



Royal Mail Group

Shipping API V2 (SOAP)

Technical User Guide

This API specification details the requirements for integrating with Shipping API V2 (SOAP). It specifically covers how the Shipping API V2 can be used by business customers to conduct shipping activity with Royal Mail and provides the technical information to build this integration. This specification must be used with the relevant accompanying specifications for customers wishing to interface their systems with Royal Mail services.

Contents

1	Document Control.....	4
1.1	Terms and Abbreviations	4
1.2	Version History	4
2	Overview	6
3	Purpose.....	7
4	Introduction to Shipping API V2.....	8
4.1	Overview	8
4.2	Interface Components	8
5	Integrating with the Shipping API V2	9
5.1	Terms & Conditions	10
5.2	API Access	10
5.3	Live Deployment.....	11
5.4	API Versioning.....	11
6	Shipping Services	12
6.1	Business Services	12
6.2	Standard process/API flow	13
6.3	Offline Barcoding.....	14
7	Message Structure.....	16
8	Message Definition and Schemas	17
8.1	Notation.....	17
8.2	WSDL and Schemas.....	18
8.3	HTTP Header Information	19
8.3.1	Description.....	19
8.3.2	Request Message.....	19
8.3.3	Example Data.....	20
8.4	Security Credentials	20
8.4.1	Description.....	20
8.4.2	wsse:Security Element	20
8.4.3	Example Security data	22
8.4.4	References.....	22
8.5	createShipmentRequest element	22
8.5.1	integrationHeader element	23
8.5.2	requestedShipment element	24
8.5.3	Example Data.....	37
8.6	createShipmentResponse.....	39
8.6.1	integrationHeader.....	40
8.6.2	completedShipmentInfo element	41
8.6.2.1	shipment element.....	42
8.6.3	integrationFooter.....	43
8.6.4	Example Data.....	45
8.7	updateShipmentRequest.....	47
8.7.1	Example Data.....	48
8.8	updateShipmentResponse.....	49
8.8.1	Example Data.....	50

8.9	cancelShipmentRequest	51
8.9.1	Example Data.....	52
8.10	cancelShipmentResponse	53
8.10.1	Example Data.....	54
8.11	printLabelRequest.....	54
8.11.1	Example Data.....	56
8.12	printLabelResponse	57
8.12.1	Example Data.....	60
8.13	createManifestRequest.....	62
8.13.1	Example Data.....	63
8.14	createManifestResponse	63
8.14.1	Example Data.....	64
8.15	printManifestRequest.....	65
8.15.1	Example Data.....	66
8.16	printManifestResponse	66
8.16.1	Example Data.....	67
8.17	printDocumentRequest	67
8.17.1	Example Data.....	68
8.18	printDocumentResponse.....	68
8.18.1	Example Data.....	69
8.19	request1DRangesRequest	70
8.19.1	Example Data.....	70
8.20	request1DRangesResponse.....	71
8.20.1	Example Data.....	72
8.21	request2DItemIDRange	73
8.21.1	Example Data.....	73
8.22	request2DItemIDRangeResponse.....	74
8.22.1	Example Data.....	74
9	Error Handling.....	75
9.1	Overview	75
9.2	Technical Errors	75
9.3	Business Errors.....	76
9.4	Warnings	77
10	Non-Functional Characteristics	78
10.1	Availability	78
10.1.1	Service Hours.....	78
10.1.2	Maintenance Windows.....	78
10.1.3	Unavailability.....	78
10.2	Performance.....	78
10.3	Security	78
11	Frequently Asked Questions.....	79
12	Appendix A – Reference Data.....	80
12.1	Allowable Character Set.....	80
12.2	Shipment Status Codes	81
12.3	XML Examples.....	81

1 Document Control

1.1 Terms and Abbreviations

Term	Meaning
Allocated	Shipment with a Service Type / Service / Service Format and shipment number but not printed
Base64	A standard binary-to-text encoding scheme that is used to represent binary data in an ASCII string format. Used to include binary data with an XML structure
DMO	Despatch Manager Online
HTTPS	Hypertext Transfer Protocol over SSL
IP	Internet Protocol
Local Collect	A Service Enhancement where delivery is made to an agreed alternate location
Manifested	Customer Collection Receipt has been created and Customer Collection Receipt has been printed
Nonce	User generated random string that is used as part of the password hashing process to ensure that messages cannot be replayed
OBA	Online Business Account
Printed	Shipment with Service Type / Service / Service Format and shipment number and the label(s) printed
SOAP	Originally an acronym for Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of web services
WSDL	Web Service Description Language
XSD	XSD is XML Schema Definition – this defines a specific template for the XML used by particular systems
XML	eXtensible Markup Language, a flexible standard for data inside a hierarchical structure of named data items

Table 1 – Terms and Abbreviations

1.2 Version History

Version	Date	Notes
2.0	12/02/2015	Version based on Version 1 SAPI, incorporating version 2 capability
2.01	16/02/2015	Added Example Data for each operation.
2.02	23/02/2015	Updated from feedback

Version	Date	Notes
2.1	24/06/2015	Updated- Safeplace corrected to 24 characters for PDF labels. More detail on Localised addresses
2.2	14/11/2015	Document updated following introduction of API Management capability.
2.3	25/11/2015	Document updated following process review of API Management capability
2.4	04/12/2015	'Accept' parameter added to HTTP Header in section 8.3. Minor updates made to assist with readability.
2.5	18/02/2016	Document updated following feedback.
2.6	06/10/2016	Includes Offline barcodes, XML examples and API operational flows, Optional End of Day Average Weight function for Average Weight Products only

Table 2 – Document Version History

2 Overview

Royal Mail Shipping API V2 exposes a web service that allows account customers to create shipments, produce labels, and produce documentation for all the tasks required to ship items with Royal Mail. Built on industry standards, Shipping API V2 provides a simple and low cost method for customers to integrate with Royal Mail, and allows them to get shipping quickly.

There are no costs to customers for using the Shipping API V2 services, however customers' own development costs must be covered by the customer developing the solution. Royal Mail will not accept any responsibility for these development, implementation and testing costs.

Customers should address initial enquiries regarding development of systems for these purposes to their account handler.

3 Purpose

This document is to provide Royal Mail's customers with guidelines and detailed specifications for integrating with the Shipping API V2 SOAP web service.

The document details:

- The specification for the web service interface
- Description of errors the API can return
- Non-functional characteristics of the API including response times, service availability and security considerations.

This document is primarily intended to be read by developers and other technical roles involved with integrating customer systems' with Shipping API. This document should be read in conjunction with the following artefacts which are available from the Shipping API V2 page on the [Royal Mail API \(Developer\) Portal \(https://developer.royalmail.net\)](https://developer.royalmail.net):

- Shipping API V2 WSDL
- Shipping API V2 XSDs
- Shipping API V2 Reference Data
- Shipping API V2 FAQ

The Shipping API web service operations which are defined in this document are:

- createShipment
- updateShipment
- cancelShipment
- printLabel
- createManifest
- printManifest
- printDocument
- request1DRangesRequest
- request2DItemIDRangeRequest

For details of how to set up GUI user IDs and their viewing permissions, as well as creating reports, see the DMO User Guides, available from the Shipping API V2 page on the [Royal Mail API \(Developer\) Portal](https://developer.royalmail.net). This set up is needed for both the Onboarding and Production Environments.

4 Introduction to Shipping API V2

4.1 Overview

Shipping API V2 provides the functionality for customers to take a shipping transaction from creation to collection.

In simplest terms, the logical flow is as follows:

- **Create Shipment** – the details of an item are provided to Royal Mail, and a shipment is created on the system. The status of the shipment is 'Allocated'.
- **Print Label** – once a shipment has been created, the label for it can be printed. Once printed, the status of the shipment is updated to 'Printed'.
- **Print Document(s)** – once a Non-European Union international shipment has been created, international documents can be printed. The printDocument request will validate you have the pre-requisite data and return the PDF documents for CN22, CN23 and/or Commercial Invoice (CI).
- **Manifest Shipments** – before items are collected, the customer must submit details of all the items to Royal Mail and print off the Customer Collection Receipt for the driver. The createManifest call submits the details for all shipments that are in the 'Printed' state to Royal Mail (those that are in the 'Allocated' state are ignored). The status of these shipments is then set to 'Manifested', and they can no longer be updated or cancelled.
- **Range Requests** – it is possible for the Shipping API platform to assign a range of 1D Linear Barcodes (for specified services) and 2D Item IDs for use offline.

Unless you have been granted an exemption, your shipments will be subject to a clean sweep process. This is a process that runs at a specific time each night, and automatically manifests any 'Printed' shipments that have not already been manifested.

Section 6 provides further detailed information especially about the different process/API flows for the business services used by you.

4.2 Interface Components

Please see Figure 1 below for a graphical representation of the interface between Royal Mail and you for Shipping API V2. This document covers what information is to be

exchanged, how this information is structured and the means by which it is transferred.

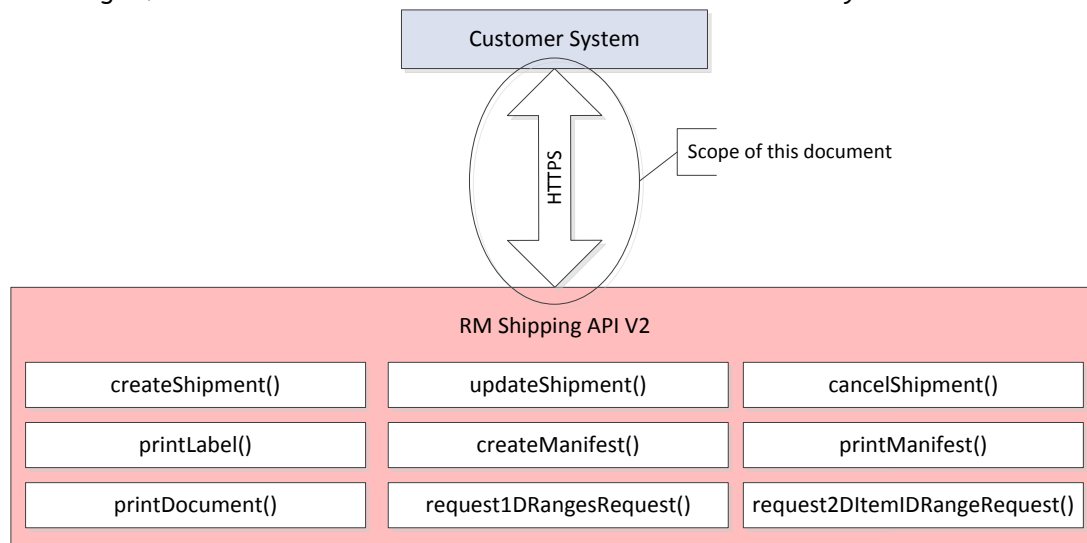


Figure 1 – Shipping API V2

5 Integrating with the Shipping API V2

The high-level process associated with integrating with Shipping API V2 is represented in the diagram below and described in the sections which follow.

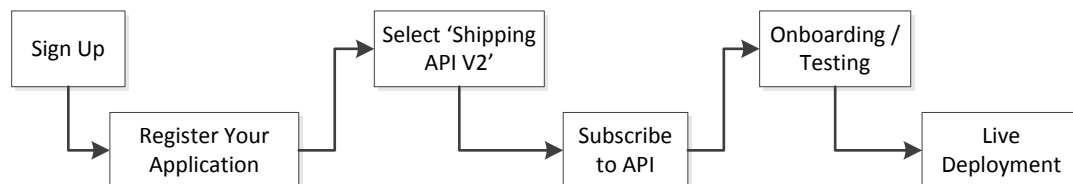


Figure 2 – Process for Integrating with the API

Access to the service is managed through RMG's API Management system.

New users of the system will need to:

1. Sign up for an account and accept the terms and conditions on the [Royal Mail API \(Developer\) Portal \(https://developer.royalmail.net\)](https://developer.royalmail.net).
2. Register the customer side 'application' which will be calling the API. When the application is registered, it will be assigned a unique system-generated Client ID and Secret which is needed to securely access the API. It is important that these credentials are noted and securely stored.
3. Request to subscribe to the API. This will result in an e-mail being automatically generated and sent to the Royal Mail Customer Solutions team.
4. Once approved, testing can be performed against the API in a sandboxed onboarding environment that allows you to test the integration.

5. Once all required testing has completed in the onboarding environment, access to the Live production system will be provided at a mutually agreed date/time.

Existing users who already have an account with Royal Mail's API Management system will need to perform step 2 onwards if the application accessing the API is different to any currently registered applications. If the application accessing the API is already registered, existing customers will need to perform step 3 onwards.

5.1 Terms & Conditions

You must accept the Royal Mail Terms and Conditions when creating your customer account. These cover the ways in which the service may be used and any integration activities must abide by these.

Of particular note to developers:

- The onboarding environment may not be used for performance testing. This is a small scale system for functional testing only.
- Repeated reprints of labels or Customer Collection Receipts will be flagged to Royal Mail and may result in an investigation.
- Where specified, weights should be accurate. Discrepancies between reported and actual weights will be investigated by Royal Mail.
- All Royal Mail APIs impose a cap on the number of transactions for each customer. Excessive volumes of traffic within a short period will result in transactions being rejected.
- The Customer Data will be held by us on our servers for a limited period of time (currently thirteen months) to enable users to run activity reports. Thereafter, we may delete or remove such Customer Data without further notice.

5.2 API Access

Both onboarding and live access to the API is obtained via the following URL:

<https://api.royalmail.net/shipping/v2>

Please note that the Client ID and Secret must be provided in the HTTP header of all API requests otherwise access to the API will be rejected and a HTTP 401 (Unauthorised) will be returned. The Client ID and Secret are obtained by registering an application on the RMG API Management site.

You must complete all required test activities in the onboarding environment prior to being permitted access to the live environment by the Royal Mail Customer Solutions Team. Once you log in on the developer portal and have been authorized to use your subscribed APP, the onboarding environment is made available to you. This will allow you to test your APP integration without data being passed through to the Royal Mail operational and billing systems and without incurring any charges against your account.

You will be provided with a contact in Royal Mail who will take you through the onboarding process and provide you with the required security credentials to access the API. Once you have successfully demonstrated that your system works with ours, and that you can

produce labels to the required level of quality for all services on your account and you can successfully manifest your shipments, you will be granted access to the live system and can begin shipping items.

Please see section 9 for a full list of technical and business error codes which are returned from this API.

5.3 Live Deployment

Once you have completed all required testing in the onboarding environment you will be provided with access to the live production system.

If new products or services are added to your account, you may be asked to demonstrate that you have these working correctly in the onboarding environment before you are allowed to use them on the live system.

5.4 API Versioning

Royal Mail is continuously working to improve its technology, and as part of this process updates to the services provided may on occasion necessitate a new WSDL version. Royal Mail will look to maintain three versions of the WSDL; as new versions are introduced, previous versions move down the stack until they are ultimately removed completely:

- Latest version
- Previous version
- Deprecated version

You will always be encouraged to integrate against the latest version as this will give you the longest stable period without the need to change, but if you have already begun integration activities when a new version is released then you will be able to integrate against the previous version. You should not integrate against the deprecated version.

6 Shipping Services

6.1 Business Services

Shipping API V2 is a service offered to RM customers to request for the creation / update / cancellation of a shipment, printing of a label, printing of international documents, requesting barcode and Item ID ranges and creation/printing of a manifest.

Table 1 provides an overview of the business services that are supported by this interface.

Business Service	Web Service Operation	Description	Technology	Conversation Style
Create Shipment	createShipment	Creates a shipment on the system	SOAP over HTTPS	Synchronous Request / Response
Update Shipment	updateShipment	Updates the details of a shipment that has been created but not manifested	SOAP over HTTPS	Synchronous Request / Response
Cancel Shipment	cancelShipment	Cancels a shipment that has been created but not manifested	SOAP over HTTPS	Synchronous Request / Response
Print Label	printLabel	Prints a label for a shipment that has been created	SOAP over HTTPS	Synchronous Request / Response
Create Manifest	createManifest	Manifests current shipments	SOAP over HTTPS	Synchronous Request / Response
Print Manifest	printManifest	Provides a printable manifest in PDF format	SOAP over HTTPS	Synchronous Request / Response
Print Document	printDocument	Provide one or more printable international document in PDF format	SOAP over HTTPS	Synchronous Request / Response
Request 1D Ranges	request1DRanges	Provide a contiguous block of 1D Barcode identifiers for specified service(s)	SOAP over HTTPS	Synchronous Request / Response
Request 2D Range	request2DItemIDRange	Provide a contiguous block of 2D Item IDs for any service	SOAP over HTTPS	Synchronous Request / Response

Table 1 – Business Services

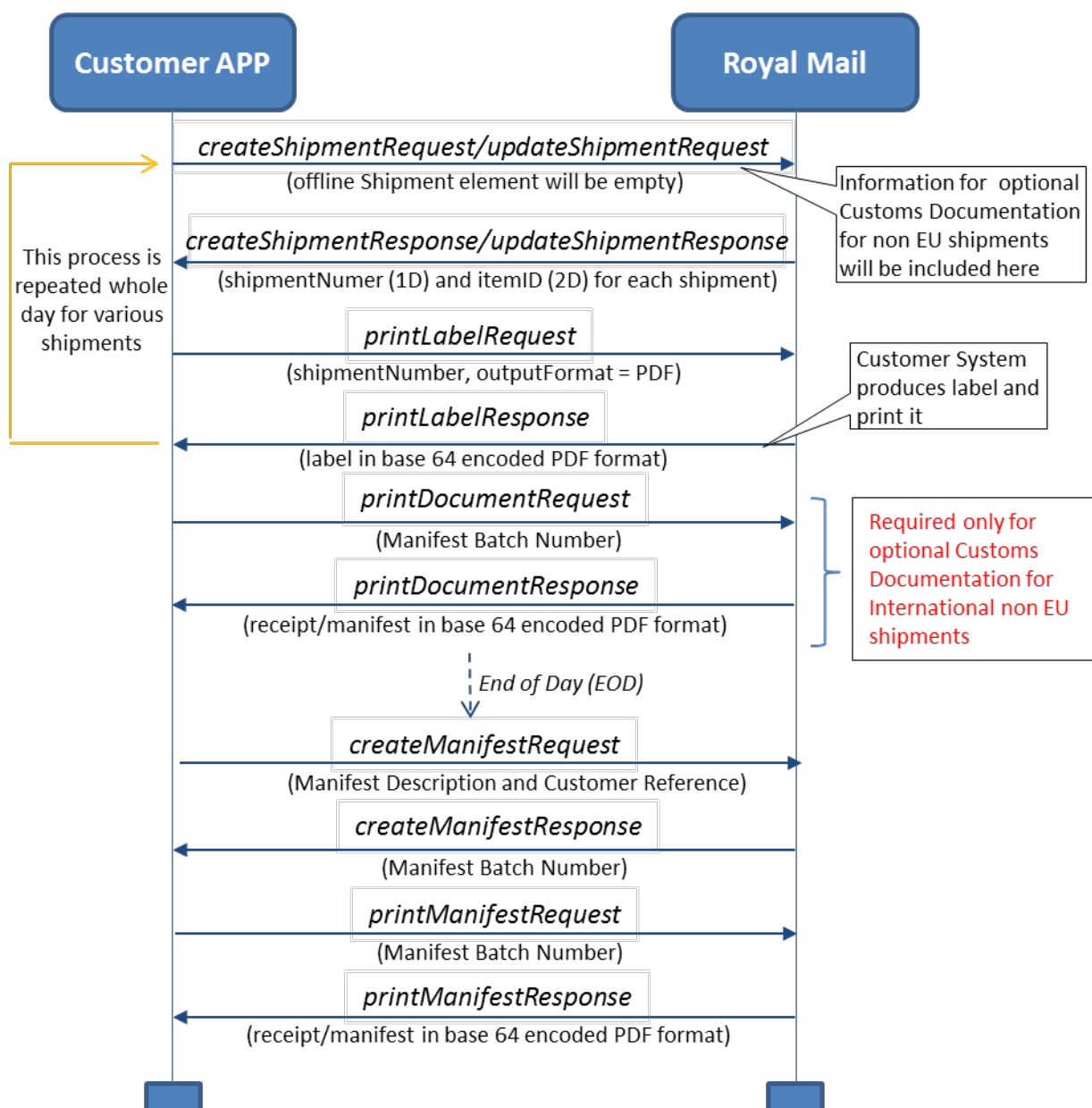
Based on the business service you are using, you have to go through different process/flows. These process/API flows related to each business service are explained below.

6.2 Standard process/API flow

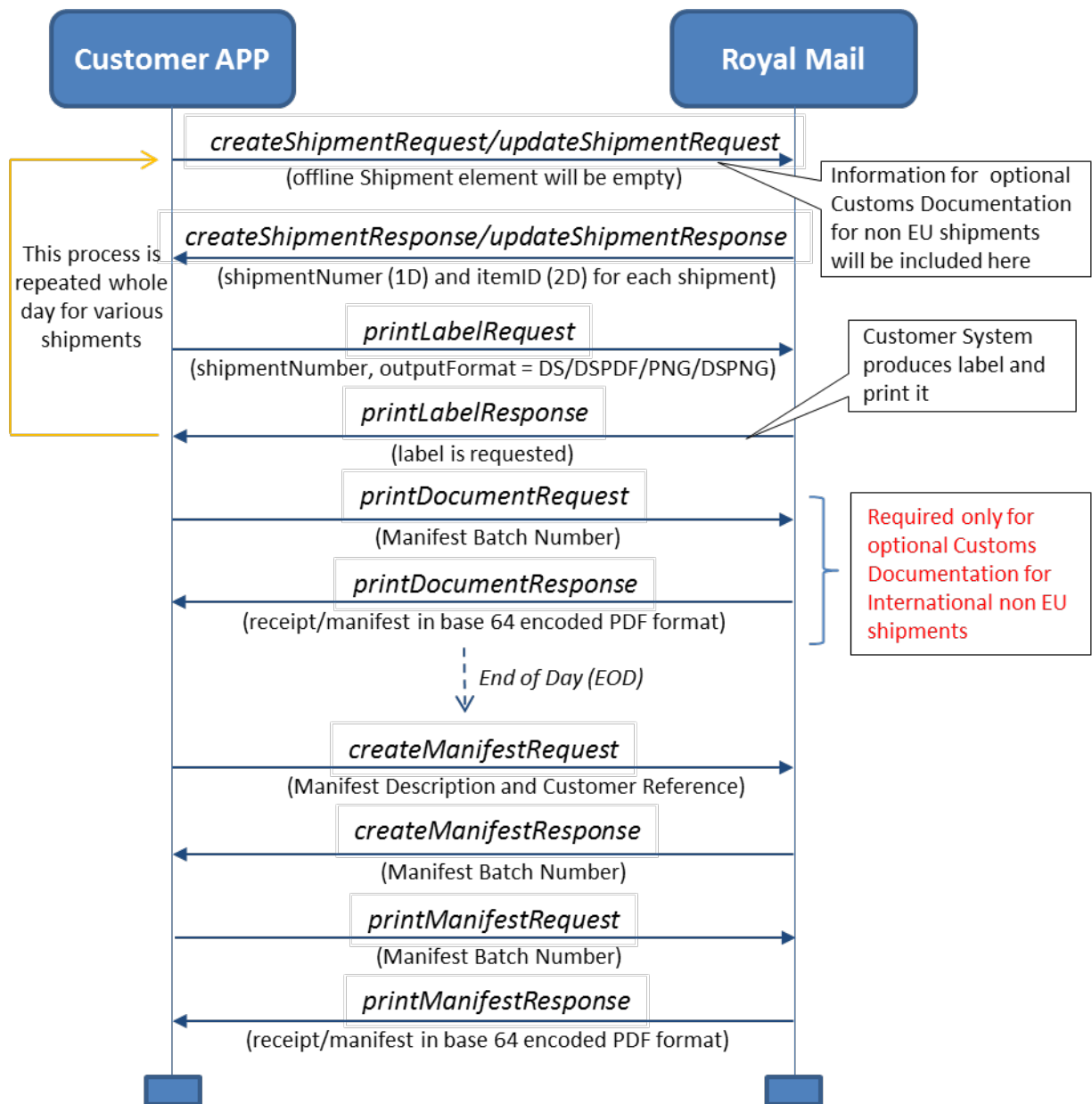
This standard process is relevant to you if you use normal API calls to create shipment, print the label and create manifest by EOD (End of Day). It should be noted that the API flow explained in this section doesn't apply if you use offline barcoding.

