - Pending - Problem
EventDescription	String	No	An internal description of the event code, not suitable for the recipient, ***legacy support only***
TrackingAttachmentLinks	TrackingAttachmentLink[]	No	Attachments available for this event
Location	LatLong	No	GPS coordinates recorded by the courier handheld if available

TrackingNumberTrackingResponse

Tracking events for each item (tracking number) found.

TrackingNumberTrackingResponse type			
Property	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	Item tracking number
TrackingEvents	TrackingEvent[]	Yes	The tracking events available

TrackingUpdate

The TrackingUpdate type contains the same fields as the TrackingEvent above, but with additional fields to associate each event with a tracked item and to control requesting more updates:

TrackingUpdate type additional properties (in addition to TrackingEvent)			
Property	**Type**	**Required**	**Contains**
TrackingID	String(50)	Yes	Sequential ID of this tracking event. Use this value to request subsequent updates following this event. See Key concepts section 10.8 on Tracking.
TrackingNumber	String(20)	Yes	Item tracking number

15.2 Tracking controller

GetConsignment

Returns the details entered and tracking events for a consignment including **all** the items in that consignment.

Method			
/Tracking/GetConsignment			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	The tracking number of any of the items manifested in the consignment

/Tracking/GetConsignment Response		
Name	**Type**	**Contains**
Consignment	ConsignmentResponse	The details of the consignment, including the details of **all** the items in the consignment
TrackingNumberTrackingResponses	TrackingNumberTrackingResponse[]	Tracking number and tracking events for each item in the consignment

GetItem

Returns tracking events for a single item.

Method			
/Tracking/GetItem			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
TrackingNumber	String(20)	Yes	The item tracking number to return tracking events for

/Tracking/GetItem Response		
Name	**Type**	**Contains**
TrackingNumberTrackingResponses	TrackingNumberTrackingResponse[]	Tracking events for the item requested

Search

Returns tracking item summaries for all items that match the SearchQuery requested. This would usually be used to support an interactive user searching based on the detail they have available. Explicit tracking requests on a systematic basis would use one of the other methods available based on tracking number.

Method			
/Tracking/Search			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
SearchQuery	String(200)	Yes	Query can contain one or more of: - Tracking number - Reference From the delivery or collection address: - Recipient - Company - Postcode - Full address The query can also contain partial information from the above fields as well. Any items that match any part of the data provided will be returned.

/Tracking/Search Response		
Name	**Type**	**Contains**
TrackingItemSummaries	ItemSummary[]	Summary details of items that matched any of the supplied query values.

GetAttachment

Returns all attachments for the tracking number, including signatures and photographs.

> ⚠ **Attachment data protection**
> These attachments are not screened by DX and should not be made available to the consumer. Do not download the attachment unless necessary as this is DX data. They can be downloaded for the purposes of viewing within a sending organisation but are not to be distributed to any third parties without legal agreement from DX.

Method			
/Tracking/GetAttachment			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
AttachmentID	Int64	Yes	The ID returned in TrackingAttachmentLink, of the attachment associated with a tracking event.

/Tracking/GetAttachment Response			
Property	**Type**	**Required**	**Contains**
TrackingNumber	String(20)	Yes	Item tracking number
AttachmentKey	String(50)	Yes	*Reserved for future use*
Type	String	Yes	The type of attachment: • Photo • Signature
Data	String	Yes	Base64 encoded binary image data

GetUpdates

Returns tracking events that have occurred since the FromTrackingID supplied. Each tracking update retuned contains a unique TrackingID that can be used to request more updates.

See Key concepts section 10.8 for a more detailed explanation of how to use this method.

Method			
/Tracking/GetUpdates			
Header parameter			
Name	**Type**	**Required**	**Use**
Token	String(50)	Yes	A valid `AccessToken` is required from a call to `CreateToken`
Body contents			
Name	**Type**	**Required**	**Use**
AccountCode	String(20)	Yes	The account code that events will be returned for
FromTrackingID	String(50)	No	Sequential ID of the last tracking event. Or NULL if no tracking events have been downloaded.
MaximumResults	Integer	No	The maximum number of updates to return. Valid values are between 100 and 1,000. If not provided, then up to 1,000 updates will be returned.

/Tracking/GetUpdates Response		
Name	**Type**	**Contains**
TrackingEvents	TrackingUpdate[]	New Tracking updates (if available) based on the AccountCode and FromTrackingID supplied. If no new updates are available, then an empty array will be returned.

16 Integration artifacts

Integration artifacts are provided by DX to support development and integration.