Below are some of the main flows which come under this category

createshipmentRequest(Domestic/International) with label in standard PDF format:



createshipmentRequest(Domestic/International) with Datastream:



Each of the operations mentioned in the flow diagrams above i.e. createshipmentRequest, printlabelRequest etc. are detailed further in sections below in this document.

NOTE: You should call printlabelRequest and printdocumentRequest (required only for optional Customs Documentation for international non-EU shipments) API for each shipment separately.

6.3 Offline Barcoding

Offline barcoding is the service provided by Royal Mail which allows you to print the labels offline without calling the APIs. If you have high volume of shipments then this will help you to ship the items faster as you will not be relying on multiple API calls.

You will use offline barcoding when you want to design produce the labels yourselves (e.g. for an integrated pick pack document and shipment label) and use the API to advise Royal Mail about the shipment. Offline Barcoding is typically used by a high volume customer as it allows you to produce the label faster, thus not relying on the createShipment and printLabel API calls.

Offline barcoding API allows you to get 1D or 2D ranges and then produce and print your own labels using these ranges. You can request the 1D or 2D ranges by calling the following APIs:

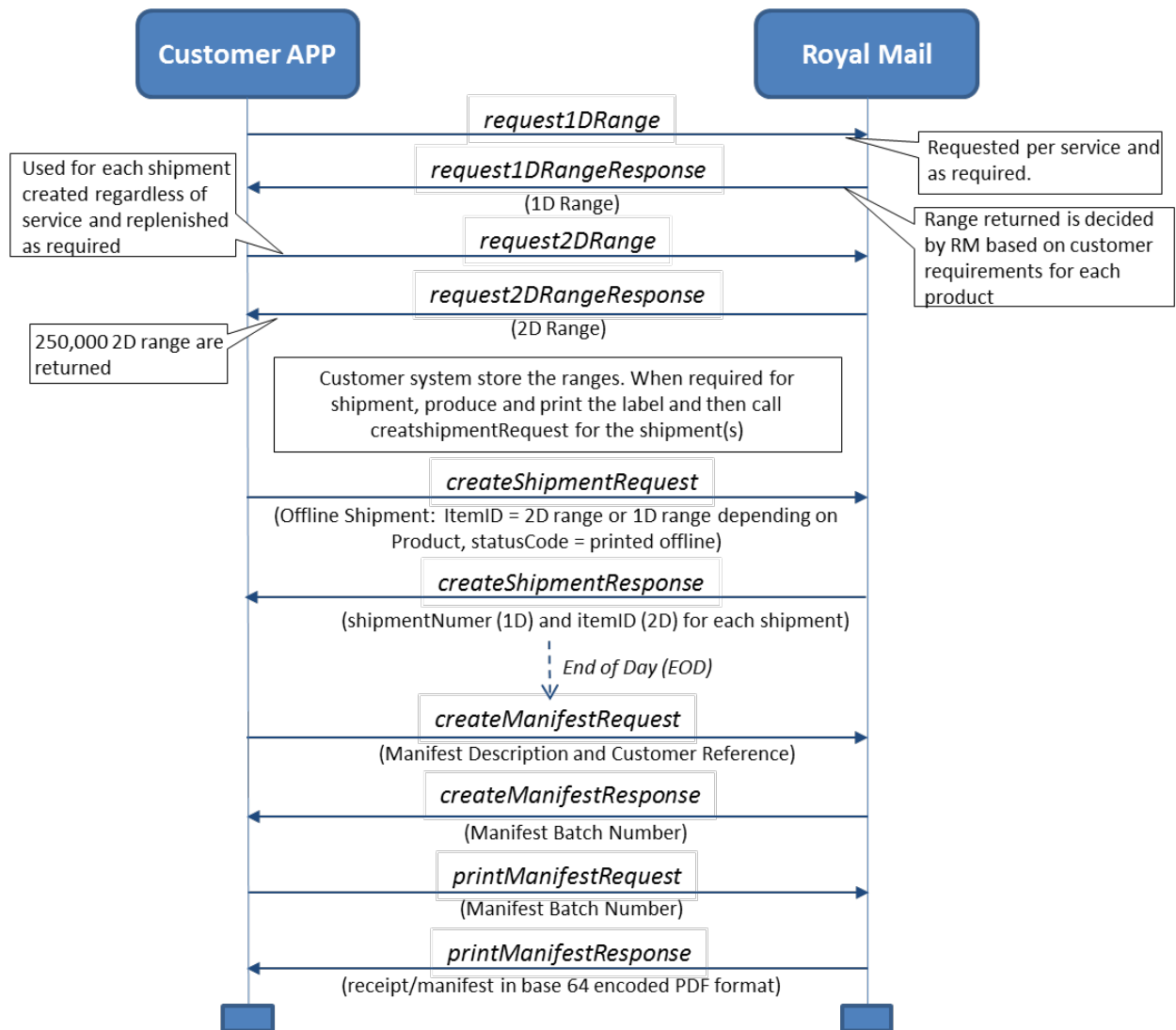
- **request1Dranges:** You can use this API to request for 1D barcode ranges to be used offline. On request the API will return 1D barcode containing the start and end barcode number. 1D barcode are used by Tracked, Tracked Returns, Special Delivery, Signed for, and International Tracked and/or Signed services. The size of the barcode range returned will depend on the volume of shipments set by Royal Mail for the customer by product.
- **request2Dranges:** You can use this API to request 2D barcode ranges (in Decimal, not Hexadecimal format) which can be used offline. This API will return a range of 250,000 unique 2D customer IDs in hexadecimal format.

You should store these 1D and 2D ranges in your system which are then allocated by your system to the shipment labels and prints them whenever there is a requirement for new shipment(s). You are provided a specification to generate the barcoding and tracking numbers included in the label. You can request the latest version of these specifications by contacting the Royal Mail Customer Solutions team.

You should then call the createShipmentRequest API including the fields or elements for offline barcoding. These elements will contain 2D item ID and if required for Tracked, Special Delivery, or Signed For services, the 1D tracking/shipment number allocated to the label for the shipment.

NOTE: You will not be able to request a new 1D and 2D range until you have used the allocated ranges. If you use offline barcoding you therefore must call the createShipmentRequest thus allowing RMG to track the used range. This process will help in the automated range replenishment which offline barcoding offers. In addition to this, you will need to contact the Royal Mail customer solutions consultant to configure access to offline barcoding for you and to provide you the Royal Mail labelling specifications.

The diagram below shows the API flow for offline barcoding customers.



Each of the operations mentioned above are explained further in later sections in this document.

7 Message Structure

The structure of the Shipping API V2 SOAP request and response messages is represented by the diagram below.

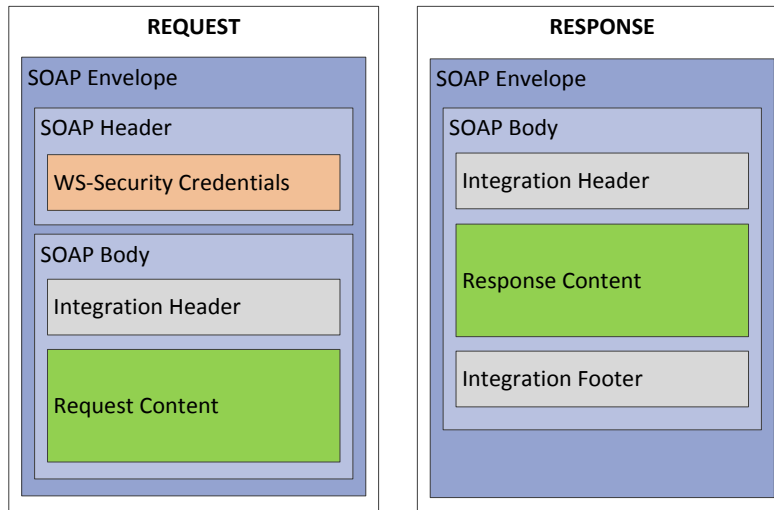


Figure 3 – API Structure

Each of the Shipping API V2 operations (see section 6.1) is defined as a SOAP operation, with separate request / response message parts. Technical details of these messages are documented in the WSDL and XSD schemas which accompany this document.

Within the SOAP body tags, each request or response message is supplemented with an integrationHeader element and is described in Sections 8.5.1 and 8.6.1. Response messages may contain an integrationFooter element that contains details of any errors or warnings and this is described in Section 8.6.3.

8 Message Definition and Schemas

8.1 Notation

The API elements described in this section are constructed using Royal Mail's Enterprise Canonical Data Model. The schema structures are described in XMLSpy notation with graphical representation meanings explained in the table below.

Graphical Representation	Meaning
	A solid line around an element indicates it is a mandatory field that will/must always be present.
	A dashed line around an element indicates the field is optional and may or may not be present. An optional element is one that indicates the data and enclosing XML tags may or may not be present in either a request or response.
	The <i>sequence</i> identifier represents an ordered set of elements.
	The <i>choice</i> identifier represents a selection of elements.

Figure 4 – Notation used to describe this API

Royal Mail's Enterprise Canonical Data Model contains optional fields not used by the Shipping API V2 web service. For example the standard Address element contains optional elements such as addressUsageType, stateOrProvince, and audit which can be considered optional and can be omitted from the message. If an element is not listed in the sections below, it is not used by the Shipping API V2 web service and data should not be placed in those elements. Any elements that are not used in the Shipping API V2 implementation are shaded in grey and indicated as such in the supporting text. Elements described in more detail later are shaded in purple.

In the context of the tables defined in this section:

- All data types are defined in the Data Types schema (see section 8.2).
- All 'Occurs' values signify whether each data entity is mandatory or optional in the context of Shipping API V2. Although an element may be defined as optional in the Canonical Data Model, it may still be considered as being mandatory for a Shipping API V2 request. Particular conditions under which an element needs to be populated are defined in the 'Description' column.

8.2 WSDL and Schemas

<p>The following table lists the documents that should be referenced for the RMG-defined canonical data types used by Shipping API V2. The XML schema documents themselves are provided in supporting files. Each schema file contains descriptions of every type using the "xs:documentation" element.</p> <p>WSDL / XML Schema File</p>	<p>Description</p>
ShippingAPI_V2_0_9.wsdl	WSDL describing the functionality offered by Royal Mail Shipping API V2.
ShippingAPI_V2_0_9.xsd	Royal Mail schema for the public API which is built from the RMG defined canonical data types. This file defines the actual request and response message structures.
oasis-200401-wss-wssecurity-secext-1.0.xsd	OASIS schema defining the standards for OASIS WS Security
oasis-200401-wss-wssecurity-utility-1.0.xsd	OASIS schema that supplements the OASIS WS Security schema
xml.xsd	Schema for the XML language, used to support the Oasis schema files.
xmldsig-core-schema.xsd	XML Signature schema used to support the Oasis WS Security standard.

CommonClassesV2_3.xsd	Royal Mail (RMG) schema which defines common objects such as “address” which are complex types with a defined structure based on RMG defined simple data types as well as reference data types.
DatatypesV2_3.xsd	RMG defined simple types that are commonly required such as “date” and “name”. This schema is the foundation for the other schema files.
CommonIntegrationSchemaV1_11_1.xsd	RMG schema which defines a common header and footer definition to be used for request and response messages.
ReferenceDataV2_3.xsd	RMG schema which defines a set of types for common reference data such as address fields that are defined using the RMG simple data types.
RMGSOAPExtensionsV1_2.xsd	RMG schema which extends the SOAP fault message construct and defined the RMG specific elements within the <detail> element.

Table 2 – API Schemas

8.3 HTTP Header Information

8.3.1 Description

The purpose of the HTTP header is to support security and logging functionally within the Royal Mail systems and it is mandatory that it is provided in the request message.

8.3.2 Request Message

All service requests to this API will be authorised in accordance with the Client ID and Secret passed in the HTTP headers. Please see table below for the elements which need to be populated in the HTTP header.	Optional	Description
Parameter		
Accept	No	This attribute accepts requests in application/soap+xml format.
X-IBM-Client-Id	No	Similar to a client username. Required to access the API.
X-IBM-Client-Secret	No	Similar to a client password. Required to access the API.
soapAction	No	“<operation name>”

Table 3 – HTTP Header Information in the API Request

The Client ID and Client-Secret is provided to you when you register your Application for new service on the developer portal. See section 5 for more details.

8.3.3 Example Data

Example request data for the HTTP Header:

Parameter	Value
Accept	application/soap+xml
X-IBM-Client-Id	f0e4f151-2041-4df2-b31d
X-IBM-Client-Secret	kT0lB2dK0wF6mK0rD8sD7oE7vP2mG7l
soapAction	"createShipment"

Table 4 – Example HTTP Header Information for API Request

8.4 Security Credentials

8.4.1 Description

Each request message to the Shipping API V2 web service must include valid security credentials in order to be accepted by the Royal Mail system. These comply with the OASIS WS security standards.

The credentials are encoded as defined in the OASIS standard *Web Services Security Username Token Profile version 1.0* and comprise the following:

- The username of the API user account,
- The SHA-1 hash of the password of the API user account.
- The Created timestamp for the request, and
- A Nonce.

The credentials are carried in a wsse:Security element included in the SOAP Header element of the request.

8.4.2 wsse:Security Element

All elements are relative to the wsse:Security element in the SOAP Header of the request message. Please note that the 'Occurs' column below is populated in the context of the Shipping API V2 implementation (as opposed to the OASIS security standard).

Element	Occurs	Description
Username	1-1	API username which will be supplied by RM
Password	1-1	Password Digest constructed as defined in OASIS ws-security Username Token Profile with the SHA-1 hash of the plain text password used as the password information. The plain text password will be provided to you by RM. See section below for details on how to construct this value.
Nonce	1-1	Random number. Although optional in the ws-security schema, this element must be provided by the customer for Shipping API V2 web services.
Created	1-1	Date and time. Although optional in the ws-security schema, this element must be provided by the customer for Shipping API V2 web service requests.

Table 5 – wsse:Security Element

Please note that none of the optional attributes in the WS-Security header need to be populated. If these are populated they will be ignored by the RM shipping system.

Nonce

- A Nonce is a random number that the sender generates to include in each usernameToken.
- The Shipping API V2 web service maintains a cache of Nonce values for a period of 5 minutes. Upon receipt of the message, a check will be made to ensure the Nonce has not already been used (i.e. it is not in the cache); messages containing Nonces that have already been used will be rejected.
- For Shipping API V2 the Nonce should be a 16 byte (128 bit) value.

The Nonce is presented as a Base64 encoded value in the wsse:Security element.

Note: when used in creating the Password Digest, the Nonce must be used in un-encoded form and not in Base64 form.

Created

- Created Timestamp is the date and time the request message was created by the customer. The string format of the timestamp should be constructed as defined in the WS-Security standards: that is YYYY-MM-DDThh:mm:ssZ.
- Messages that contain a Created timestamp older than five minutes will be rejected. Customers are recommended to synchronise their systems with a common time source to help prevent this error condition from occurring.

Password

The OASIS UsernameToken profile defines and describes the formula that computes the unique Password_Digest string submitted in the XML. For Shipping API V2 the password used in the below formula is the base 64 encoding of the SHA-1 hash of the plain text password.

This password value is then concatenated to generate the password digest with the Nonce and Created Time. It will then be rehashed using SHA-1.

The formula to use to construct the Password_Digest value is

Password_Digest = Base64(SHA-1(Nonce + Created + SHA-1(Password)))

Note the + symbol in the above algorithm represents a string concatenation of the three strings: Nonce from xml request, Created from xml request and the SHA-1 digest of the Password.

When processing a request, Shipping API V2 will:

- Check the Created timestamp is within five minutes of current time
- Check the Nonce has not been seen within the last five minutes
- Construct a Password_Digest using the Nonce and Created timestamp from the request message and the SHA-1 hash of API user account's password from its internal database.

- Compare the Password_Digest created with the one in the request message. If they match, authorisation will pass, otherwise it will fail and an authorisation failure SOAP fault will be returned.

8.4.3 Example Security data

Full XML examples of SOAP requests and responses are provided in Section 12.3 at the end of this document. This section provides a simplified example to illustrate the ws-security header only.

```
<soapenv:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-C0C91DCD9FA920EC8414557843873626">
      <wsse:Username>xxx</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest">RQeyP2a3n6b0JWNQxKYARZgSHrs=</wsse:Password>
      <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">E3NzHDHoQy8sUabCIqUhpw==</wsse:Nonce>
      <wsu:Created>2016-02-18T08:33:07.362Z</wsu:Created>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
```

8.4.4 References

For further details around the key concepts of WS-Security please see the following Microsoft TechNet article on WS-Security:

<http://msdn.microsoft.com/en-gb/library/ms977327.aspx>

8.5 createShipmentRequest element

The createShipment operation either results in a shipment being created (based on the information provided in the request) on RM platform or an appropriate error / warning message being returned to the customer. To invoke the createShipment operation, the customer shipping system constructs a SOAP request message as described in Section 7.

The createShipmentRequest element is contained in the SOAP Body and contains the following:

- An integrationHeader element (see section 8.5.1)
- A requestedShipment element (see section 8.5.2)

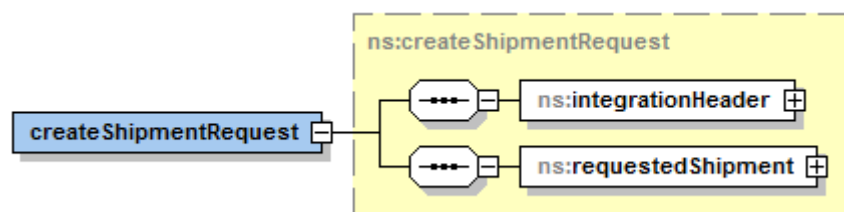


Figure 5 – createShipmentRequest element Structure

All elements in the table below are relative to the root createShipmentRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.5.1
requestedShipment	N/A	1-1	element	Container for the requested shipment details. See section below for more information. For allowable character set please see section 12.1.

Table 6 – createShipmentRequest Element

8.5.1 integrationHeader element

The purpose of this element is to support security and logging functionally within Royal Mail systems and it is mandatory that it is provided in all request messages. The integrationHeader element will also be present in all response messages.

Please see diagram below for a representation of the integrationHeader element:

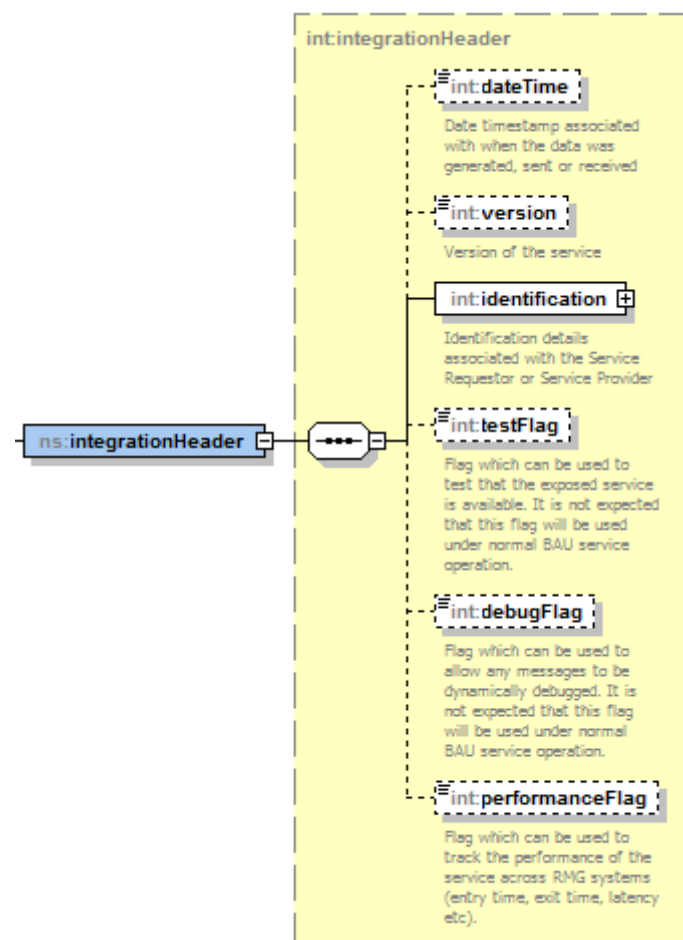


Figure 6 – integrationHeader element Structure

All elements defined below are relative to the integrationHeader element in the request message.

Element	Occurs	Data Type (Refer section 8.2)	Description
dateTime	0-1	dateTime	This should be populated with the date timestamp when the message was generated.
version	0-1	decimal	The version of the API currently being used (currently 2).

Element	Occurs	Data Type (Refer section 8.2)	Description
identification	1-1	element	Identification element used to hold the identity and transaction details associated with the customer. See identification table below for structure.
testFlag	0-1	boolean	Not used in the Shipping API V2 implementation.
debugFlag	0-1	boolean	Not used in the Shipping API V2 implementation.
performanceFlag	0-1	Boolean	Not used in the Shipping API V2 implementation

Table 7 – integrationHeader Element (Request)

Please see diagram below for a representation of the identification element:

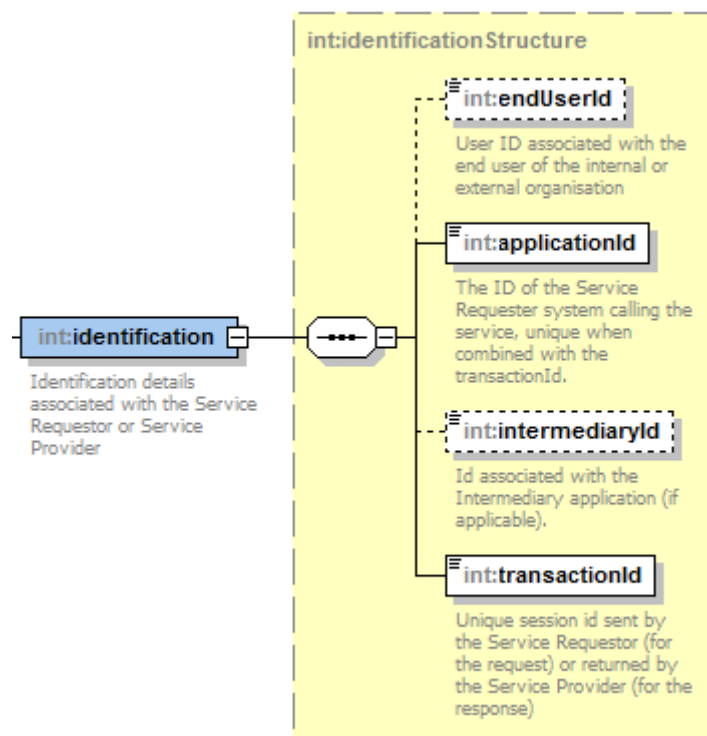


Figure 7 – identification Structure

All elements defined below are relative to the identification sub element in the integrationHeader element.

Element	Occurs	Data Type	Description
enduserID	0-1	identifier	Not used in the Shipping API V2 implementation.
applicationId	1-1	identifier	Value provided by the Royal Mail Customer Solutions Team
intermediaryId	0-1	identifier	Not used in the Shipping API V2 implementation.
transactionId	1-1	identifier	This is a unique number used to identify the transaction as provided by the customer system. Any value can be provided in this field but must contain only the characters 'a-z', 'A-Z', '0-9', '/' and '-'. It allows the consuming application to correlate the response message to its request.

Table 8 – integrationHeader identification Element

8.5.2 requestedShipment element

The requestedShipment element acts as a container for the requested shipment details and is represented by the diagram below.

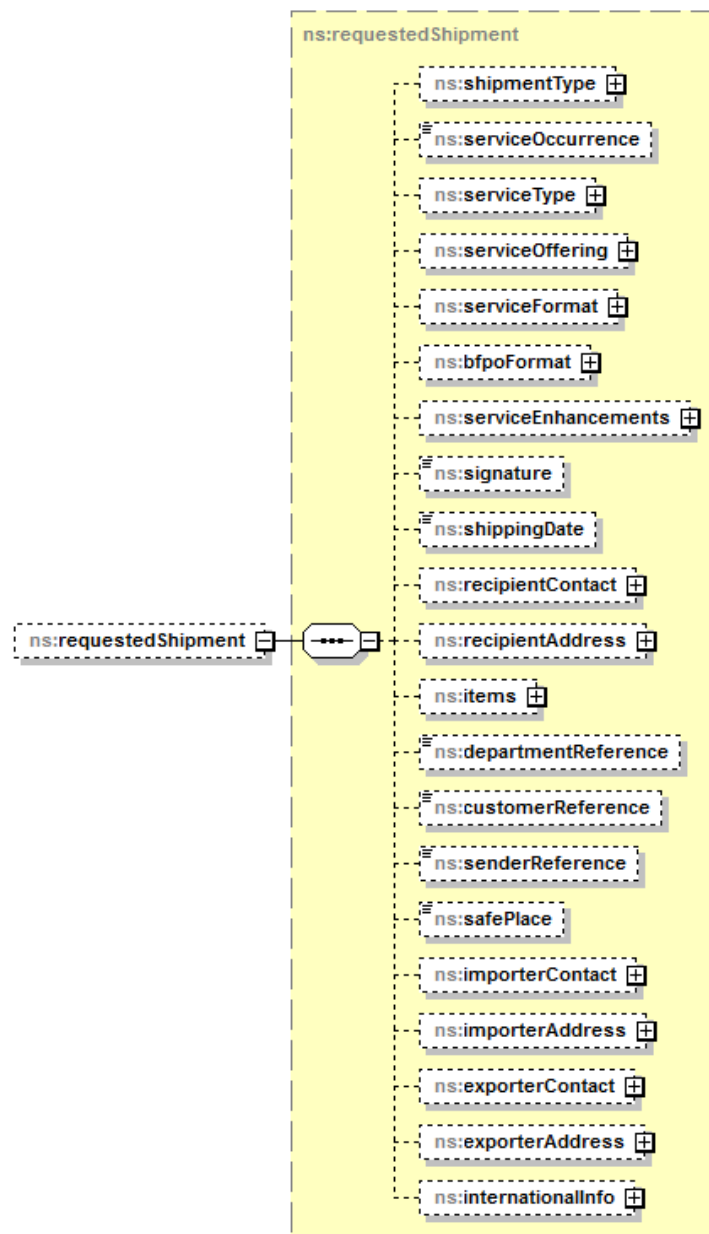


Figure 8 – requestedShipment element Structure

All elements in the table below are relative to the requestedShipment element.

Element	Max Length	Occurs	Data Type	Description
shipmentType	N/A	0-1	element	Container for shipmentType code.
shipmentType/code	8	0-1	identifier	Mandatory. Specifies whether the shipment being created is a standard delivery service or a returns service. Accepted values are 'Delivery' and 'Return'. Please note that this field is not case sensitive.
serviceOccurrence	2	0-1	ordinal	Part of the customer's contract identifier. In conjunction with the Service Offering it identifies an agreement line on the customer's account. If only one Service Reference exists then this is not required. No leading zero is required.
serviceType	N/A	0-1	element	Mandatory. Container for serviceType code

Element	Max Length	Occurs	Data Type	Description
serviceType/code	4	0-1	identifier	Mandatory. The system Service Type of the shipment. For the list of permissible values, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data.
serviceOffering	N/A	0-1	element	Mandatory. Container for serviceOfferingCode
serviceOffering/ serviceOfferingCode/	N/A	1-1	element	Mandatory. Container for serviceOffering code value
../serviceOfferingCode/code	3	0-1	identifier	Mandatory. The Service Offering code for the mail item ordered. Please note that this field is case sensitive. For the list of permissible values, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data.
serviceFormat	N/A	0-1	element	Container for serviceFormatCode
serviceFormat/ serviceFormatCode	N/A	1-1	element	Container for serviceFormat code value
../serviceFormatCode/code	4	0-1	identifier	The Service Format code for the shipment. Note that this field is case sensitive. For the list of permissible values, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data.
bfpoFormat	N/A	0-1	element	Container for bFPOFormatCode
bfpoFormat/bFPOFormatCode	N/A	1-1	element	Container for bFPOFormat code value
../bFPOFormatCode/code	4	0-1	identifier	For HM Forces (BFPO) Service Type only when the Service Format is not International Flat, International Letter or International Packet. For the list of permissible values, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data.
serviceEnhancements	N/A	0-1	element	This is a complex type and is detailed below
Signature	1	0-1	boolean	For RM Tracked items only, this element specifies whether a signature is required on delivery. If this element is not included then it defaults to false.
shippingDate	10	0-1	date	This is the date that the item will be physically sent (in the format YYYY-MM-DD). This may be up to 28 days in the future. Please note that for returns a Shipping date must be provided.
recipientContact	N/A	0-1	element	Mandatory. This is a complex type and is detailed below in the Contact element
recipientAddress	N/A	0-1	element	Mandatory. This is a complex type and is detailed below in the Address element
Items	N/A	0-1	element	Optional. This is a complex type and is detailed below
departmentReference	10	0-1	identifier	This is the department reference code that customers can define in OBA. This is visible in the system departmental references GUI.
customerReference	12	0-1	identifier	This field allows customers to supply a reference that applies to multiple shipments and is included to mirror the functionality offered by the Customer Reference field in OBA, whereby a reference can be associated to a group of items. For references that apply to a single shipment, the senderReference field should be used. Warning: Misuse of this field may result in incorrect billing.

Element	Max Length	Occurs	Data Type	Description
senderReference	20	0-1	identifier	This field allows the user to supply their own reference number. Where supported (e.g. Tracked Returns) this number will appear on the label.
safePlace	30 (24)	0-1	comment	For Tracked non-signature service offerings only; this field allows a string that gives details of the recipient's designated safeplace (e.g. "inside the porch").
importerContact	N/A	0-1	element	Optional. This is a complex type and is detailed below in the Contact element
importerAddress	N/A	0-1	element	Optional. This is a complex type and is detailed below in the Address element
exporterContact	N/A	0-1	element	Optional. This is a complex type and is detailed below in the Contact element
exporterAddress	N/A	0-1	element	Optional. This is a complex type and is detailed below in the Address element
internationalInfo	N/A	0-1	element	Optional. This is a complex type and is detailed below

Table 9 – requestedShipment Element

Note: If you specify more than 24 characters in the Safeplace element and generate a PDF label, this field will be truncated to 24 characters in order to fit onto the label. It is therefore recommended that customers enter values within the 24 character display limit.

serviceEnhancements element

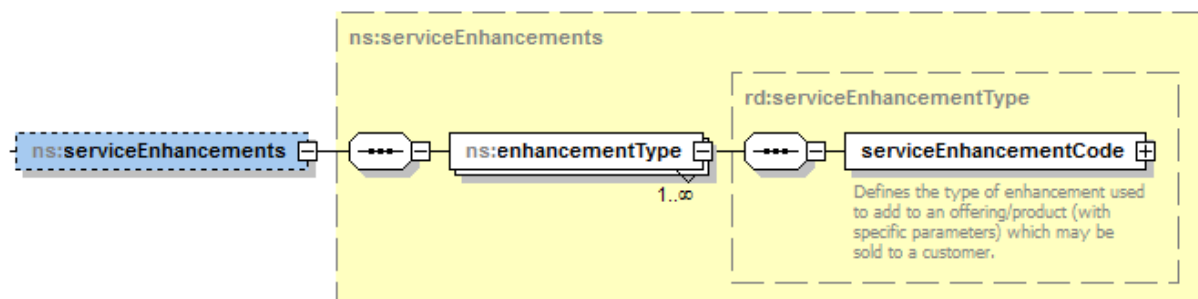


Figure 9 – serviceEnhancements element Structure

All elements in the table below are relative to serviceEnhancements.

Element	Max Length	Occurs	Data Type	Description
enhancementType	N/A	1-∞	element	Container for serviceEnhancementCode
enhancementType/ serviceEnhancementCode	N/A	1-1	element	Container for service enhancement code
../serviceEnhancementCode /code	4	0-1	identifier	If serviceEnhancement is included then enhancementType is Mandatory. There can never be more than one enhancementType specified from each Service Enhancement Group. Note that this field is case sensitive. For the list of permissible values, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data .

Table 10 – serviceEnhancements Element

contact element

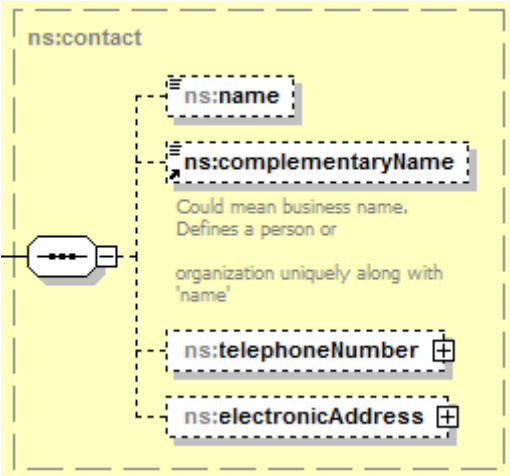


Figure 10 – contact Structure

Please note that all elements under the 'telephoneNumber' and 'electronic Address' elements and which are not listed in the table below, do not need to be populated.

Element	Max Length	Occurs	Data Type	Description
Name	80	0-1	longName	Mandatory. Contact name
complementaryName	64	0-1	longName	Business name
telephoneNumber	N/A	0-1	element	Container for telephone number details
telephoneNumber/ telephoneNumber	12	0-1	integer	UK mobile phone number. Required if a SMS service enhancement is selected.
electronicAddress	N/A	0-1	element	Container for electronic address details
electronicAddress/ electronicAddress	60	0-1	description	Email address to be used for notifications. Required if Email enhancement is selected

Table 11 – contact Element

Note: If you specify more than 35 characters in name or complimentary name fields and generate a PDF label, these fields will be truncated to 35 characters in order to fit onto the label. It is therefore recommended that customers enter values within the 35 character display limit.

address element

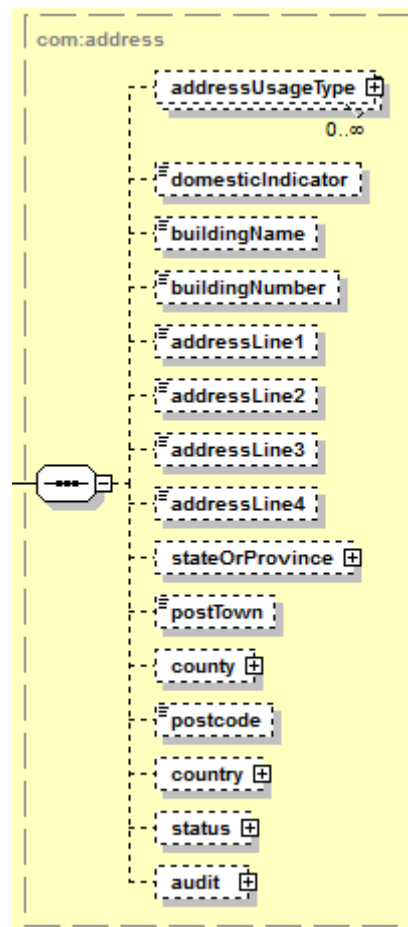


Figure 11 – address element Structure

Please note that elements which are not listed in the table below do not need to be populated.

Element	Max Length	Occurs	Data Type	Description
buildingName	35	0-1	identifier	This field will be used if supplied but is not necessary to process a shipment. It does not replace the need for a full addressLine1 input.
buildingNumber	4	0-1	identifier	This field will be used if supplied but is not necessary to process a shipment. It does not replace the need for a full addressLine1 input.
addressLine1	80 (35)	0-1	description	Mandatory. First line of the address
addressLine2	80 (35)	0-1	description	Second line of the address
addressLine3	80 (35)	0-1	description	Third line of the address
postTown	40 (35)	0-1	name	Mandatory. Town or City. The postTown will be updated to match the postTown resolved postcode.
Postcode	15	1-1	identifier	Mandatory for UK addresses. If the Shipment Type is "Return" then this must match the postcode of the registered return address.
Country	N/A	0-1	element	Container for country details
country/countryCode	N/A	0-1	element	Container for country code details

Element	Max Length	Occurs	Data Type	Description
../countryCode/code	2	1-1	identifier	The relevant country code for the address. Note that this field is case sensitive. For the list of allowable values, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data . For domestic services this must be GB.

Table 12 – address Element

Note: If you specify more than 35 characters in addressLine1, addressLine2, addressLine3 or PostTown fields and generate a PDF label, these fields will be truncated to 35 characters in order to fit onto the label. It is therefore recommended that customers enter values within the 35 character limit.

items element

The items element can contain multiple item elements. Each item element is constructed as follows:

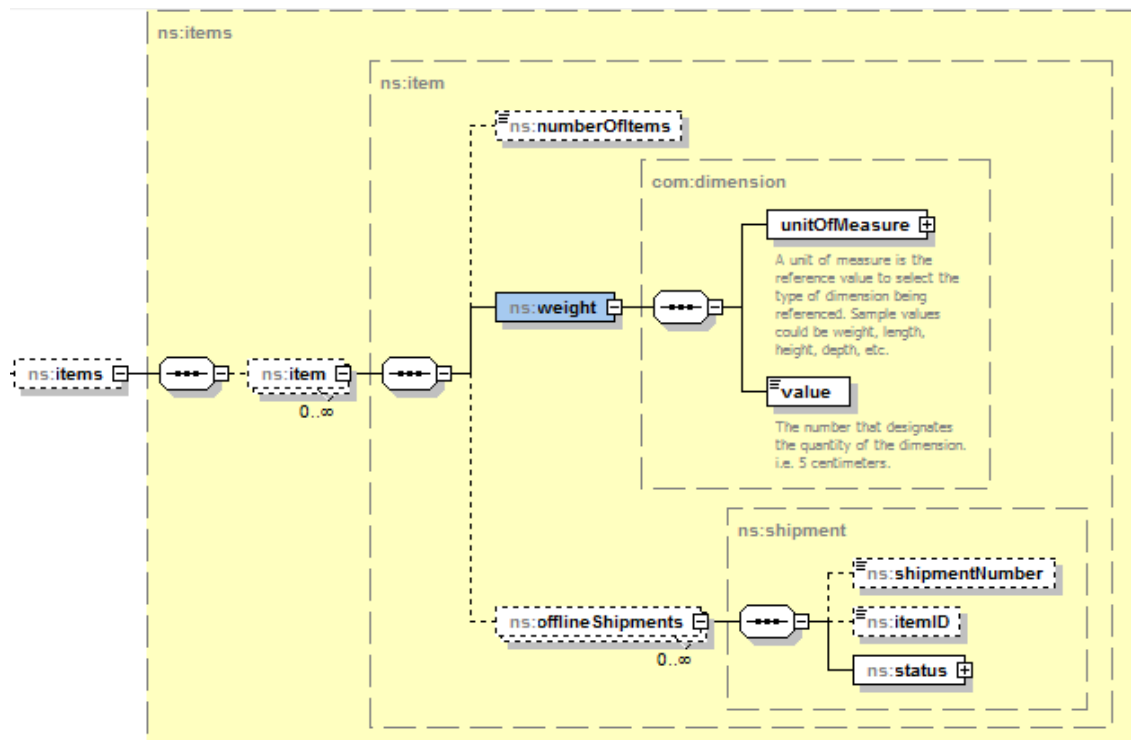


Figure 12 – items element Structure

Please note that elements which are not listed in the table below do not need to be populated.

Element	Max Length	Occurs	Data Type	Description
Item	N/A	0-99	element	Container for item details. Up to 9 shipments can be created with one request.
item/numberOfItems	2	0-1	cardinal	Number of items for the associated weight
item/weight	N/A	1-1	element	Container for item weight details
../weight/value	5	1-1	float	Weight in grams of each item (no decimal places). If the service has a weight band, for example Special Delivery, then the upper band will be used. For example, 150 grams will use the 100 to 200 grams band. Tracked services, for example, do not have a band and therefore take the actual weight. Note: Where Average Weight End of Day option is turned ON, for Average Weight Products populate with '0'. For more information go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data for Average Weight Products
../weight/unitOfMeasure	N/A	1-1	element	Container for unit of measure
../unitOfMeasure/unitOfMeasureCode	N/A	1-1	element	Container for unit of measure code
../unitOfMeasure/unitOfMeasureCode/code	N/A	1-1	identifier	Must be 'g' for grams
offlineShipments	N/A	0-9	Element	Optional. This is a complex type and is detailed below

Table 13 – items Element

internationalInfo element

As suggested by the name, this element is filled in the request shipment API when the shipment is for a non-European international destination. The printDocument request will validate you have the pre-requisite data and return the PDF documents for CN22 or CN23 and/or Commercial Invoice (CI). Note- This element cannot produce CN22, CN23 or Commercial Invoices for Average Weight Products (Go to Royal Mail API (Developer) Portal and refer to Shipping API Reference Data for list of Average Weight Products) when the End of Day Average Weight setting is turned on (by Royal Mail for you); as the weight is not then submitted in the createShipmentRequest call.



Figure 13 – internationalInfo element Structure

Please note that elements which are not listed in the table below do not need to be populated.

Element	Max Length	Occurs	Data Type	Description
parcels	N/A	0-1	element	This is a complex type and is detailed below
shipperExporterVat No	35	0-1	identifier	Exporter's VAT number
recipientImportersVatNo	35	0-1	identifier	Recipient / Importer's VAT number
originalExportShipmentNo	14	0-1	identifier	Second line of the address
documentsOnly	N/A	0-1	boolean	Indication of export of documents only
shipmentDescription	30	0-1	shortDescription	Description of the shipment
comments	128	0-1	shortDescription	Comments regarding the shipment
invoiceDate	N/A	0-1	date	Commercial Invoice Date
termsOfDelivery	3	0-1	Identified	3 Character code indicating terms of delivery (IncoTerms). The relevant international standard code for the terms of delivery; refer to the International Chamber of Commerce (ICC) for required listing of codes
purchaseOrderRef	35	0-1	identifier	Vendor's Purchase Order Number for the shipment

Table 14 – internationalInfo Element

parcels element

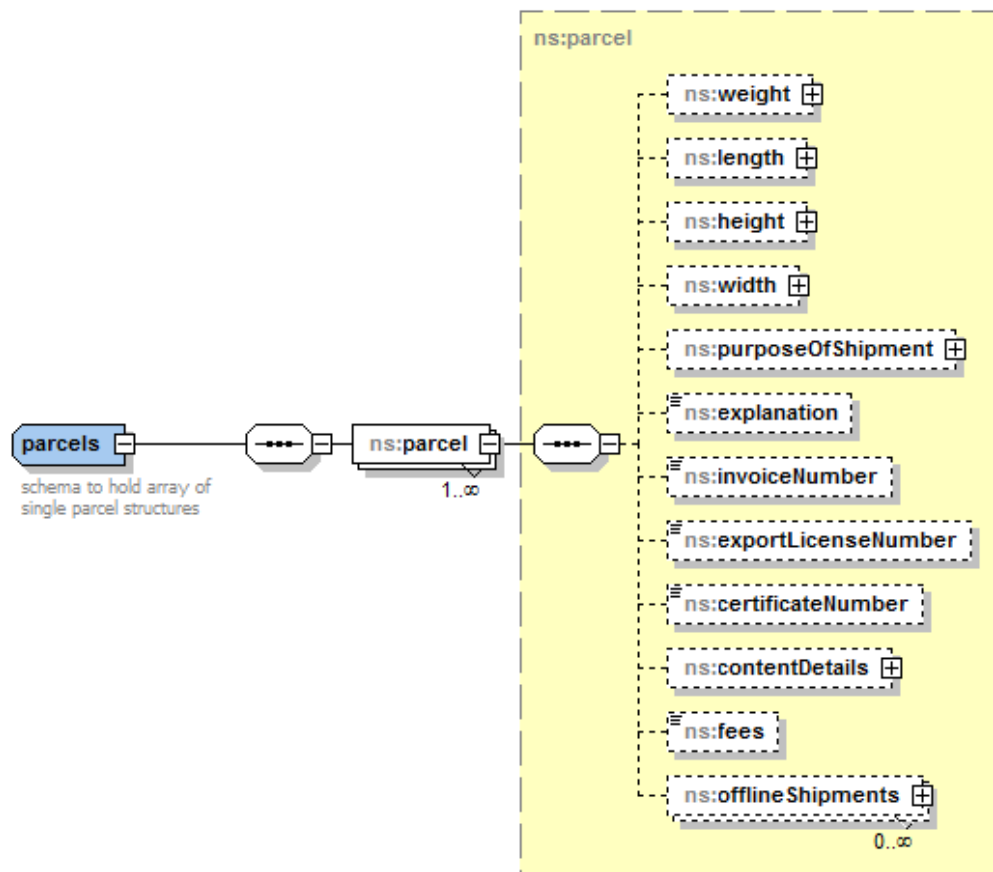


Figure 14 – parcels element Structure

Please note that elements which are not listed in the table below do not need to be populated.