Unless required by applicable law or agreed to in writing, software artifacts are supplied on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

DX reserves all rights in its Confidential Information. The disclosure of Confidential Information by one party does not give the other party or any other person any licence or other right in respect of any Confidential Information beyond the rights expressly set out in this agreement.

You may use the supplied artifacts to:

- design and develop an integration to the DX Despatch Manager API
- create derivative software works that enable you or your customers to integrate to DX
- document and maintain your integration with DX

You may **not**:

- copy, modify, sell, advertise, display, perform, make available, or otherwise transfer or license these artifacts to any other party that are not part of, contracted to, or directly employed by your organisation
- use the trade names, trademarks, service marks, or product names of DX, except as required for reasonable use in describing the services DX offers, or unless with the express permission of DX
- apply for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of using of the artifacts provided

16.1 OpenAPI specification

Please contact the DX Customer Integration Team to obtain a copy of the OpenAPI specification.

16.2 Postman collection

Please contact the DX Customer Integration Team to obtain a copy of the sample Postman collection.

Appendix 1 - API error messages

The following table details many of the API error messages that can be returned, with some more detail on the potential reason for the error being returned.

If you receive an error that is not in the list, and you need further guidance on how to handle it please contact the DX Customer Integration Team.

Error message	Status	Description
Account	401	The account that this entity relates to is not accessible based on the token provided
AccountCode	401	The account code is not accessible based on the token provided
AccountCode	400	The account code was not provided
AccountCode not allowed to create consignment. Account is OnStop	422	The account has been placed on stop and is not allowed to create consignments
AccountCode not allowed to create consignment. No ExpressBillingAccountNo	422	A parent account code that does not have a billing account associated with it cannot be used to create a consignment
AccountCode not allowed to print. Account is OnStop	422	The account has been placed on stop and is not allowed to print any labels
Collection address cannot be specified on a delivery service	422	A collection address was provided when the service does not allow it to be specified. The API will not accept information that would be ignored
CollectionAddress.County is not accepted for GB	422	County is not accepted on GB addresses
CollectionAddress.Recipient or CollectionAddress.Company required	400	One or both of the recipient and company fields must be provided
CollectionCountryCode required	400	The country code must be provided for the collection address
CollectionPostcode required for CollectionCountryCode of GB	400	A postcode must be provided for addresses in the United Kingdom and Crown Dependencies (GB country code)
Consignment DeclarationType: <Code> does not need documents	422	Attempting to attach a document to a customs declaration that does not require documents is not allowed; the declaration type will be returned in the error message
Consignment not International	422	Attempting to attach a document to a consignment that does not have any customs declaration is not allowed
Controlled goods not allowed	400	DX does not permit controlled goods to be carried as the detailed customs procedures that must be followed are not supported
CoverRequired must be true if supplying a CoverValue	400	Mismatch between cover required and cover value properties
CoverValue must be greater than zero if CoverRequired is true	400	Mismatch between cover required and cover value properties
Current Origin to Destination not possible	422	Due to customs requirements that are not supported or cannot be fulfilled by the service chosen, the collection and delivery addresses are not allowed

CustomsDetail supplied but not required	400	A customs detail object was included in the request, but the current customs request does not require this (e.g. NotRequired, Consumer). The API will not accept information that would be ignored
DeliveryAddress does not match return address	422	The delivery address does not match the return address based on the key provided, the fields of the address (Address1-3, Town, County, Postcode) must match
DeliveryAddress.County is not accepted for GB	422	County is not accepted on GB addresses
DeliveryAddress.Recipient or DeliveryAddress.Company required	400	One or both of the recipient and company fields must be provided
DeliveryCountryCode required	400	The country code must be provided for the delivery address
DeliveryPostcode required for DeliveryCountryCode of GB	400	A postcode must be provided for addresses in the United Kingdom and Crown Dependencies (GB country code)
DespatchDate can't be more than <n> days in future	422	The despatch date cannot be over n days in the future; the actual number of days applicable will be returned in the error message
DespatchDate is before Cut Off Day	422	The despatch date cannot be before the number of cut off days configured; this is usually 0 (i.e. not before today)
DocumentData exceeds maximum size	422	The size of the document exceeds the allowed limit
Documents not allowed for this type of declaration	400	This type of customs request object does not support documentation being added
DxDocumentNumber cannot be null or empty	400	A document entity was sent in the request, but no document number was provided
For cancellation the collection date must be in the future	400	A cancellation request for a collection can only be made before the requested date of collection
Invalid Account	400	The account code was not valid
Invalid AddDocumentRequest. DocumentType <Code>	400	The document type provided was not recognised as a valid document type enumeration value; the document type will be returned in the error message
Invalid CollectionAddress.CountryCode <Code>	400	The country code provided on the collection address was not valid; the country code will be returned in the error message
Invalid CustomsDeclarations. OriginCountryCode <Code>	400	The origin country code provided on the customs declaration was not valid; the country code will be returned in the error message
Invalid DeliveryAddress.CountryCode <Code>	400	The country code provided on the delivery address was not valid; the country code will be returned in the error message
Invalid DxDocumentNumber <n>	422	The document number provided was not recognised or accessible based on the token provided; the document number will be returned in the error message
Invalid file extension	422	The filename extension of the document is not allowed