Element	Max Length	Occurs	Data Type	Description
parcel	N/A	0-9	element	Container for the details pertaining to a single shipment
parcel/weight	35	0-1	identifier	Weight of shipment in Kg
parcel/length	35	0-1	identifier	Length dimension of shipment, in cm
parcel/height	14	0-1	identifier	Height dimension of shipment, in cm
parcel/width	N/A	0-1	boolean	Width dimension of shipment, in cm
parcel/purposeOfShipment	30	0-1	shortDescription	Purpose of shipment (also known as Nature of Goods). These are 2-3 character codes as defined below: "21" for Returned Goods "31" for Gift "32" for Commercial Sample "91" for Documents "991" for Mixed Content "999" for Other
parcel/explanation	128	0-1	shortDescription	Comments regarding the shipment
parcel/invoiceNumber	N/A	0-1	date	Invoice number
parcel/exportLicenseNumber	N/A	0-1	date	Export licence number
parcel/certificateNumber	N/A	0-1	date	Certificate number
parcel/contentDetails	N/A	0-1	element	This is a complex type and is detailed below

Element	Max Length	Occurs	Data Type	Description
parcel/fees	N/A	0-1	date	Parcel fees
parcel/offlineShipments	3	0-1	identifier	Optional. This is a complex type and is detailed below

Table 15 – parcels Element

contentDetail element

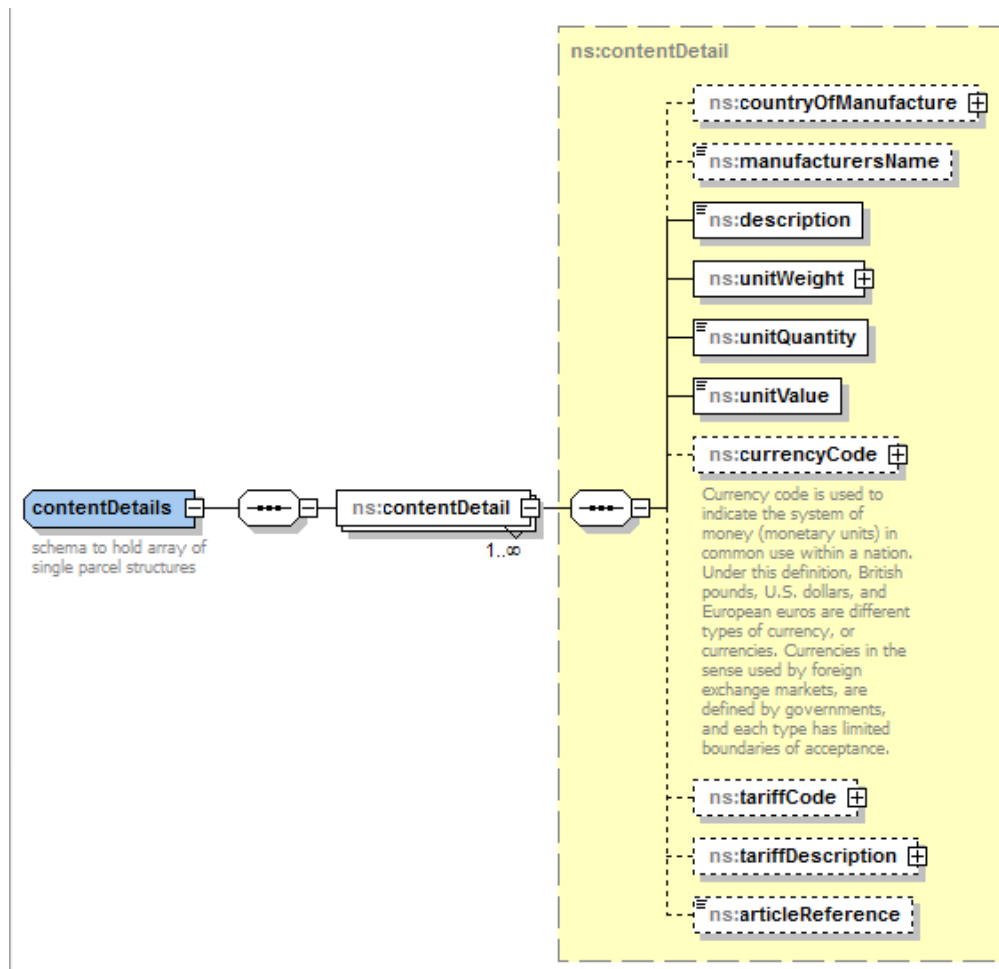


Figure 15 – contentDetail element Structure

Please note that elements which are not listed in the table below do not need to be populated.

Element	Max Length	Occurs	Data Type	Description
contentDetail	N/A	0-9	element	Container for the details pertaining to a single shipment
contentDetail /countryOfManufacture	N/A	0-1	element	Country code representing the country in which the goods where manufactured.
contentDetail /countryOfManufacture/countryCode	N/A	1-1	element	Container for country in which the goods where manufactured

Element	Max Length	Occurs	Data Type	Description
contentDetail/countryOfManufacture/countryCode/code	N/A	0-1	identifier	Mandatory. Container for country code representing the country in which the goods were manufactured. Note that this field is case sensitive. For the list of allowable values, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data .
contentDetail/manufacturersName	35	0-1	identifier	Name of manufacturer of goods in the shipment
contentDetail/description	14	0-1	shortDescription	Description of goods being delivered
contentDetail/unitWeight	N/A	0-1	boolean	Container for weight dimension of goods being delivered
contentDetail/unitWeight/unitOfMeasure	N/A	0-1	element	Container for Unit of weight dimension of goods being delivered
contentDetail/unitWeight/unitOfMeasure/unitOfMeasureCode	N/A	0-1	element	Container for code for Unit of weight dimension of goods being delivered
contentDetail/unitWeight/unitOfMeasure/unitOfMeasureCode/code	N/A	0-1	identifier	Unit of weight dimension of goods being delivered. Default is KG.
contentDetail/unitWeight/value	N/A	1-1	float	Value of weight dimension of goods being delivered
contentDetail/unitQuantity	N/A	1-1	integer	Quantity of content items within the shipment
contentDetail/unitValue	N/A	1-1	decimal	Value of content items within the shipment
contentDetail/currencyCode	N/A	0-1	element	Container for currency code for value of content items within the shipment
contentDetail/currencyCode/code	N/A	0-1	identifier	Mandatory. Currency code for value of content item within the shipment
contentDetail/tariffCode	N/A	0-1	element	Container for tariff code for content items within the shipment
contentDetail/tariffCode/code	11	0-1	identifier	Tariff code for content item within the shipment This is a number – see https://www.gov.uk/trade-tariff
contentDetail/tariffDescription	35	0-1	date	Description that compliments the tariff code supplied

Table 16 – contentDetail Element

offlineShipments element

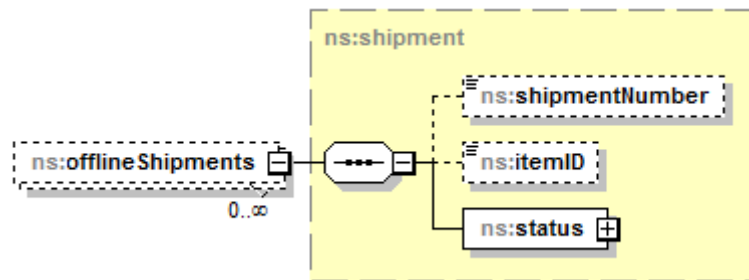


Figure 16 – offlineShipments element Structure

This element is filled in createShipmentRequest API only when you are using the offline barcoding. As mentioned in section 4.2 prior to calling this API you should have requested 1D and 2D barcode ranges through the request1Drange and request2Drange API.

Please note that elements which are not listed in the table below do not need to be populated.

Element	Max Length	Occurs	Data Type	Description
shipmentNumber	13	0-1	identifier	Offline 1D Linear Barcode number assigned to the shipment. This should be from the 1D barcode range received by you in response to request1Drange
itemID	35	0-1	identifier	Offline 2D Item ID number assigned to the shipment. This should be from the 2D barcode range received by you in response to request2Drange
status	35	1-1	element	Container for status of shipment
status/status	N/A	1-1	element	Container for statusCode of shipment
status/status/statusCode	N/A	1-1	element	Container for statusCode code of shipment
status/status/statusCode/code		1-1	identifier	Code for status of Offline Shipment. Valid values are: 'AllocatedOffline': You have to call printlabel request API before adding the shipment to manifest, and 'PrintedOffline': printLabel is not required when using this status in the createShipment call

Table 17 – offlineShipments Element

NOTE:

- If you use the Status Code within the OfflineShipments element in the createShipment request as 'PrintedOffline', you do not need to use the printLabel request to add the item to the manifest / End of Day process.
- If you do not use the Status Code field, ie does not populate it with a value, then the shipment is flagged 'AllocatedOffline' by default and you will need to complete the printLabel request in order to add the shipment to the manifest / End of Day process.

8.5.3 Example Data

Online Barcoding:

The section below shows a simplified (without SOAP wrappers and WS-Security elements) createShipmentRequest(Domestic) SOAP operation as an example. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set

```
<v2:createShipmentRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-09-02T14:55:00Z</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>0000000000</v1:applicationId>
      <v1:transactionId>0000000151</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:requestedShipment>
    <v2:shipmentType>
      <code>Delivery</code>
    </v2:shipmentType>
    <v2:serviceOccurrence>1</v2:serviceOccurrence>
    <v2:serviceType>
      <code>1</code>
    </v2:serviceType>
    <v2:serviceOffering>
      <serviceOfferingCode>
        <code>CRL</code>
      </serviceOfferingCode>
    </v2:serviceOffering>
    <v2:serviceFormat>
      <serviceFormatCode>
        <code>P</code>
      </serviceFormatCode>
    </v2:serviceFormat>
    <v2:bfpoFormat>
      <bFPOFormatCode>
        <code>
        </bFPOFormatCode>
      </v2:bfpoFormat>
    <v2:serviceEnhancements>
      <!--Zero or more repetitions:-->
      <v2:enhancementType>
        <serviceEnhancementCode>
          <code>
        </serviceEnhancementCode>
      </v2:enhancementType>
    </v2:serviceEnhancements>
    <v2:signature>0</v2:signature>
    <v2:shippingDate>2015-08-28</v2:shippingDate>
    <v2:recipientContact>
      <v2:name>TEST</v2:name>
      <v2:complementaryName>SHIPMENT</v2:complementaryName>
      <v2:telephoneNumber>
        <countryCode>0044</countryCode>
        <telephoneNumber>07436039784</telephoneNumber>
      </v2:telephoneNumber>
      <v2:electronicAddress>
        <electronicAddress>sonja.blackburn@royalmail.com</electronicAddress>
      </v2:electronicAddress>
    </v2:recipientContact>
    <v2:recipientAddress>
      <!--Conditional: Building Name or Building Number (2D)-->
      <buildingName>
      </buildingName>
      <buildingNumber>13</buildingNumber>
      <addressLine1>AddressLine 1</addressLine1>
      <addressLine2>AddressLine 2</addressLine2>
      <addressLine3>AddressLine 3</addressLine3>
      <addressLine4>AddressLine 4</addressLine4>
      <stateOrProvince>
        <stateOrProvinceCode>
          <!--Optional:-->
          <identifier>
          </code>
        </name>
      </stateOrProvince>
    </v2:recipientAddress>
  </v2:requestedShipment>
</v2:createShipmentRequest>
```

```

        </description/>
        </stateOrProvinceCode>
    </stateOrProvince>
    <postTown>LONDON</postTown>
    <postcode>EC 1A 1BB</postcode>
    <country>
        <countryCode>
            <code>GB</code>
        </countryCode>
    </country>
</v2:recipientAddress>
<v2:items>
    <v2:item>
        <v2:numberOfItems>1</v2:numberOfItems>
        <v2:weight>
            <unitOfMeasure>
                <unitOfMeasureCode>
                    <code>g</code>
                </unitOfMeasureCode>
            </unitOfMeasure>
            <value>250</value>
        </v2:weight>
    </v2:item>
</v2:items>
<!--Customer Reference should be populated the same as your OBA Reference Number-->
<v2:customerReference/>
<!--sendersReference is used for individual shipment reference-->
<v2:senderReference>SendersRef</v2:senderReference>
<v2:safePlace>0</v2:safePlace>
</v2:requestedShipment>

```

```
</v2:createShipmentRequest>
```

Offline Barcoding:

The section below shows createShipmentRequest as an example where, unlike the example above, offline barcoding is used. Please note that this request is sent after you have already received the 1D and 2D barcode ranges through the request1Drange and request2Drange APIs

```

<v2:createShipmentRequest>
  <v2:integrationHeader>
    <v1:dateTime>2016-01-08T08:52:03</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>11111113</v1:applicationId>
      <v1:transactionId>0000000001</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:requestedShipment>
    <v2:shipmentType>
      <code>Delivery</code>
    </v2:shipmentType>
    <v2:serviceOccurrence>1</v2:serviceOccurrence>
    <v2:serviceType>
      <code>D</code>
    </v2:serviceType>
    <v2:serviceOffering>
      <serviceOfferingCode>
        <code>SD6</code>
      </serviceOfferingCode>
    </v2:serviceOffering>
    <v2:serviceFormat>
      <serviceFormatCode>
        <code>N</code>
      </serviceFormatCode>
    </v2:serviceFormat>
    <v2:bfpoFormat>
      <bFPOFormatCode>
        <identifier/>
        <code/>
        <name/>
        <description/>
      </bFPOFormatCode>
    </v2:bfpoFormat>
    <v2:serviceEnhancements>
      <v2:enhancementType>
        <serviceEnhancementCode>
          <identifier/>
          <code/>

```

```

    <name/>
    <description/>
    </serviceEnhancementCode>
    </v2:enhancementType>
  </v2:serviceEnhancements>
  <v2:signature>1</v2:signature>
  <v2:shippingDate>2016-01-11</v2:shippingDate>
  <v2:recipientContact>
    <v2:name>MS J BLOGGS</v2:name>
    <v2:complementaryName>LADY JO</v2:complementaryName>
    <v2:telephoneNumber>
      <countryCode>0044</countryCode>
      <telephoneNumber>07801123456</telephoneNumber>
    </v2:telephoneNumber>
    <v2:electronicAddress>
      <electronicAddress>sonja.blackburn@royalmail.com</electronicAddress>
    </v2:electronicAddress>
  </v2:recipientContact>
  <v2:recipientAddress>
    <addressLine1>TEST ADDRESS LINE 1</addressLine1>
    <postTown>London</postTown>
    <postcode>S49 1HQ</postcode>
    <country>
      <countryCode>
        <code>GB</code>
      </countryCode>
    </country>
  </v2:recipientAddress>
  <v2:items>
    <v2:item>
      <v2:numberOfItems>2</v2:numberOfItems>
      <v2:weight>
        <unitOfMeasure>
          <unitOfMeasureCode>
            <code>g</code>
          </unitOfMeasureCode>
        </unitOfMeasure>
        <value>100</value>
      </v2:weight>
      <v2:offlineShipments>
        <v2:shipmentNumber>BQ070802658GB</v2:shipmentNumber>
        <!--itemID is the 2D barcode range -- it is the unique sequence number in the 2D barcode-->
        <v2:itemID>7250171</v2:itemID>
        <v2:status>
          <status>
            <!--! If Status includes PRINTED OFFLINE then customer does not need to complete the printLabel request -->
          </status>
          <statusCode/>
        </v2:status>
        <validFrom>2016-01-08T08:00:05</validFrom>
      </v2:offlineShipments>
      <v2:offlineShipments>
        <v2:shipmentNumber>BQ070802661GB</v2:shipmentNumber>
        <v2:itemID>7250172</v2:itemID>
        <v2:status>
          <status>
            <statusCode/>
          </status>
          <validFrom>2016-01-08T08:00:05</validFrom>
        </v2:status>
      </v2:offlineShipments>
    </v2:item>
  </v2:items>
  <v2:customerReference> </v2:customerReference>
  <v2:senderReference>SenderReference1</v2:senderReference>
</v2:requestedShipment>

```

8.6 createShipmentResponse

The response for a requested createShipment operation is constructed as a SOAP response message as described in Section 7. The createShipmentResponse element is contained in the SOAP Body element and contains the following:

- An integrationHeader element (see section 8.6.1)
- An optional completedShipmentInfo element (see section 8.6.2) which will be populated on completion of having successfully fulfilled a createShipment request. The completedShipmentInfo element will not be present in the response if an

error occurred in the processing of the request – the error details will be included in the integrationFooter element.

- An optional integrationFooter element (see section 8.6.3)

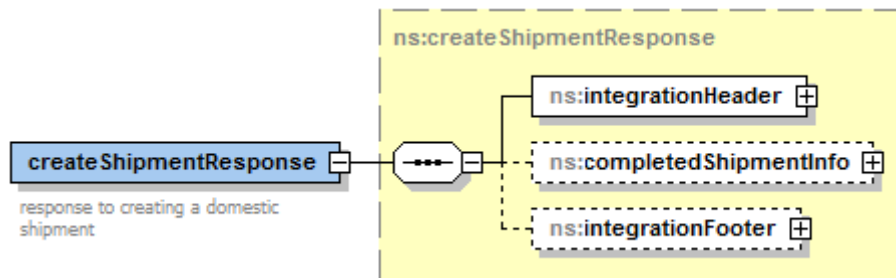


Figure 17 – createShipmentResponse Structure

All elements in the table below are relative to the root createShipmentResponse element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.6.1
completedShipmentInfo	N/A	0-1	element	Container for the completed shipment information. See section 8.6.2 below for more information.
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the createShipment operation. Please see section 8.6.3 for more information.

Table 18 – createShipmentResponse Element

8.6.1 integrationHeader

The integrationHeader element will also be present in all response messages. Please see diagram in Figure 6 in section 8.5.1 for a representation of the integrationHeader element:

Element	Occurs	Data Type	Description
dateTime	0-1	dateTime	This is always returned and contains the same value provided in the request.
Version	0-1	decimal	This is always returned and contains the same value provided in the request.
Identification	1-1	element	This is always returned and contains the same values provided in the request.
testFlag	0-1	boolean	Not used in the Shipping API V2 implementation.
debugFlag	0-1	boolean	Not used in the Shipping API V2 implementation.
performanceFlag	0-1	Boolean	Not used in the Shipping API V2 implementation

Table 19 – integrationHeader Element (Response)

The integrationHeader contains an identification element which is described below.

Element	Occurs	Data Type	Description
enduserID	0-1	identifier	Not used in the Shipping API V2 implementation.
applicationId	1-1	identifier	SSame value as provided in the request.
intermediaryId	0-1	identifier	Not used in the Shipping API V2 implementation.
transactionId	1-1	identifier	This is always returned with the same unique transaction number provided in the request.

Table 20 – intgrationHeader identification Element (Response)

8.6.2 completedShipmentInfo element

The `completedShipmentInfo` element is a container for information related to the completed shipment.

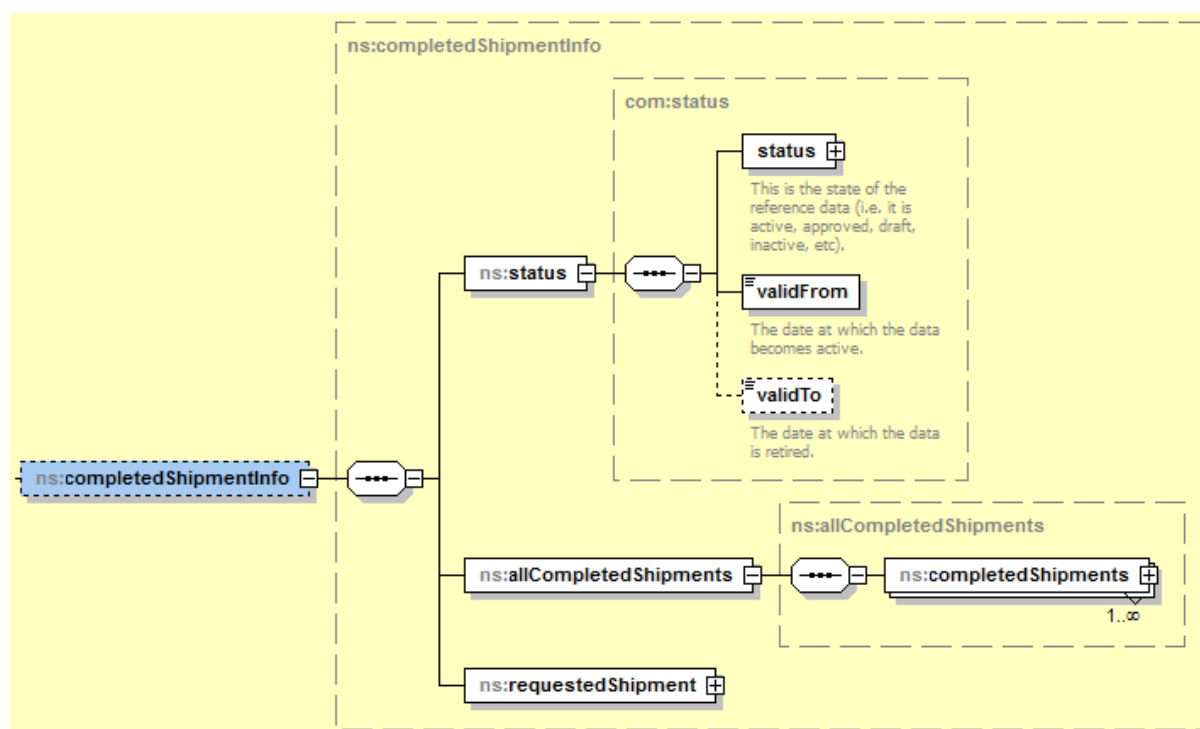


Figure 18 – completedShipmentInfo element Structure

All elements in the table below are relative to the `completedShipmentInfo` element.

Element	Max Length	Occurs	Data Type	Details
Status	N/A	1-1	element	Container for shipment status information
status/status	N/A	1-1	element	Container for shipment status details
../status/statusCode	N/A	1-1	element	Container for shipment status code
../../statusCode/code	9	1-1	identified	This will be set to 'Allocated'. For the full set of values the shipment status can hold, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to Shipping API Reference Data .
../status/validFrom	N/A	1-1	dateTime	Datetime value associated with when the shipment status code is valid from.
../status/validTo	N/A	0-1	dateTime	Datetime value associated with when the shipment status code is valid to. This is NOT USED.
allCompletedShipments	N/A	1-1	element	Container for completed shipments
allCompletedShipments/completedShipments	N/A	1- ∞	element	Container for completed shipments from processing the associated request.
../completedShipments/weight	N/A	0-1	element	The weight band the completed shipments are within
../../weight/value	5	1-1	Float	Weight in grams of each of the items (no decimal places)
../../weight/unitOfMeasure	N/A	1-1	element	Container for unit of measure

Element	Max Length	Occurs	Data Type	Details
../weight/unitOfMeasure/unitOfMeasureCode	N/A	1-1	element	Container for unit of measure code
../weight/unitOfMeasure/unitOfMeasureCode/code	5	1-1	identified	Must be 'g' for grams
../completedShipments/shipments	N/A	1- ∞	element	Container for completed shipment details
../completedShipments/shipments/shipmentNumber	13	1-9	identified	For barcoded products i.e. you have used offline barcoding operation, this field will contain the 1D barcode number sent in the request. For non-barcoded products, this field will contain the Shipping API V2 internal reference number. For requests where there are multiple items, there will be a corresponding shipmentNumber for each item.
../completedShipments/shipments/shipment	n/a	1-9	element	Container for each completed shipment containing 1D shipment number and 2D Item ID and shipment status. See section 8.6.3 below for more information.
requestedShipment	n/a	1-1	element	The requestedShipment information as provided in the associated createShipmentRequest message. See section 8.5.2 for more information.

Table 21 – completedShipmentInfo Element

8.6.2.1 shipment element

This is a Container for each completed shipment containing 1D shipment number and 2D Item ID and shipment status.

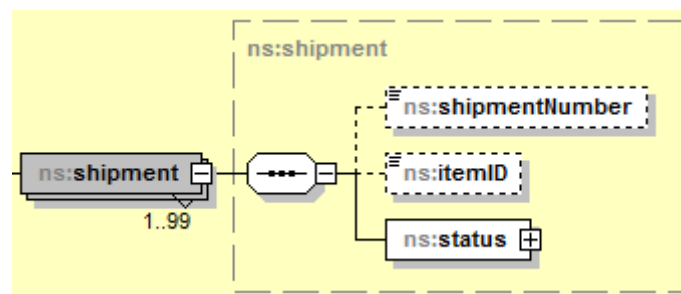


Figure 19: shipment element Structure

Element	Max Length	Occurs	Data Type	Description
../shipment/shipment Number	n/a	1-1	element	For barcoded products i.e. You have used offline barcoding operation, this field will contain the 1D barcode number sent in the request. For non-barcoded products, this field will contain the Shipping API V2 internal reference number
itemID	n/a	1-1	element	This is the 2D item ID used in the label
Status	n/a	1-1	element	Container for the status code
../status/statusCode	n/a	1-1	element	Container for code
../../statusCode/Code	n/a	1-1	element	Current status of the item i.e. Allocated

Table 222 – shipment element

8.6.3 integrationFooter

The purpose of this element is to return any business error and warning messages back to the customer. The integrationFooter element will only be present in a response message if there are any business errors or warnings associated with processing a web service request. Please note that all technical exceptions will result in a SOAP fault being generated.

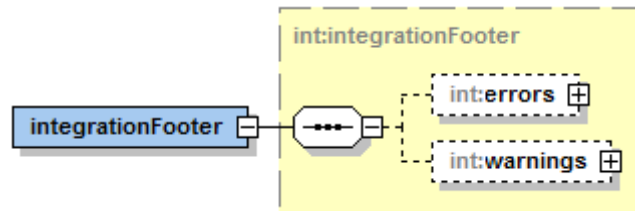


Figure 19 – integrationFooter Structure

All elements in the table below are relative to the integrationFooter element.

Element	Occurs	Data Type	Description
Errors	0-1	element	See table below for structure. This will contain details of any errors encountered in processing a web service request. Please see the Shipping API V2 page on the Royal Mail API (Developer) Portal for the full list of all errors which can be returned.
Warnings	0-1	element	See table below for structure. This will contain details of any warnings encountered in processing a web service request. Please see the Shipping API V2 page on the Royal Mail API (Developer) Portal for the full list of all warnings which can be returned.

Table 23 – integrationFooter Element

errors

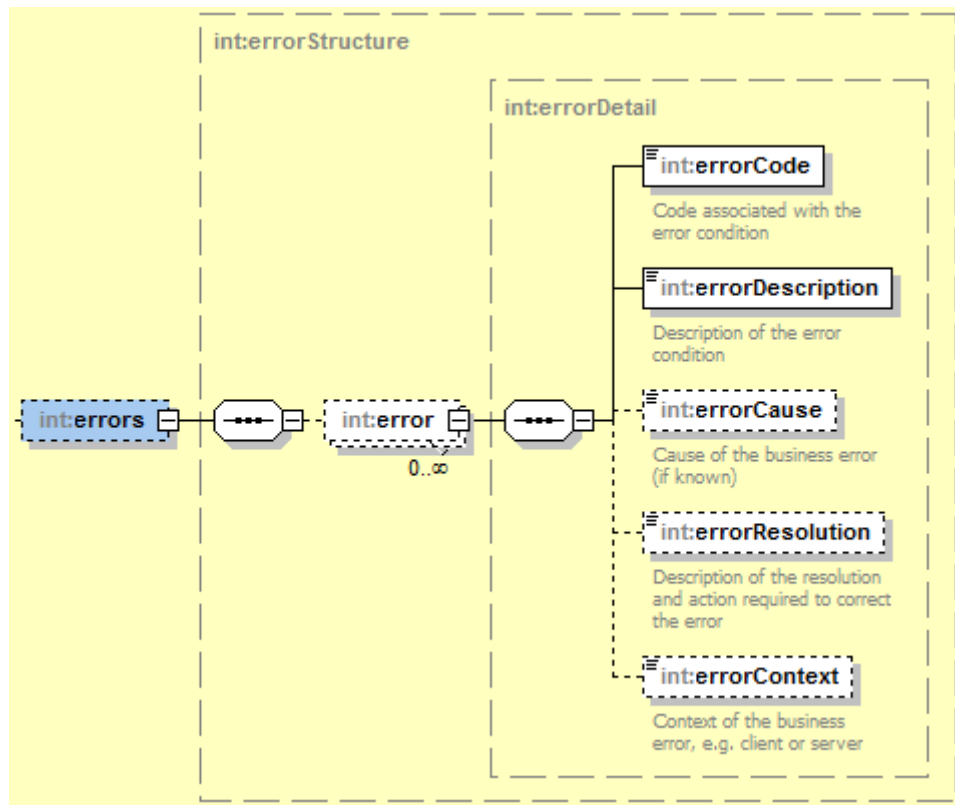


Figure 20 – integrationFooter errors Structure

All elements in the table below are relative to the errors element.

Element	Occurs	Data Type	Description
Error	0-∞	element	Element containing from zero to many errors.
error/errorCode	1-1	identifier	This is the code for the error message.
error/errorDescription	1-1	description	This is the description associated with the error code.
error/errorCause	0-1	description	This is the cause of the error (if known).
error/errorResolution	0-1	description	This is the description of the resolution and action required to correct the error (if known).
error/errorContext	0-1	description	This is the context of the error, e.g. client or server.

Table 24 – integrationFooter errors Element

warnings

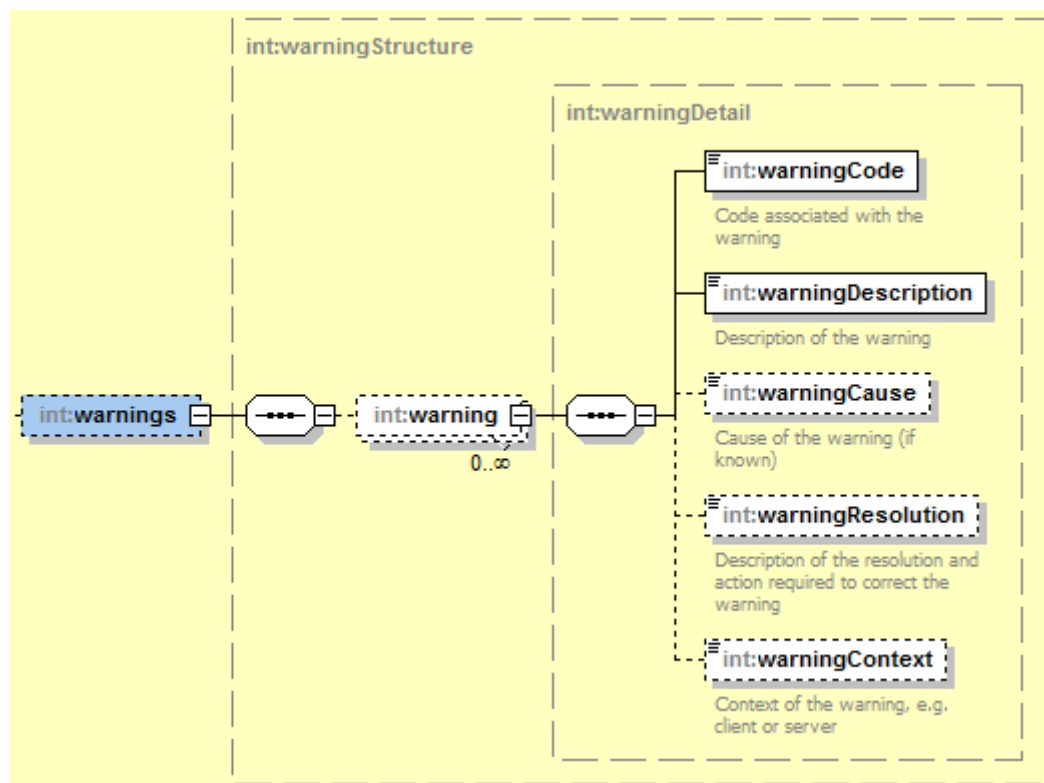


Figure 21 – integrationFooter warnings Structure

All elements in the table below are relative to the warnings element.

Element	Occurs	Data Type	Description
Warning	0-∞	element	Element containing from zero to many warnings.
warning/warningCode	1-1	identifier	This is the code for the warning message.
warning/warningDescription	1-1	description	This is the description associated with the warning code
warning/warningCause	0-1	description	This is the cause of the warning if known.
warning/warningResolution	0-1	description	This is the description of the resolution and action required to correct the warning if known.
warning/warningContext	0-1	description	This is the context of the warning, e.g. client or server.

Table 25 – integrationFooter warnings Element

8.6.4 Example Data

This section provides a simplified example to illustrate the createShipmentResponse (with a populated integrationFooter which contains warnings) without the SOAP wrappers. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```
<createShipmentResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T08:52:03</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>730222611</transactionId>
    </identification>
  </integrationHeader>
  <warnings>
    <warning>
      <warningCode>1</warningCode>
      <warningDescription>Warning description</warningDescription>
      <warningCause>Warning cause</warningCause>
      <warningResolution>Warning resolution</warningResolution>
      <warningContext>Warning context</warningContext>
    </warning>
  </warnings>
</createShipmentResponse>
```

```

</identification>
</integrationHeader>
<completedShipmentInfo>
  <status>
    <status xmlns="">
      <statusCode>
        <code>Allocated</code>
      </statusCode>
    </status>
    <validFrom xmlns="">2015-02-09T09:52:06.000+02:00</validFrom>
  </status>
  <allCompletedShipments>
    <completedShipments>
      <weight>
        <unitOfMeasure xmlns="">
          <unitOfMeasureCode>
            <code>g</code>
          </unitOfMeasureCode>
        </unitOfMeasure>
        <value xmlns="">100</value>
      </weight>
      <shipments>
        <shipmentNumber>HY188980152GB</shipmentNumber>
        <shipmentNumber>HY188980166GB</shipmentNumber>
        <shipment>
          <shipmentNumber>HY188980152GB</shipmentNumber>
          <itemID>1000076</itemID>
          <status>
            <status xmlns="">
              <statusCode>
                <code>Allocated</code>
              </statusCode>
            </status>
            <validFrom xmlns="">2015-02-09T09:52:06.000+02:00</validFrom>
          </status>
        </shipment>
        <shipment>
          <shipmentNumber>HY188980166GB</shipmentNumber>
          <itemID>1000077</itemID>
          <status>
            <status xmlns="">
              <statusCode>
                <code>Allocated</code>
              </statusCode>
            </status>
            <validFrom xmlns="">2015-02-09T09:52:06.000+02:00</validFrom>
          </status>
        </shipment>
      </shipments>
    </completedShipments>
  </allCompletedShipments>
  <requestedShipment>
    <shipmentType>
      <code xmlns="">Delivery</code>
    </shipmentType>
    <serviceOccurrence>1</serviceOccurrence>
    <serviceType>
      <code xmlns="">T</code>
    </serviceType>
    <serviceOffering>
      <serviceOfferingCode xmlns="">
        <code>TRM</code>
      </serviceOfferingCode>
    </serviceOffering>
    <serviceFormat>
      <serviceFormatCode xmlns="">
      </serviceFormatCode>
    </serviceFormat>
    <serviceEnhancements>
      <enhancementType>
        <serviceEnhancementCode xmlns="">
        </serviceEnhancementCode>
      </enhancementType>
    </serviceEnhancements>
    <shippingDate>2015-02-09</shippingDate>
    <recipientContact>
      <name>Mr Tom Smith</name>
      <complementaryName>Department 98</complementaryName>
      <telephoneNumber>
        <countryCode xmlns="">0044</countryCode>
        <telephoneNumber xmlns="">07801123456</telephoneNumber>
      </telephoneNumber>
      <electronicAddress>
        <electronicAddress xmlns="">tom.smith@royalmail.com</electronicAddress>
      </electronicAddress>
    </recipientContact>
    <recipientAddress>
      <addressLine1 xmlns="">44-46 Morningside Road</addressLine1>
    </recipientAddress>
  </requestedShipment>
</completedShipmentInfo>

```

```

<postTown xmlns="">Edinburgh</postTown>
<postcode xmlns="">EH10 4BF</postcode>
<country xmlns="">
  <countryCode>
    <code>GB</code>
  </countryCode>
</country>
</recipientAddress>
<items>
  <item>
    <numberOfItems>2</numberOfItems>
    <weight>
      <unitOfMeasure xmlns="">
        <unitOfMeasureCode>
          <code>g</code>
        </unitOfMeasureCode>
      </unitOfMeasure>
      <value xmlns="">100</value>
    </weight>
  </item>
</items>
<customerReference>CustSuppRef1</customerReference>
<senderReference>SenderReference1</senderReference>
</requestedShipment>
</completedShipmentInfo>
<integrationFooter>
  <warnings xmlns="http://www.royalmailgroup.com/integration/core/V1">
    <warning>
      <warningCode>W0042</warningCode>
      <warningDescription>Missing data - the Service Format is required has been omitted so a default value has been
used</warningDescription>
    </warning>
    <warning>
      <warningCode>W0036</warningCode>
      <warningDescription>E-mail option not selected so e-mail address will be ignored</warningDescription>
    </warning>
    <warning>
      <warningCode>W0035</warningCode>
      <warningDescription>SMS option not selected so Telephone Number will be ignored</warningDescription>
    </warning>
  </warnings>
</integrationFooter>
</createShipmentResponse>

```

8.7 updateShipmentRequest

The updateShipment operation allows customers to update the details of a previously created (but not manifested) shipment, provided that doing so does not violate the validation rules applied, or change the barcode number (e.g. it is not possible to update a Special Delivery item to become a Tracked Next Day item).

Only one shipment (identified by a single shipment number) can be updated per request, although multiple fields can be updated each time. If any field fails validation, then an error code will be returned and no fields will be updated. There is no limit to the number of times a shipment can be updated.

The status of the shipment is not affected by the update (e.g. a shipment with status 'Allocated' before an update will be 'Allocated' afterwards; a shipment with status 'Printed' will be 'Printed' afterwards).

To invoke the updateShipment operation, the customer shipping system constructs a SOAP request message (see section 7) with the updateShipmentRequest element contained within the SOAP Body. The updateShipmentRequest element is described in the figure and table below:

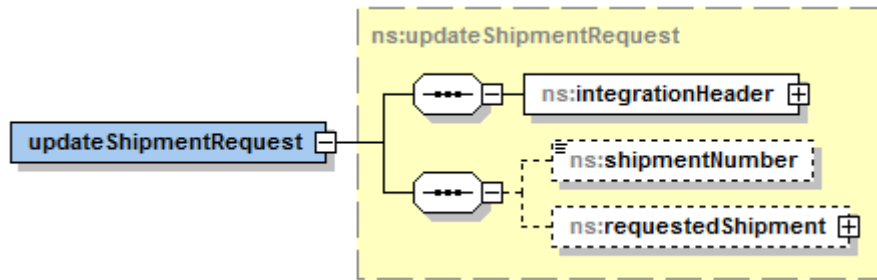


Figure 22 – updateShipmentRequest element Structure

All elements in the table below are relative to the updateShipmentRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.5.1
shipmentNumber	13	0-1	identifier	Mandatory. The shipment number to update which was returned/received in createShipmentResponse.
requestedShipment	N/A	0-1	element	The updated details for the identified shipment. This is described for the createShipment operation in Section 8.5.2.

Table 26 – updateShipmentRequest Element

NOTE: The barcode which was allocated during the earlier create shipment operation cannot be changed through the updateShipment operation and therefore the serviceType and serviceEnhancements elements cannot be altered. If changes to these elements are included in the requestedShipment element of the updateShipment, an error will be returned.

8.7.1 Example Data

This section provides a simplified example to illustrate the updateShipmentRequest without the SOAP wrappers and associated security elements. For full XML examples including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set

```

<v2:updateShipmentRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T09:27:58</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>163397663</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <!--Optional:-->
  <v2:shipmentNumber>RQ221150289GB</v2:shipmentNumber>
  <!--Optional:-->
  <v2:requestedShipment>
    <v2:recipientAddress>
      <addressLine1>1234 The Pyramids</addressLine1>
      <addressLine2>Valley of the Kings</addressLine2>
      <postTown>Cairo</postTown>
      <postcode>245678</postcode>
      <country>
        <countryCode>
          <code>EG</code>
        </countryCode>
      </country>
    </v2:recipientAddress>
  </v2:requestedShipment>
</v2:updateShipmentRequest>
  
```



```

</countryCode>
</country>
</v2:recipientAddress>
</v2:requestedShipment>
</v2:updateShipmentRequest>

```

8.8 updateShipmentResponse

The response for a requested updateShipment operation is constructed as a SOAP message with the updateShipmentResponse element contained in SOAP Body. The updateShipmentResponse element is described in the figure and table below:

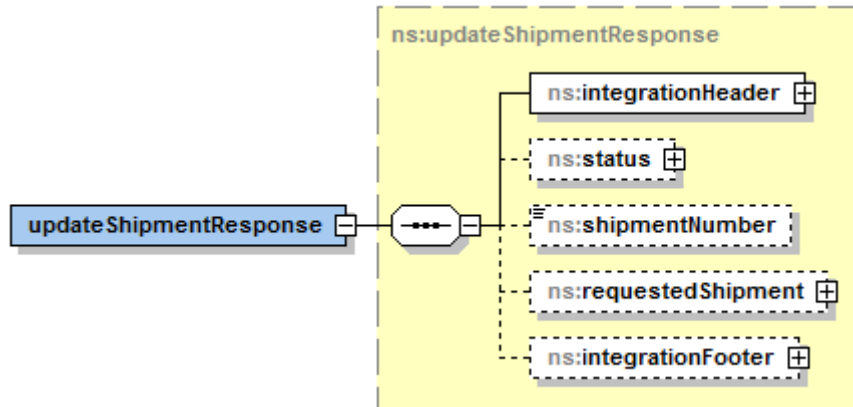


Figure 23 – updateShipmentResponse Structure

All elements in the table below are relative to the updateShipmentResponse element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	See section 8.6.1
status	N/A	1-1	element	Container for shipment status information
status/status	N/A	1-1	element	Container for shipment status details
../status/statusCode	N/A	1-1	element	Container for shipment status code
../status/statusCode/code	9	1-1	identifier	This will be set to current status of the updated shipment: 'Allocated' or 'Printed'. Please see section 12.2 for the full set of values shipment status can have.
../status/validFrom	N/A	1-1	dateTime	Datetime value associated with when the shipment status code is valid from.
../status/validTo	N/A	0-1	dateTime	NOT USED. Datetime value associated with when the shipment status code is valid to.
shipmentNumber	13	0-1	identifier	The shipment number that has been updated
requestedShipment	N/A	0-1	element	As described for the createShipment operation in section 8.5.2. Note that the barcode allocated to a shipment cannot be changed through the updateShipment operation and therefore the serviceType and serviceEnhancements elements cannot be altered by the updateShipment operation. If changes to these elements are included in the requestedShipment element of the updateShipment, an error will be returned.
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the updateShipment operation. Please see section 8.6.3 for more information.

Table 27 – updateShipmentResponse Element

8.8.1 Example Data

This section provides a simplified example to illustrate the updateShipmentResponse without the SOAP wrappers and WS-security elements. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```
<updateShipmentResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T09:27:58</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>163397663</transactionId>
    </identification>
  </integrationHeader>
  <status>
    <status xmlns="">
      <statusCode>
        <code>Allocated</code>
      </statusCode>
    </status>
    <validFrom xmlns="">2015-02-09T10:27:59.000+02:00</validFrom>
  </status>
  <shipmentNumber>RQ221150275GB</shipmentNumber>
  <requestedShipment>
    <recipientContact>
      <name/>
      <complementaryName/>
      <electronicAddress>
        <electronicAddress xmlns=""/>
      </electronicAddress>
    </recipientContact>
    <recipientAddress>
      <buildingName xmlns=""/>
      <buildingNumber xmlns="">0</buildingNumber>
      <addressLine1 xmlns="">1234 The Pyramids</addressLine1>
      <addressLine2 xmlns="">Valley of the Kings</addressLine2>
      <addressLine3 xmlns=""/>
      <postTown xmlns="">Cairo</postTown>
      <postcode xmlns="">245678</postcode>
      <country xmlns="">
        <countryCode>
          <code>EG</code>
        </countryCode>
      </country>
    </recipientAddress>
    <departmentReference/>
    <customerReference/>
    <senderReference/>
    <safePlace/>
    <importerContact>
      <name/>
      <complementaryName/>
      <electronicAddress>
        <electronicAddress xmlns=""/>
      </electronicAddress>
    </importerContact>
    <importerAddress>
      <domesticIndicator xmlns="">false</domesticIndicator>
      <buildingName xmlns=""/>
      <buildingNumber xmlns="">0</buildingNumber>
      <addressLine1 xmlns=""/>
      <addressLine2 xmlns=""/>
      <addressLine3 xmlns=""/>
      <addressLine4 xmlns=""/>
      <stateOrProvince xmlns="">
        <stateOrProvinceCode>
          <code/>
        </stateOrProvinceCode>
      </stateOrProvince>
      <postTown xmlns=""/>
      <county xmlns="">
        <countyCode>
          <code/>
        </countyCode>
      </county>
      <postcode xmlns=""/>
    </importerAddress>
  </requestedShipment>
</updateShipmentResponse>
```

```

<country xmlns="">
  <countryCode>
    <code/>
  </countryCode>
</country>
</importerAddress>
<exporterContact>
  <name/>
  <complementaryName/>
  <electronicAddress>
    <electronicAddress xmlns=""/>
  </electronicAddress>
</exporterContact>
<exporterAddress>
  <domesticIndicator xmlns="">false</domesticIndicator>
  <buildingName xmlns=""/>
  <buildingNumber xmlns="">0</buildingNumber>
  <addressLine1 xmlns=""/>
  <addressLine2 xmlns=""/>
  <addressLine3 xmlns=""/>
  <addressLine4 xmlns=""/>
  <stateOrProvince xmlns="">
    <stateOrProvinceCode>
      <code/>
    </stateOrProvinceCode>
  </stateOrProvince>
  <postTown xmlns=""/>
  <country xmlns="">
    <countryCode>
      <code/>
    </countryCode>
  </country>
  <postcode xmlns=""/>
  <country xmlns="">
    <countryCode>
      <code/>
    </countryCode>
  </country>
</exporterAddress>
<internationalInfo>
  <parcels>
    <parcel>
      <purposeOfShipment>
        <code xmlns=""/>
      </purposeOfShipment>
      <explanation/>
      <invoiceNumber/>
      <exportLicenseNumber/>
      <certificateNumber/>
      <fees>0</fees>
    </parcel>
  </parcels>
  <shipperExporterVatNo/>
  <recipientImporterVatNo/>
  <originalExportShipmentNo/>
  <documentsOnly>false</documentsOnly>
  <documentsDescription/>
  <shipmentDescription/>
  <comments/>
  <termsOfDelivery/>
  <purchaseOrderRef/>
</internationalInfo>
</requestedShipment>
<integrationFooter>
  <warnings xmlns="http://www.royalmailgroup.com/integration/core/V1">
    <warning>
      <warningCode>W0020</warningCode>
      <warningDescription>signature is not a valid option for the service offering selected and will be ignored. If a signature is required
cancel this shipment and re-raise specifying a valid Service Offering</warningDescription>
    </warning>
  </warnings>
</integrationFooter>
</updateShipmentResponse>

```

8.9 cancelShipmentRequest

The cancelShipment operation allows customers to update the details of a previously created (but not manifested) shipment. Once a shipment has been cancelled its status will change from 'Allocated' or 'Printed' to 'Cancelled'.