Invalid Package type <Code>	400	The package type provided was not valid; the package type will be returned in the error message
Invalid RefreshToken	401	The refresh token provided is not valid (it may have expired)
Invalid search query	400	Search query was either empty, too long or too short
Invalid UserCredentials	401	The credentials provided were not valid, check the credentials are correct and active
Item not found	400	The item requested to be cancelled cannot be found
Label printing is not allowed for this consignment	422	The service type of the consignment does not allow label printing (e.g. Transfer and Return)
Maximum Quantity of Items exceeded. Max Quantity of Items <n>	422	The number of items in the request exceeds the maximum quantity allowed. This is to protect customers from making large requests accidentally. The limits can be increased if genuinely required, or simply manifest large volumes in batches. The maximum allowed quantity will be returned in the error message
Maximum Weight exceeded. Max Amount of item <n>	422	The total weight of all items in the request exceeds the maximum consignment weight allowed. This is to protect customers from making large requests accidentally. The limits can be increased if genuinely required, or simply manifest heaver weights in smaller quantities. The maximum allowed consignment weight will be returned in the error message
No return address found for the ReturnAddressKey	400	The return address key provided was not recognised
No service available for given postcode	422	The postcode was not recognised, or is not available for the requested service
No service available for given postcode	422	No service is available based on the requested delivery address (or collection address if provided)
Password	400	The password was not provided
Past cut off for despatching that day. Cut off: <HH:mm:ss>	422	The despatch cut off has passed, the actual cut off time applicable is returned in the error message in the format HH:mm:ss e.g. 16:00:00
RefreshToken	400	The refresh token was not provided
Retroactive cancellation is not allowed	400	A cancelation request for a delivery can only be made on or before the despatch date
ReturnAddressKey required for Return service	400	Return service requested without providing the return address key
Service has DeclarationType of NotAllowed	422	The movement is not allowed based on the collection and/or delivery address specified
Service not available for given postcode	422	The service requested is not available to the requested delivery address (or from the collection address if provided)
ServiceCode <Code> does not allow CoverRequired	422	The service requested does not allow Transit Liability cover to be added; the service code will be returned in the error message
ServiceCode <Code> has a maximum CoverValue of <n>		The service requested does not allow that amount of Transit Liability cover to be added; the service code and the maximum liability allowed will be returned in the error message

The manifest date range must be a single day	400	Currently only a single day's manifest can be requested
This user type is not authorized to use this interface	403	Attempting to authenticate to the API using a user's credentials (only API credentials can be used)
Unable to find Origin Zone for Account	422	The collection location was not found for the account code provided. Possible DX configuration error
Unable to find TrackingNumber	422	The tracking number provided was not found, or it is not accessible based on the token provided
Unable to print, all items have been cancelled	422	All the items in the consignment have been cancelled
Unable to print, item has been cancelled	422	The item requested has been cancelled
Unable to print, item has been cancelled. Item was reprinted and replaced by <n>	422	The item requested has already been reprinted; the new tracking number will be returned in the error message
Unable to print, item was changed during processing	422	Possibly caused by a race condition between parallel requests acting on the same consignment or item
Unable to route Collection. Collection Postcode <Postcode>, Collection Country <Code>	422	The collection location was not recognised based on the postcode and country code supplied in the collection address; the postcode and country code will be returned in the error message
Unable to route Delivery. Delivery Postcode <Postcode>, Delivery Country <Code>	422	The delivery location was not recognised based on the postcode and country code supplied in the delivery address; the postcode and country code will be returned in the error message
Username	400	The username was not provided
You must supply a Customs Request object	400	A customs request is required based on the collection and/or delivery address specified
You must supply a Customs Request of type <Code>	400	A customs request is required based on the collection and/or delivery address specified; the customs request type required will be returned in the error message