Shipments created by the API can be cancelled by the API call or the DMO GUI, and cancelled shipments are visible in the system. Any Tracked Returns shipments must be cancelled before midnight as this is when they will be automatically processed and archived by the system. Up to 1,000 shipments can be cancelled in a single request. Any shipments that can't be cancelled will be communicated as an error message in the integrationFooter as described in section 8.6.3.

To invoke the cancelShipment operation, the customer shipping system constructs a SOAP message (see section 7) with the cancelShipmentRequest element contained within the SOAP Body. The cancelShipmentRequest element is described in the figure and table below:

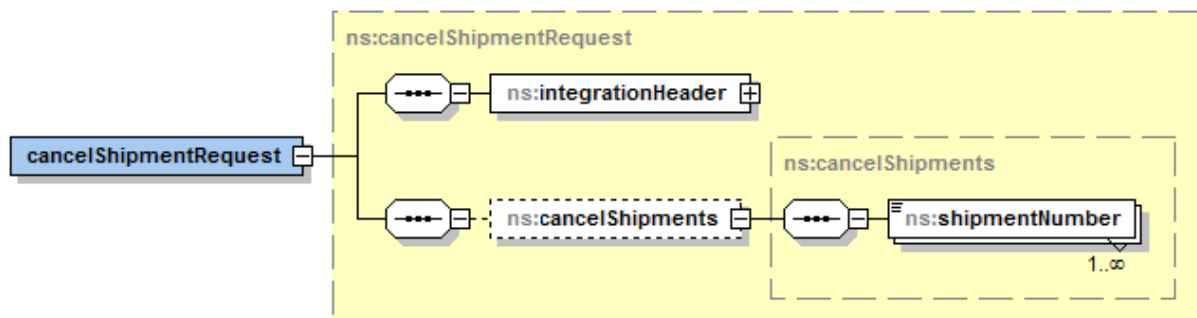


Figure 24 – cancelShipmentRequest Structure

All elements in the table below are relative to the cancelShipmentRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	Element	As described in section 8.5.1
cancelShipments	13	0-1	Element	Container for the list of shipments to cancel
cancelShipments/shipmentNumber	13	1- ∞	Identifier	The shipment number to be cancelled. The shipment no(s) should be same as that was received/returned in create shipment response.

Table 28 – cancelShipmentRequest Element

8.9.1 Example Data

The section below only shows a cancelShipmentRequest SOAP operation as an example. For full XML examples of other SOAP operations including requests and responses please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```

<v2:cancelShipmentRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T09:35:28</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>420642961</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:cancelShipments>
    <v2:shipmentNumber>RQ221150289GB</v2:shipmentNumber>
  </v2:cancelShipments>
</v2:cancelShipmentRequest>
  
```

8.10 cancelShipmentResponse

The response for a requested cancelShipment operation is constructed as a SOAP message described (see section 7) with the cancelShipmentResponse element contained within the SOAP Body. The cancelShipmentResponse element is described in the figure and table below:

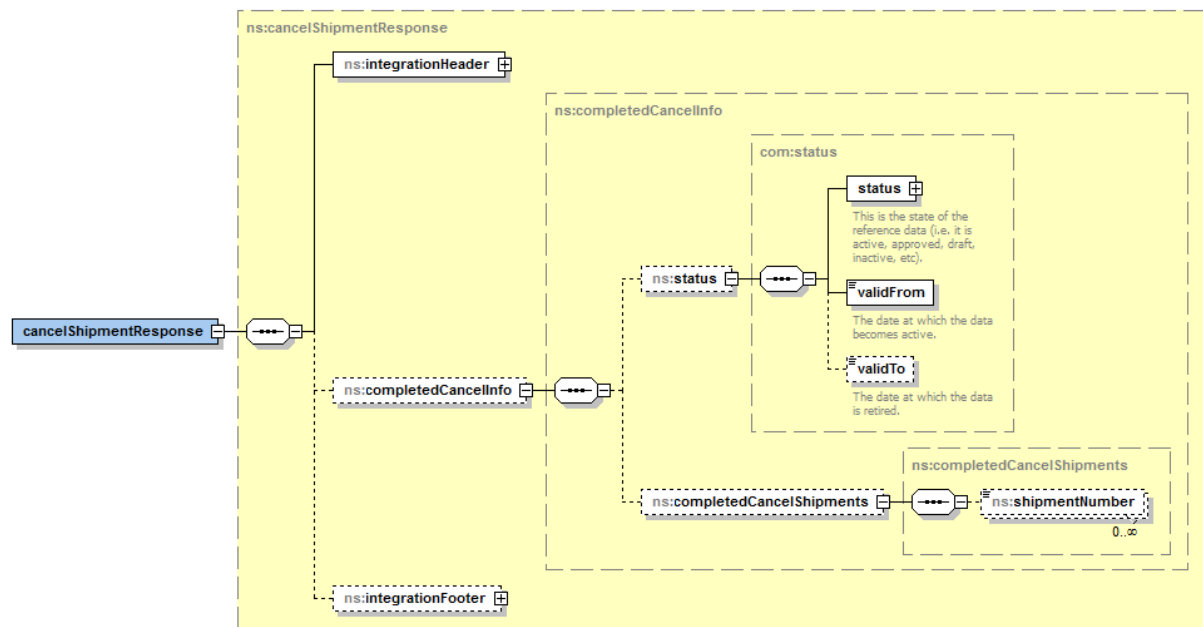


Figure 25 – cancelShipmentResponse Structure

All elements in the table below are relative to the cancelShipmentRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.6.1
completedCancelInfo	N/A	0-1	element	Container for the set of successfully cancelled shipments
completedCancelInfo/status	N/A	1-1	element	Container for shipment status information
../status/status	N/A	1-1	element	Container for shipment status details
../status/status/statusCode	N/A	1-1	element	Container for shipment status code
../status/status/statusCode/code	9	1-1	identifier	This will be set to 'Cancelled'. Please see section 12.2 for the full set of values shipment status can have.
../status/status/validFrom	N/A	1-1	dateTime	Datetime value associated with when the shipment status code is valid from.
../status/status/validTo	N/A	0-1	dateTime	NOT USED. Datetime value associated with when the shipment status code is valid to.
completedCancelInfo/completedCancelShipments	N/A	0-1	element	Container for cancelled shipment numbers
../completedCancelShipments/shipmentNumber	13	0- ∞	identifier	The number of each shipment that has been successfully cancelled.
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the cancelShipment operation. Please see section 8.6.3 for more information.

Table 29 – cancelShipmentResponse Element

8.10.1 Example Data

The section below only shows a cancelShipmentResponse SOAP operation as an example. For full XML examples of other SOAP operations including requests and responses please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```
<cancelShipmentResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T09:35:28</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>420642961</transactionId>
    </identification>
  </integrationHeader>
  <completedCancelInfo>
    <status>
      <status xmlns="">
        <statusCode>
          <code>Cancelled</code>
        </statusCode>
      </status>
      <validFrom xmlns="">2015-02-09T10:35:28.000+02:00</validFrom>
    </status>
    <completedCancelShipments>
      <shipmentNumber>RQ221150275GB</shipmentNumber>
    </completedCancelShipments>
  </completedCancelInfo>
</cancelShipmentResponse>
```

8.11 printLabelRequest

The printLabel operation has several functions.

- It allows customers to request a label in Base64 encoded PDF format for a specific shipment.
- It returns shipment data in XML format for production of custom labels.
- Lastly, it allows the overriding of an English address on the label with a localised version of the same address.

Once the printLabel operation has been called on a shipment with status 'Allocated' or 'Allocated Offline', the status for that shipment is changed to 'Printed' or 'Printed Offline'.

As mentioned in above section, if you use the Status Code within the OfflineShipments element in the createShipment request as 'PrintedOffline', you do not need to use the printLabel request to add the item to the manifest / End of Day process.

Shipments created by either the GUI or API can be printed by the API call. There is no limit on the number of times the printLabel request can be used on a shipment, though high numbers of reprints will be flagged to Royal Mail.

Overriding the English address on the label will allow you to use characters beyond the allowable character set, only for the purpose of printing a label in the native language of the destination of the shipment.

To invoke the printLabel operation, your shipping system constructs a SOAP message (see section 7) with the printLabelRequest element contained within the SOAP Body. The printLabelRequest element is described in the figure and table below:

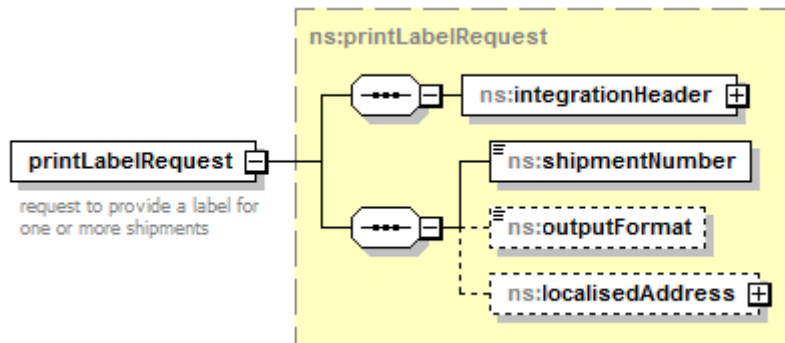


Figure 26 – printLabelRequest Structure

All elements in the table below are relative to the printLabelRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.5.1
shipmentNumber	13	1-1	identifier	The shipment number of the shipment to be printed
outputFormat	5	0-1	identified	The content of response. Unspecified / PDF: returns the standard Base64 Encoded PDF Label DS : returns a data stream (See response. DSPDF : returns both the data stream and the Base64 Encoded PDF Label. PNG : returns Base64 Encoded PNG images of the 2D Data Matrix and 1D Linear Barcode. DSPNG : returns both the data stream and the Base64 Encoded PNG images of the 2D Data Matrix and 1D Linear Barcode.
localisedAddress	N/A	0-1	Element	Optional. This is a complex type.

Table 30 – printLabelRequest Element

NOTE: The outputFormat element in the above table by default is set to PDF format for you. To enable other format options e.g. DS or PNG, you should request the Customer Solutions team to configure all other options under their API login account.

localisedAddress

Localised address allows you to use foreign characters on the label, typically for non-English recipient names or non UK delivery addresses.

Element	Max Length	Occurs	Data Type	Description
recipientContact	N/A	0-1	Element	Complex element to hold localised version of the name / business name
recipientContact/name	80	0-1	longName	Localised Contact name i.e. Contact name in local or destination address character or language

Element	Max Length	Occurs	Data Type	Description
recipientContact/complementaryName	64	0-1	longName	Localised Business name
recipientAddress	N/A	0-1	Element	Complex element to hold localised version of the address
recipientAddress/addressLine1	80 (35)	0-1	Description	First line of the address
recipientAddress/addressLine2	80 (35)	0-1	Description	Second line of the address
recipientAddress/addressLine3	80 (35)	0-1	Description	Third line of the address
recipientAddress/postTown	40 (35)	0-1	Name	Destination Town or City
recipientAddress/postcode	15	0-1	Identifier	Destination postcode/area code/pin code.

Table 31 - localisedAddress Element

Note: If you specify more than 35 characters in addressLine1, addressLine2, addressLine3 or PostTown fields and generate a PDF label, these fields will be truncated to 35 characters in order to fit onto the label. It is therefore recommended that customers enter values within the 35 character limit.

8.11.1 Example Data

This section provides simplified examples to illustrate the printLabelRequest without the SOAP wrappers and WS-Security elements. For a full XML example with other output format and other SOAP operations, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

PrintLabelRequest: Default label format

```

<v2:printLabelRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-08-05T15:45:00Z</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>0000000000</v1:applicationId>
      <v1:transactionId>0000000002</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:shipmentNumber>TTT000441351GB</v2:shipmentNumber>
  <v2:outputFormat>PDF</v2:outputFormat>
</v2:printLabelRequest>

```


PrintLabelRequest: DSPNG label format

```
<soapenv:Body>
  <v2:printLabelRequest>
    <v2:integrationHeader>
      <v1:dateTime>2015-08-05T15:45:00Z</v1:dateTime>
      <v1:version>2</v1:version>
      <v1:identification>
        <v1:applicationId>0000000000</v1:applicationId>
        <v1:transactionId>0000000002</v1:transactionId>
      </v1:identification>
    </v2:integrationHeader>
    <v2:shipmentNumber>AAA382597271GB</v2:shipmentNumber>
    <!--DSPNG provides images of barcodes only-->
    <v2:outputFormat>DSPNG</v2:outputFormat>
  </v2:printLabelRequest>
```

PrintLabelRequest: Localised Address

```
<v2:printLabelRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T09:06:41</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>933025655</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:shipmentNumber>RQ221150275GB</v2:shipmentNumber>
  <v2:outputFormat>DSPNG</v2:outputFormat>
  <v2:localisedAddress>
    <v2:recipientContact>
      <v2:name>كثيري نوج</v2:name>
      <v2:complementaryName>ديربل ايرصملا</v2:complementaryName>
    </v2:recipientContact>
    <v2:recipientAddress>
      <addressLine1>1</addressLine1>
      <addressLine2>كولملا يداو</addressLine2>
      <postTown>قراقل</postTown>
      <postcode>245678</postcode>
    </v2:recipientAddress>
  </v2:localisedAddress>
</v2:printLabelRequest>
```

8.12 printLabelResponse

The response for a requested printLabel operation is constructed as a SOAP message described (see section 7) with the printLabelResponse element contained within the SOAP Body. The printLabelResponse element is described in the figure and table below.

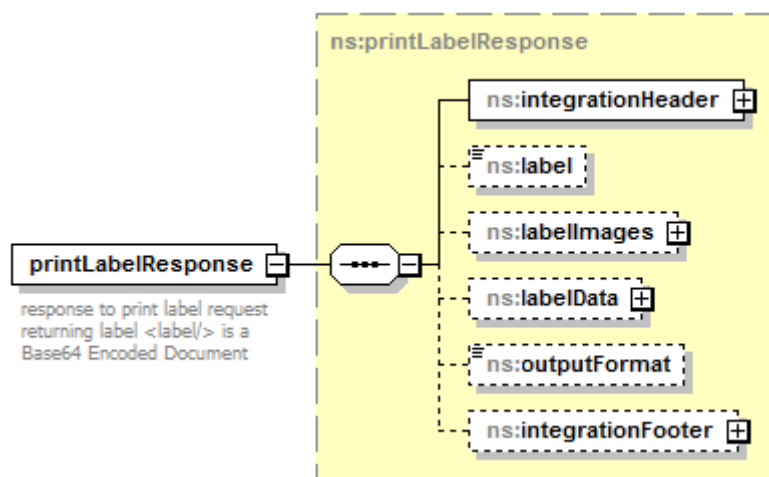


Figure 27 – printLabelResponse Structure

All elements in the table below are relative to the `printLabelResponse` element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.6.1
label	N/A	0-1	document	Label in PDF format and Base64 encoded
labelImages	N/A	0-1	element	Container for Base64 Encoded PNG Images
labelImages/image1D Barcode	N/A	1-1	document	Base64 Encoded PNG Image of the 1D Linear Barcode n.b. This Tag will be empty for non-barcode services
labelImages/image2D Matrix	N/A	1-1	document	Base64 Encoded PNG Image of the 2D Data Matrix
labelData	N/A	0-1	element	Optional. This is a complex type and is detailed below
outputFormat	5	0-1	identified	The format of the response as requested in the <code>printLabelRequest</code> i.e. PDF/DS/DSPDF/PNG/DSPNG
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the <code>printLabel</code> operation. Please see section 8.6.3 for more information.

Table 32 – printLabelResponse Element

labelData

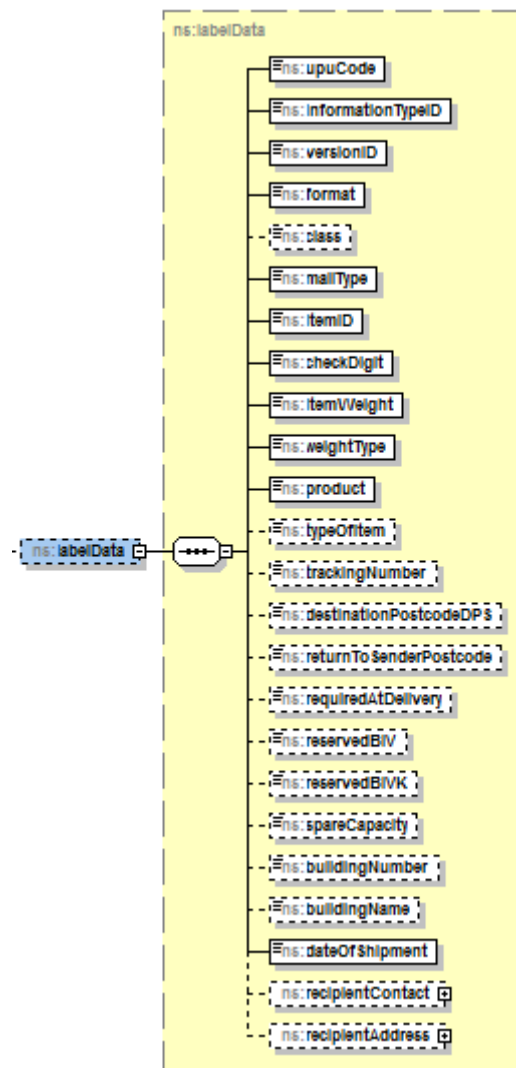


Figure 28 – labelData Structure

All elements in the table below are relative to the labelData element.

Element	Max Length	Occurs	Data Type	Description
upuCode	4	0-1	identifier	Always "JGB"
informationTypeID	1	0-1	identifier	Always 6
versionID	1	0-1	integer	Always 1
format	2	0-1	identifier	Dependant on Service Format selected
mailType	1	0-1	identifier	Depends on what service was selected e.g. Inland Large Letter
itemID	8	0-1	identifier	Unique Item ID for shipment which is same as received in response to creatshipmentRequest. If customer is not using the 2D barcode returned by the API, then they must convert this number to hex and then use this hex number in the 2D barcode.
checkDigit	1	0-1	checkDigit	System Calculated value
itemWeight	7	0-1	integer	Shipment/Item Weight
weightType	1	0-1	identifier	Dependant on Service selected i.e. g, kg etc.
product	5	0-1	identifier	Dependant on Service Reference selected

Element	Max Length	Occurs	Data Type	Description
typeOfItem	N/A	0-1	identifier	Not Used
trackingNumber	13	0-1	identifier	1D Linear Barcode for shipment which is the same as shipment number received in response to createShipmentRequest
destinationPostcodeDPS	2	0-1	identifier	The post/pin code for the destination.
returnToSenderPostcode	9	0-1	identifier	Post/Pin code for the sender for returns
requiredAtDelivery	1	0-1	identifier	If a signature is required, the value will be S, otherwise blank.
reservedBIV	N/A	0-1	identifier	Not Used
reservedBIVK	N/A	0-1	identifier	Not Used
spareCapacity	N/A	0-1	identifier	Not Used
buildingNumber	4	0-1	integer	Building Number (Label)
buildingName	35	0-1	name	Building Name (Label)
dateOfShipment	6	0-1	date	Provisional Date Of Shipment
recipientContact	N/A	0-1	element	Destination Recipient Contact Details (Label)

Table 33 – labelData Element

8.12.1 Example Data

This section provides couple of simplified examples to illustrate the printLabelResponse without the SOAP wrappers and WS-Security elements. For a full XML example with other output format and other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

PrintLabelResponse: Default label format

```
<printLabelResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T09:03:19</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>467089462</transactionId>
    </identification>
  </integrationHeader>
  <label>JVBERi0xLjYKJeTjz9IKMSAwIG9iagpbL1BERi9JbWFnZUIvSW1hZ2VZDL0ltYWdlSS9UZXBhXQpl
bmRvYmoKNCAlG9iag08PC9MZW5ndGggNSAwIFIKL0ZpbHRlci9GbGF0ZURlY29kZQo+PgpzdHJl
YW0KeJwDAAAAAAEKZW5kc3RyZWFtCmVuZG9iag01IDAgb2JqCjgKZW5kb2JqCjYgMGBvYmoKPDwv
+fr/xAAfAQADAQEBAQEBAQEBAQAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncA
AQIDEQQFITEGEkFRB2FxEylygQgUQpGhscEJlZNS8BVicEKFiQ04SXxXgZGiYnKCkqNTY3ODk6
Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeliYqSk5SVlpeYmZqio6Slpqeo
MCBSCi9JbmZvIDAgUgovSURbPDZDM0VCNEREOEE2OTNEMTVDQUE4NkRCODJCNTc2MTIzPjw2
QzNFQjRERDhBNjkzRDE1Q0FBODZEQjgyQjU3NjEyMz5dCj4+CnN0YXJ0eHJlZgoxMz1OTYKJSVF
T0YK</label>
  <outputFormat>PDF</outputFormat>
</printLabelResponse>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <printLabelResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
      <integrationHeader>
        <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-07-20T15:45:00Z</dateTime>
        <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
        <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
          <applicationId>0000000000</applicationId>
          <transactionId>0000000001</transactionId>
        </identification>
      </integrationHeader>
      <labelImages>
        <image1DBarcode>iVBORw0KGgoAAAANSUHEUgAAAFUAAABvAgMAAACIrLETAADFBMVEXMzMz///8AAD/AADq3Ybv
        AAAACXBIWXAAB7pAAAE6QF1y32iAAAACXZwQWcAAAH1AAAAbwAAUn0l/AAAAjUIEQVRo3u3NUQ3A
        MAgFwJlkH5UwFTNRCf0YKgergQm4hJAWXrgj1xvJRlZPyCei+plZ3zNn5qxVz0dWrHp09bC2Feu+
        olfRvSY7XIHswOyDu0bfjH38e3/5+8Dj8Xg8Ho/H4/F4PB6Px+PxeDwej8fj8Xg8Ho/H4/F4PB6P
        x+PxeDwej8fj8Xg8/hf/AgOtTbVIW17PAAAAJXRfWVRkYXRlOmNyZWZ0ZQAyMDE1LTA4LTlwVDEz
        OjMwOjAyKzAyOjAwaNwXVgAAACV0RvH0ZGF0ZTptb2RpZnZkAMjAxNS0wOC0yMFQxMzozMDowMisw
        MjowMBmBr+oAAAAASUVORK5CYII=</image1DBarcode>

        <image2DMatrix>iVBORw0KGgoAAAANSUHEUgAAABQAAAAUAQAAAAACi8iCgAAAAAmJLR0QAAd2KE6QAAAAJcEhZCwA
        A
        HukAAB7pAXXLfalAAAAJdnBBZwAAABQAAAAUAUKM7KiEAAAA1SURBVjAxyZa2NmAAyfvzDGDmYP8B
        jI0PHABj5jMfwJjnDAMYGxt8AGObwwgzPzfAlYhAACHQRVDZcDogAAACV0RvH0ZGF0ZTpcjcmVh
        dGUAMjAxNS0wOC0yMFQxMzozMDowMyswMjowMM6rHOIAAAAlDEVYdGRhdGU6bW9kaWZ55ADlwMTUt
        MDgtMjBjUMTM6MzA6MDMrMDI6MDC/9qReAAAAAEIFTkSuQmCC</image2DMatrix>

        </labelImages>
        <labelData>
          <upuCode>JGB</upuCode>
          <informationTypeID>6</informationTypeID>
          <versionID>1</versionID>
          <format>P</format>
          <mailType>Inland Parcel</mailType>
          <itemID>459</itemID>
          <checkDigit>3</checkDigit>
          <itemWeight>250</itemWeight>
          <weightType>0</weightType>
          <product>CRL_1</product>
          <trackingNumber>TTT000527313GB</trackingNumber>
          <destinationPostcodeDPS>YT6 1BB</destinationPostcodeDPS>
          <returnToSenderPostcode>LU3 1SY</returnToSenderPostcode>
          <buildingNumber>13</buildingNumber>
          <buildingName/>
          <dateOfShipment>2015-08-20</dateOfShipment>
          <recipientContact>
            <name>TEST</name>
            <complementaryName>SHIPMENT</complementaryName>
            <telephoneNumber>
              <telephoneNumber xmlns="">0044xxxxxxxx</telephoneNumber>
            </telephoneNumber>
            <electronicAddress>
              <electronicAddress xmlns="">name@email.com</electronicAddress>
            </electronicAddress>
          </recipientContact>
          <recipientAddress>
            <buildingName xmlns=""/>
            <buildingNumber xmlns="">13</buildingNumber>
            <addressLine1 xmlns="">ADDRESSLINE 1</addressLine1>
            <addressLine2 xmlns="">ADDRESSLINE 2</addressLine2>
            <addressLine3 xmlns="">ADDRESSLINE 3</addressLine3>
            <stateOrProvince xmlns="">
              <stateOrProvinceCode>
                <code/>
              </stateOrProvinceCode>
            </stateOrProvince>
            <postTown xmlns="">LONDON</postTown>
            <county xmlns="">
              <countyCode>
                <code/>
              </countyCode>
            </county>
          </recipientAddress>
        </labelData>
      </printLabelResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
</xml>
```

```

<postcode xmlns=""/>
<country xmlns="">
  <countryCode>
    <code>GB</code>
  </countryCode>
</country>
</recipientAddress>
</labelData>
<outputFormat>DSPNG</outputFormat>
</printLabelResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

8.13 createManifestRequest

The createManifest operation allows you to submit to Royal Mail details of all of the items that will be despatched on the current date. Refer to the createShipment call, shippingDate element to createShipments for future dates and the updateShipment call, shippingDate element to alter the shipment date or the cancelShipment call to exclude shipments from the manifest. Once the createManifest operation has been called, all shipments that have status 'Printed' will be set to status 'Manifested' and it will no longer be possible to update or cancel them.

Manifests can be created by Service Reference or by Service Code, or if neither is specified then all shipments that have status 'Printed' will be included (N.B. Tracked Returns are not included in any part of the manifesting process).

The operation creates the PDF manifest in an asynchronous operation and may not be immediately available for printing using the printManifest operation.

To invoke the createManifest operation, your shipping system constructs a SOAP message (see section 7) with the createManifestRequest element contained within the SOAP Body.

The createManifestRequest element is described in the figure and table below:

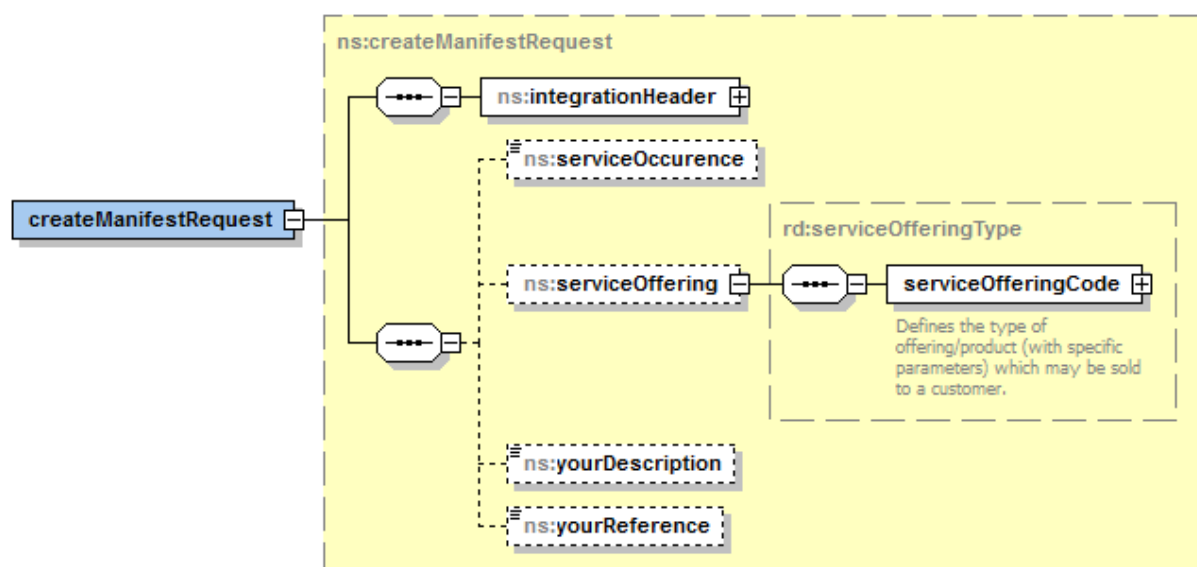


Figure 29 – createManifestRequest element Structure

All elements in the table below are relative to the createManifestRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.5.1
serviceOccurrence	2	0-1	ordinal	All or one Service Occurrence. If not included then ALL Service Occurrences are included in the manifest.
serviceOffering	N/A	0-1	element	Container for service offering details
serviceOffering/ serviceOfferingCode	N/A	1-1	element	Container for service offering code
../serviceOfferingCode/code	3	0-1	identifier	All or one Service Offerings. If not included then ALL Service Offerings are included in the manifest. Please note that this field is case sensitive.
yourDescription	40	0-1	comment	This is a description field that corresponds to the Your Description field in RM's Online Business Account (OBA). This is for customer reference and will not appear on any paperwork.
yourReference	40	0-1	identifier	Included on the Customer Collection Receipt.

Table 34 – createManifestRequest Element

NOTE: If Average Weight End of Day option is turned ON then for average weighted products, manifesting should be done manually from DMO as average weighted products cannot be manifested by API.

8.13.1 Example Data

This section provides a simplified example to illustrate the createManifestRequest without the SOAP wrappers and WS-Security elements. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```

<v2:createManifestRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T09:16:56</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>102122313</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:yourDescription>ManifestDesc</v2:yourDescription>
  <!--Optional:-->
  <v2:yourReference>ManifestRef</v2:yourReference>
</v2:createManifestRequest>

```

8.14 createManifestResponse

The response for a requested createManifest operation is constructed as a SOAP message (see section 7) with the createManifestResponse element contained within SOAP Body. The createManifestResponse element is described in the figure and table below:

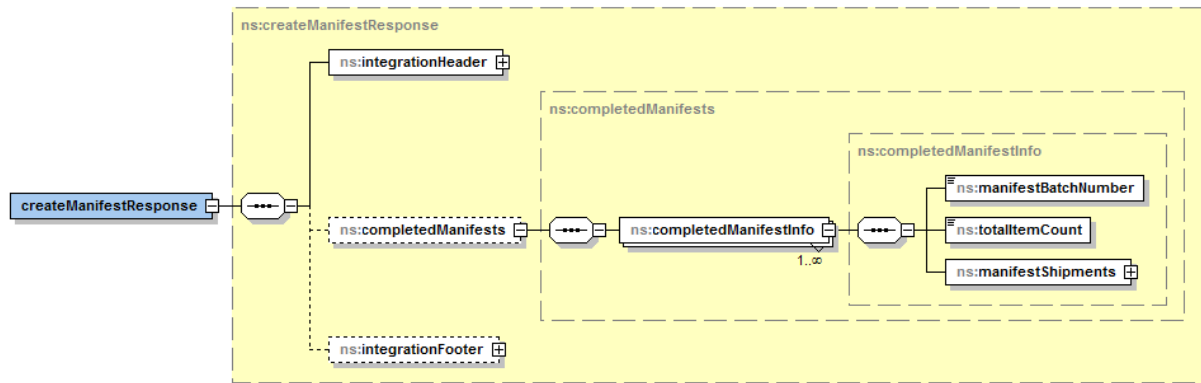


Figure 30 – createManifestResponse Structure

All elements in the table below are relative to the createManifestResponse element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.6.1
completedManifests	N/A	0-1	element	Container for completed manifest details
completedManifests/ completedManifestInfo	N/A	1- ∞	element	Container for information on the manifest that was created
../completedManifestInfo/ manifestBatchNumber	4	1-1	identifier	The batch number of the manifest. It is a sequentially allocated number. Used in subsequent call to the printManifest operation.
../completedManifestInfo/ totalItemCount	2	1-1	cardinal	Total number of shipments on this manifest
../completedManifestInfo/ manifestShipments	N/A	1-1	element	Container for the list of shipments included in the manifest
../../manifestShipments/ manifestShipment	N/A	1- ∞	element	Container for manifest shipment information
../../manifestShipment/serviceOffering	N/A	1-1	element	Container for service offering details
../../manifestShipment/serviceOffering/ serviceOfferingCode	N/A	1-1	element	Container for service offering code
../../manifestShipment/serviceOffering/ serviceOfferingCode/code	N/A	1-1	identifier	The service offering code of a shipment on the manifest. Please note that this field is case sensitive.
../../manifestShipments/ manifestShipment/shipmentNumber	13	1-1	identifier	The 1D shipment number of a shipment on the manifest
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the createManifest operation. Please see section 8.6.3 for more information.

Table 35 – createManifestResponse Element

8.14.1 Example Data

This section provides a simplified example to illustrate the createManifestResponse without the SOAP wrappers and WS Security. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.


```

<createManifestResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T09:16:56</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>102122313</transactionId>
    </identification>
  </integrationHeader>
  <completedManifests>
    <completedManifestInfo>
      <manifestBatchNumber>81</manifestBatchNumber>
      <totalItemCount>2</totalItemCount>
      <manifestShipments>
        <manifestShipment>
          <serviceOffering>
            <serviceOfferingCode xmlns="">
              <code>MP6</code>
            </serviceOfferingCode>
          </serviceOffering>
          <shipmentNumber>RQ221150275GB</shipmentNumber>
        </manifestShipment>
        <manifestShipment>
          <serviceOffering>
            <serviceOfferingCode xmlns="">
              <code>TRM</code>
            </serviceOfferingCode>
          </serviceOffering>
          <shipmentNumber>HY188980152GB</shipmentNumber>
        </manifestShipment>
      </manifestShipments>
    </completedManifestInfo>
  </completedManifests>
</createManifestResponse>

```

8.15 printManifestRequest

The printManifest operation allows you to request a manifest (Customer Collection Receipt) in Base64 encoded PDF format. Once the printManifest operation has been called, all shipments that have status 'Manifested' will be set to status 'ManifestedPrinted'. Manifests can be specified by either the manifestBatchNumber or the salesOrderNumber (the salesOrderNumbers are available via the GUI the day after the manifest was created).

There is no limit on the number of times the printManifest request can be used on a manifest, however reprints will not include the barcodes and excessive requests for reprints will be flagged to Royal Mail.

As the createManifest operation is asynchronous, the manifest may not be immediately available for printing. An error message will be returned if this is the case.

To invoke the printManifest operation, your shipping system constructs a SOAP message (see section 7) with the printManifestRequest element contained within the SOAP Body. The printManifestRequest element is described in the figure and table below:

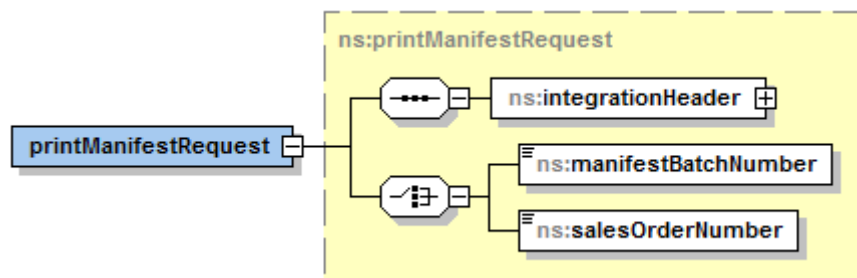


Figure 31 – printManifestRequest element Structure

All elements in the table below are relative to the printManifestRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.5.1
manifestBatchNumber	20	1-1	identifier	This is the batch number to print and is returned by a prior call to createManifest operation.
salesOrderNumber	20	1-1	identifier	The Sales Order Number, which is available via the GUI the day after the manifest was created.

Table 36 – printManifestRequest Element

8.15.1 Example Data

This section provides a simplified example to illustrate the printManifestRequest without the SOAP wrappers and WS-Security elements. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```

<v2:printManifestRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T09:19:19</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>335379752</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:manifestBatchNumber>79</v2:manifestBatchNumber>
</v2:printManifestRequest>

```

8.16 printManifestResponse

The response for a requested printManifest operation is constructed as a SOAP message (see section 7) with the printManifestResponse element contained within the SOAP Body. The printManifestResponse element is described in the figure and table below:

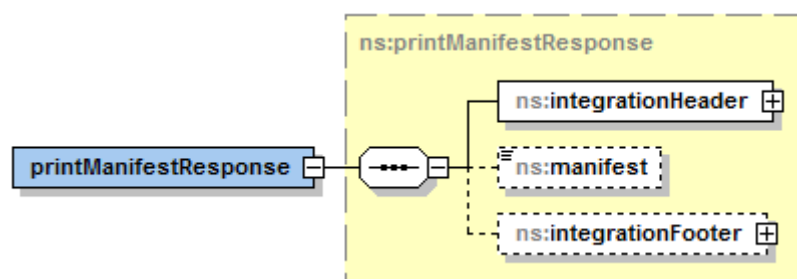


Figure 32 – printManifestResponse Structure

All elements in the table below are relative to the printManifestResponse element.

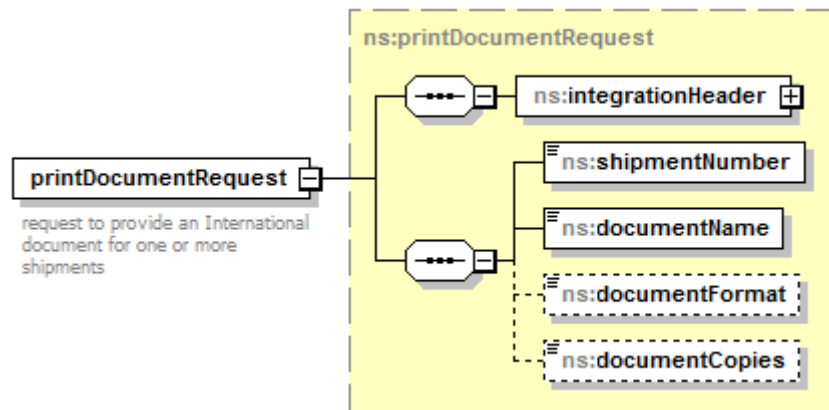


Figure 33 – printDocumentRequest element Structure

All elements in the table below are relative to the printDocumentRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.5.1
shipmentNumber	13	1-1	identifier	Mandatory. The number of the shipment to document.
documentName	4	1-1	identifier	Mandatory. The name of the document to output. Valid values are: CN22, CN23 and CI (for Commercial Invoice).
documentCopies	1	0-1	identifier	Number of copies of the International Document within the single Base64 Encoded PDF document output. Valid values: 1 or 3 3 for Commercial Invoice Only.

Table 38 – printDocumentRequest Element

8.17.1 Example Data

This section provides a simplified example to illustrate the printDocumentRequest without the SOAP wrappers and WS-Security elements. For a full XML example including other SOAP operations with requests and responses, please go to Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```

<v2:printDocumentRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T09:14:32</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>667767512</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:shipmentNumber>RQ221150275GB</v2:shipmentNumber>
  <v2:documentName>CN22</v2:documentName>
</v2:printDocumentRequest>
  
```

8.18 printDocumentResponse

The response for a requested printDocumentResponse operation is constructed as a SOAP message (see section 7) with the printDocumentResponse element contained within the

SOAP Body. The printDocumentResponse element is described in the figure and table below:

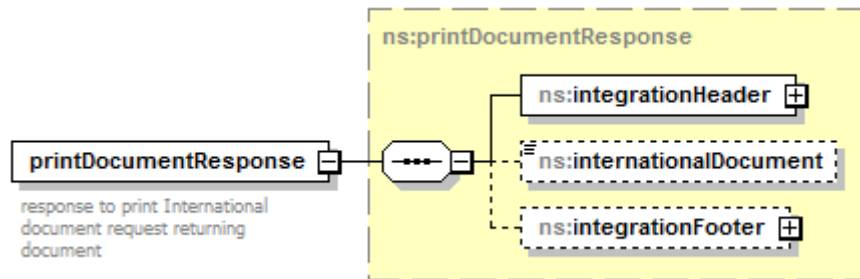


Figure 34 – printDocumentResponse element Structure

All elements in the table below are relative to the `printDocumentResponse` element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.6.1
internationalDocument		0-1	document	International Document in PDF format -Base64 encoded for transfer
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the printDocument operation. Please see Section 8.6.3 for more information.

Table 39 – printDocumentResponse Element

8.18.1 Example Data

This section provides a simplified example to illustrate the `printDocumentResponse` without the SOAP wrappers. For a full XML example including other SOAP operations with requests and responses, please go to the [Shipping API V2](#) page on the [Royal Mail API \(Developer\) Portal](#) and refer to the [Shipping API V2 \(SOAP\) Sample Data Set](#).

```
<printDocumentResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T09:14:32</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>11111113</applicationId>
      <transactionId>667767512</transactionId>
    </identification>
  </integrationHeader>
  <internationalDocument>JVBERi0xLjMKJeLjz9MKMSAwIj9Gaiw8L1Byb2RlY2VyKGh0bWxkb2MgMS44LjI3IEh0bWwHyaAwa
dCAxOTk3LTlwMDYgRWFWZeSBtb2Z0d2FyZSBQcm9kdWN0cywgQWxsIFJpZ2h0cyBSZXNlcnciZlZC4p
L0NyZWFOaW9uRGF0ZShEoJlWMTUwMjA5MTAxNDMyLTAtXmDApPj5lbmRvVmoKMiAwIj9Gaiw8L1R1Z
cGUvRW5jb2RpbmcvRGlmc2MvY2ZS9jaW5jZXNlIDMyL3NuYWwNIL2V4Y2xhbS9xdW90ZWRibC9udW1iZXJz
aWdul2RvbGxhcic9wZXRjZW50L2FtcGVyc2FuZC9xdW90ZXNpbmdsZS9wYXJlbnmxLnQvcGFyZW5y
aWdodC9hc3Rlcmlyay9wbHVzL2NvbW1hL2h5cGhlbi9wZXJpb2Qvc2xhc2gvemVyby9vbmUvdHdv
L3RocmVLl2ZvdXlvZml2ZS9zaXgvc2V2ZW4vZWlnaHQvbmlluzS9jb2xvbi9zZW1pY29sb24vbGVz
CnRyYWsIZXIKPDwwU2l6ZSAyNi9Sb290IDI1IDAgUi9JbmZvlDEgMCBLS0IEWzw1Nzc0ODI2ZGMx
MDI3ZmlzODgwZGYmMDE3YjQ5MG12OT48NTc3NDgyNmRjMTAyN2ZiMzg4MGRjMjA5N2I0OTBiNjk+
XT4tCnNOYXJ0eHJlZGoxNzEzMjUKJSVFOT0YK</internationalDocument>
  <integrationFooter>
    <warnings xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <warning>
        <warningCode>W0045</warningCode>
        <warningDescription>CN23 is normally used for shipments with given value</warningDescription>
      </warning>
    </warnings>
  </integrationFooter>
</printDocumentResponse>
```

8.19 request1DRangesRequest

The request1DRanges operation allows you to request a range of 1D Linear Barcode numbers for use in processing offline shipments.

It should be noted that before calling this API you should call the Customer Solutions team to get your API login/account enabled for this feature.

To invoke the request1DRanges operation, the customer shipping system constructs a SOAP message (see section 7) with the Request1DRanges element contained within the SOAP Body. The Request1DRanges element is described in the figure and table below:

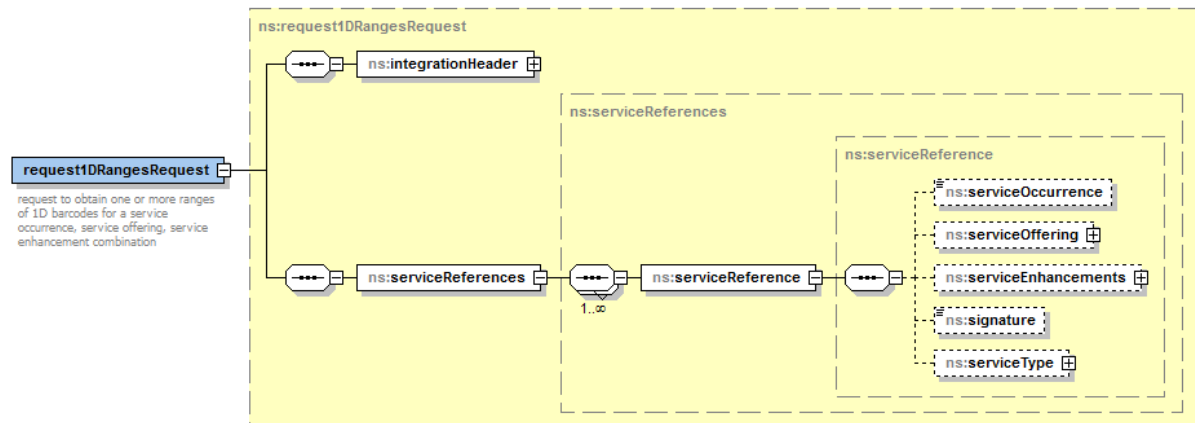


Figure 35 – request1DRangesRequest Structure

All elements in the table below are relative to the request1DRangesRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.5.1
serviceReferences	N/A	1-1	element	Mandatory. Container to hold the references for the services under request
serviceReferences/serviceReference	N/A	1- ∞	element	Mandatory. Container to hold the service definition for the 1D Linear Barcode range under request
serviceReferences/serviceReference/*	N/A		element	See create shipment for values for: <ul style="list-style-type: none">• serviceOccurrence• serviceOffering• serviceEnhancements• signature• serviceType

Table 40 – request1DRangesRequest Element

8.19.1 Example Data

This section provides a simplified example to illustrate the request1DRangesRequest without the SOAP wrappers and WS-Security elements. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```

<v2:request1DRangesRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T09:24:49</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>916422861</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:serviceReferences>
    <v2:serviceReference>
      <v2:serviceOccurrence>1</v2:serviceOccurrence>
      <v2:serviceOffering>
        <serviceOfferingCode>
          <code>MP5</code>
        </serviceOfferingCode>
      </v2:serviceOffering>
      <v2:serviceType>
        <code>l</code>
      </v2:serviceType>
    </v2:serviceReference>
  </v2:serviceReferences>
</v2:request1DRangesRequest>

```

8.20 request1DRangesResponse

The response for a requested request1DRangesResponse operation is constructed as a SOAP message (see section 7) with the request1DRangesResponse element contained within the SOAP Body. The request1DRangesResponse element is described in the figure and table below:

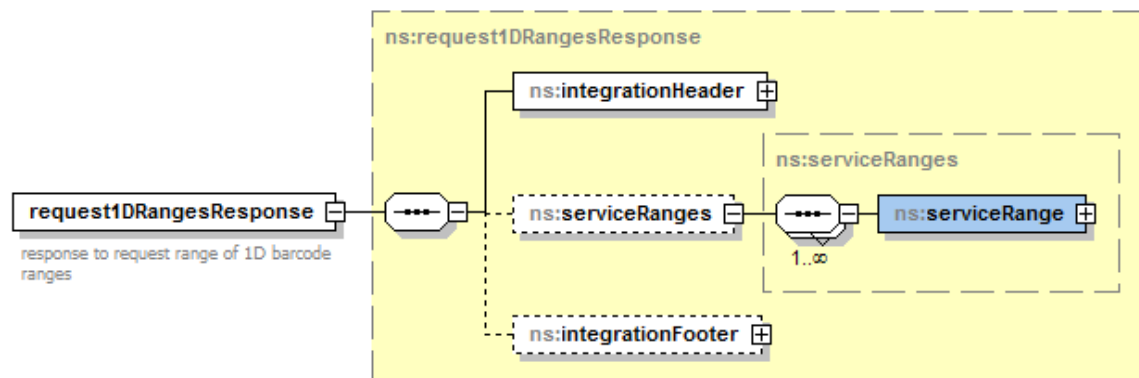


Figure 36 – request1DRangesResponse Structure

All elements in the table below are relative to the request1DRangesResponse element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.6.1
serviceRanges		0-1	element	Container to hold array of ranges per service requested
serviceRanges/service Range		1- ∞	element	This is a complex type and is detailed below
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the Request1DRanges operation. Please see Section 8.6.3 for more information.

Table 41 – request1DRangesResponse Element

serviceRange

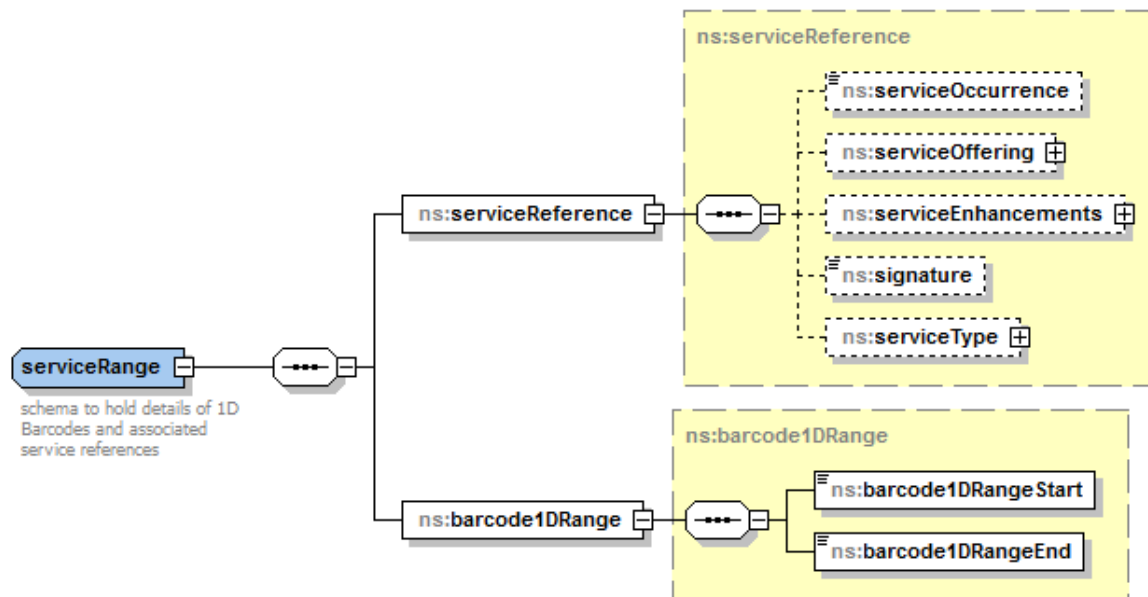


Figure 37 – serviceRange Structure

All elements in the table below are relative to serviceRange.

Element	Max Length	Occurs	Data Type	Description
serviceReference	N/A	1-1	element	See create shipment for values for: <ul style="list-style-type: none"> • serviceOccurrence • serviceOffering • serviceEnhancements • signature • serviceType
barcode1DRange		1-1	element	Container to hold array of ranges per service requested
barcode1DRange/barcode1DRangeStart		1-1	identifier	1D Linear Barcode range start value
barcode1DRange/barcode1DRangeEnd		1-1	identifier	1D Linear Barcode range end value

Table 42 – serviceRange Element

8.20.1 Example Data

This section provides a simplified example to illustrate the request1DRangesResponse without the SOAP wrappers. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```
<request1DRangesResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T09:24:49</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>916422861</transactionId>
    </identification>
  </integrationHeader>
  <serviceRanges>
    <serviceRange>
      <serviceReference>
```



```

<serviceOccurrence>1</serviceOccurrence>
<serviceOffering>
  <serviceOfferingCode xmlns="">
    <code>MP5</code>
  </serviceOfferingCode>
</serviceOffering>
<serviceEnhancements>
  <enhancementType>
    <serviceEnhancementCode xmlns=""/>
  </enhancementType>
</serviceEnhancements>
<signature>>false</signature>
<serviceType>
  <code xmlns="">I</code>
</serviceType>
</serviceReference>
<barcode1DRange>
  <barcode1DRangeStart>RQ285500433GB</barcode1DRangeStart>
  <barcode1DRangeEnd>RQ285510427GB</barcode1DRangeEnd>
</barcode1DRange>
</serviceRange>
</serviceRanges>
</request1DRangeResponse>

```

8.21 request2DItemIDRange

The request2DItemIDRange operation allows you to request a range of 2D Item ID numbers for use in processing offline shipments.

It should be noted that before calling this API you should call the Customer Solutions team to get your API login/account enabled for this feature.

To invoke the request2DItemIDRange operation, the customer shipping system constructs a SOAP message (see section 7) with the request2DItemIDRangeRequest element contained within the SOAP Body. The request2DItemIDRangeRequest element is described in the figure and table below:

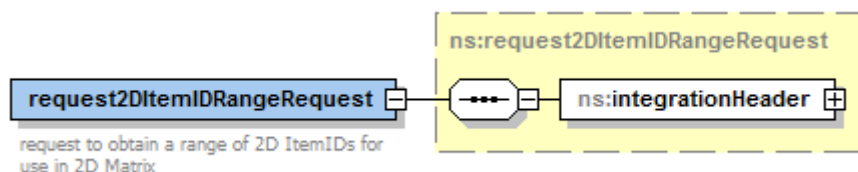


Figure 38 – request2DItemIDRangeRequest Structure

All elements in the table below are relative to the request2DItemIDRangeRequest element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.5.1

Table 43 – request2DItemIDRangeRequest Element

8.21.1 Example Data

This section provides a simplified example to illustrate the request2DItemIDRangeRequest without the SOAP wrappers and WS-Security elements. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```

<v2:request2DItemIDRangeRequest>
  <v2:integrationHeader>
    <v1:dateTime>2015-02-09T10:45:33</v1:dateTime>
    <v1:version>2</v1:version>
    <v1:identification>
      <v1:applicationId>111111113</v1:applicationId>
      <v1:transactionId>350301134</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
</v2:request2DItemIDRangeRequest>

```

8.22 request2DItemIDRangeResponse

The response for a requested request2DItemIDRangeResponse operation is constructed as a SOAP message (see section 7) with the request2DItemIDRangeResponse element contained within the SOAP Body. The request2DItemIDRangeResponse element is described in the figure and table below:

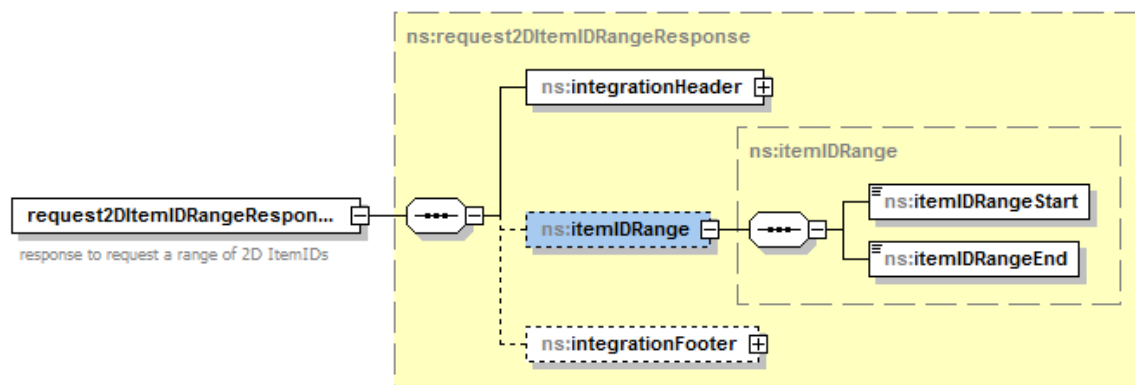


Figure 39 – request2DItemIDRangeResponse Structure

All elements in the table below are relative to the request2DItemIDRangeResponse element.

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.6.1
itemIDRange	N/A	0-1	element	Container for the 2D Item ID Range start and end field
itemIDRange/itemIDRangeStart		1-1	identifier	2D Item ID range start value
itemIDRange/itemIDRangeEnd		1-1	identifier	2D Item ID range end value
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the Request2DItemIDRange operation. Please see Section 8.6.3 for more information.

Table 44 – request2DItemIDRangeResponse Element

8.22.1 Example Data

This section provides a simplified example to illustrate the request2DItemIDRangeResponse without the SOAP wrappers. For a full XML example including other SOAP operations with requests and responses, please go to the Shipping API V2 page on the Royal Mail API (Developer) Portal and refer to the Shipping API V2 (SOAP) Sample Data Set.

```
<request2DItemIDRangeResponse xmlns="http://www.royalmailgroup.com/api/ship/V2">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2015-02-09T10:45:33</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">2</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>350301134</transactionId>
    </identification>
  </integrationHeader>
  <itemIDRange>
    <itemIDRangeStart>0002250001</itemIDRangeStart>
    <itemIDRangeEnd>0002500000</itemIDRangeEnd>
  </itemIDRange>
</request2DItemIDRangeResponse>
```

9 Error Handling

9.1 Overview

The Shipping API V2 service can communicate issues in three ways:

1. **Technical Errors**

Highlights fundamental problems with either the system or the request. All technical errors (e.g. schema validation failure, service unavailable etc.) are returned as SOAP Faults to the service requester.

2. **Business Errors**

While the request was correctly formatted; it contained invalid data that cannot be automatically corrected. All business errors are returned in the integrationFooter of the SOAP response message.

3. **Warnings**

While the request was correctly formatted and the data was mostly accurate, there were minor corrections or assumptions that the system had to make to allow the request to be processed. All warnings are returned in the integrationFooter of the SOAP response message.

Errors and warnings are the same in both the onboarding and live environments. Please refer to the Shipping API V2 page on the [Royal Mail API \(Developer\) Portal](#) for the full list of all error and warning codes which can be returned.

9.2 Technical Errors

Technical Errors highlight that there is either a problem with the Royal Mail system or there is a fundamental problem with the messages being sent to Royal Mail by the customer. Examples of technical errors caused by the customer would include the submission of a request message that failed schema validation.

Technical errors indicate that the request was not successful, and has not been processed by Royal Mail (e.g. createShipment requests that receive a Technical Error will not result in a shipment being created on the system).

Technical errors may also be an indication of serious problems with the interaction between the customer and Royal Mail, and customer systems should be built in such a way that these errors are gracefully handled, captured and reported to the relevant technical resources. Failure to do so may result in a disruption to service.

All technical exceptions are returned to the customer using the SOAP Fault message construct. Please see figure below for a graphical representation of the SOAP Fault construct which has been extended by the Royal Mail Group to specify a number of sub-elements under the <detail> element.

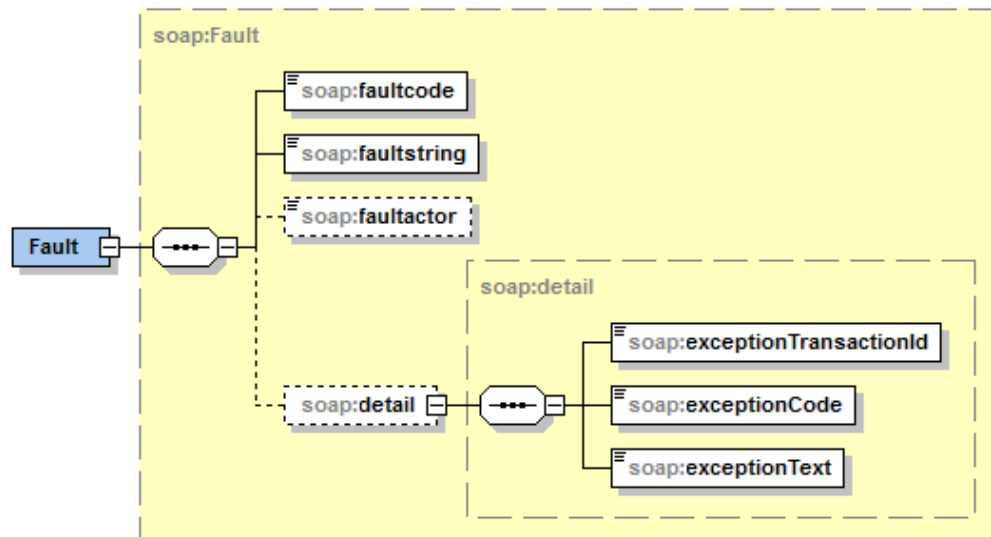


Figure 40 – SOAP Fault Structure

The SOAP Fault element has the following sub-elements:

Element	Max Length	Occurs	Data Type	Description
faultcode	N/A	1-1	QName	A code for identifying the fault
faultstring	N/A	1-1	string	A human readable explanation of the fault
faultactor	N/A	0-1	anyURI	Information about who caused the fault to happen.
detail	N/A	0-1	element	Container for application specific error information
detail/exceptionTransactionId	50	1-1	identifier	Used to identify the transaction Id associated with the request message which generated this technical exception.
detail/exceptionCode	5	1-1	identifier	Error code associated with the technical exception.
detail/exceptionText	256	1-1	description	A meaningful human-readable error description of the error condition.

Table 45 – SOAP Fault Elements

Please see the Shipping API V2 page on the [Royal Mail API \(Developer\) Portal](#) for the full set of technical errors that may be returned by the API and the values which will be populated in the above fields for each error condition.

9.3 Business Errors

Business errors indicate that the data provided in the request, while correctly formatted, is not valid. Examples of issues that would lead to a business error being received include attempting to use a service that the customer does not have an agreement line for, specifying a shipment date too far in the future, or attempting to update a shipment that has already been manifested.

Business errors indicate that the request was not successful, and has not been processed by Royal Mail (e.g. updateShipment requests that receive a business error will not result in any changes to the specified shipment).

All business errors are returned in the integrationFooter of the SOAP response message. Please see the Shipping API V2 page on the [Royal Mail API \(Developer\) Portal](#) for the set of business errors that may be returned by the API.

9.4 Warnings

Warning messages are received when the data supplied to Royal Mail in the request is not completely accurate, but the system is able to make assumptions or corrections automatically. Examples of problems that would result in a warning message include: address lines with too many characters (that are then truncated), a telephone number being specified without an SMS option being selected (and hence being ignored), and a shipment date being specified in the past (and being automatically updated to today's date).

Requests that result in a warning message are processed by the system (e.g. a createShipment request where the customer reference is longer than 12 characters will be created on the system, but with the customer reference truncated to 12 characters).

All warnings are returned in the integrationFooter of the SOAP response message. Please see the Shipping API V2 page on the [Royal Mail API \(Developer\) Portal](#) for the set of warnings that may be returned by the API.

10 Non-Functional Characteristics

10.1 Availability

10.1.1 Service Hours

The Shipping API is available 24 hours per day x 365 days per year.

10.1.2 Maintenance Windows

Royal Mail Online Services Terms and Conditions define the maintenance for this service.

10.1.3 Unavailability

In the unlikely event of the Shipping API being unavailable, customer systems must be able to handle this appropriately. Royal Mail will endeavour to proactively contact customers in the event of an outage to this API, using the email address provided to us during onboarding.

10.2 Performance

Performance will be slower during peak periods of activity (between 3pm and 6pm Monday to Friday). To avoid issues associated with slower response times during periods of heavy traffic, customers are strongly advised to spread their traffic out throughout the day (wherever possible).

10.3 Security

All API service calls will be made using mutually authenticated HTTPS bound SOAP web services. The Shipping API exposes the services using SOAP version 1.1 with a document / literal document-style encoding.

All service requests via the API Management solution will be authorised in accordance with the Client ID and Secret passed in the HTTP headers. This will ensure that any external service requests are authorised and authenticated in line with RMG Security Policies and Standards.

11 Frequently Asked Questions

Please see the [FAQ page](#) on the [Royal Mail API \(Developer\) Portal](#) for a general list of frequently asked questions with responses.

Please see the Shipping API V2 API page on the [Royal Mail API \(Developer\) Portal](#) for a list of frequently asked questions with responses.

12 Appendix A – Reference Data

12.1 Allowable Character Set

Please see the diagram below which defines the supported character set for Shipping API V2.

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	‘
010 0011	043	35	23	#
010 0110	046	38	26	&
010 0111	047	39	27	’
010 1000	050	40	28	(
010 1001	051	41	29)
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1101	135	93	5D]
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	
111 1101	175	125	7D	}
111 1110	176	126	7E	~

Figure 41 – Allowable Character Set for Shipping API V2

12.2 Shipment Status Codes

The values for a shipment status are defined in the table below:

<status>	Description
Allocated	Shipment with a Service Type / Service / Service Format and shipment number but not printed
AllocatedOffline	
Printed	Shipment with Service Type / Service / Service Format and shipment number and the label(s) printed
PrintedOffline	
Manifested	Customer Collection Receipt has been created Customer Collection Receipt has been printed
Cancelled	Shipment has been cancelled

Table 46 – Shipment Status Codes

12.3 XML Examples

Please see the Shipping API V2 API page on the [Royal Mail API \(Developer\) Portal](#) to download the following XML examples for each web service operation.

createShipment

- Request (createShipmentRequest.xml)
- Response (createShipmentResponse.xml)

updateShipment

- Request (updateShipmentRequest.xml)
- Response (updateShipmentResponse.xml)

cancelShipment

- Request (cancelShipmentRequest.xml)
- Response (cancelShipmentResponse.xml)

printLabel

- Request (printLabelRequest.xml)
- Response (printLabelResponse.xml)

createManifest

- Request (createManifestRequest.xml)
- Response (createManifestResponse.xml)

printManifest

- Request (printManifestRequest.xml)
- Response (printManifestResponse.xml)

request1DRangesRequest

- Request (request1DRangesRequest.xml)
- Response (request1DRangesResponse.xml)

request2DItemIDRangeRequest

- Request (request2DItemIDRangeRequest.xml)
- Response (request2DItemIDRangeResponse.xml)

printDocument

- Request (printDocumentRequest.xml)
- Response (printDocumentResponse.xml)



Royal Mail, the cruciform, the colour red and all ® are registered trademarks and all ™ are trademarks of Royal Mail Group Ltd. Shipping API V2 Technical User Guide October 2016 © Royal Mail Group Ltd 2016. All rights reserved.