API Specification

Royal Mail Group

Local Collect (SOAP) API

Technical User Guide

This API specification details the requirements for integrating with the Local Collect (SOAP) API. It specifically covers how the Local Collect API can be used by business customers looking to book capacity slots at a Post Office outlet convenient to the recipient, and provides the technical information to build this integration.

4th December 2015

Version 1.1

Contents

1 Document Control

1.1 Terms and Abbreviations

Term	Meaning
BIG	Business Integration Gateway
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over SSL
IP	Internet Protocol
LC	Local Collect
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 Services Description Language
XML	Extensible Markup Language

Table 1 – Terms and Abbreviations

1.2 Version History

Version	Date	Author	Notes
1.0	01/11/2015	Mark Cornforth	Document created and baselined. Document now replaces "COSS spec 08 – Local Collect API v1.9'.
1.1	04/12/2015	Mark Cornforth	'Accept' parameter added to HTTP Header in section 8.3. Minor updates made to assist with readability.

Table 2 – Document Version History

2 Overview

The Royal Mail Local Collect API exposes a web service that allows customers to obtain Post Office location data and subsequently confirm selection of a particular Post Office for parcel delivery.

There are no costs to customers for using the Local Collect API, 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.

Local Collect is available as an option with Royal Mail shipping solutions Despatch Manager Online (DMO), Shipping API (SAPI) and a Customers own system solution (COSS). Specifications for these shipping solutions are available from the Royal Mail Customer Solutions team.

Customers wishing to develop or implement a system to produce barcode labels on their own system (COSS) should gain agreement from Royal Mail before commencing work. Labels produced are subject to approval by Royal Mail before being used, and live barcode number ranges will not be issued before this approval is given. Further sample labels will also need to be submitted to Royal Mail periodically for quality checks.

Note that following approval, any changes or additions to the outputs relating to Royal Mail despatches will also need to be discussed and validated before implementation. This includes the addition of new services.

3 Purpose

This document is to provide Royal Mail customers with guidelines and detailed specifications for integrating with the Local Collect SOAP web service.

The document details:

- The specification for the web service interface for customers who want to query Post Office location data and subsequently confirm selection of a Post Office for parcel delivery
- 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 integrated customer systems' with the Local Collect API. This document should be read in conjunction with the following artefacts which are available from the Local Collect API page on the Royal Mail API (Developer) Portal:

- Local Collect WSDL
- Local Collect XSDs

4 Introduction to Local Collect API

4.1 Overview

The Local Collect Service is a delivery enhancement to Royal Mail Tracked and Special Delivery Guaranteed products that enables customers to benefit from 'Click and Collect' options by having their parcels delivered to participating Post Offices. With over 10,000 Post Offices providing collection points, consumers have a wide choice of where their parcel is delivered to, and when they collect it.

To support this delivery enhancement, and to ensure Post Offices have space to accept parcels, a capacity management system has been developed and presented through a set of web services. These web services, using industry standard SOAP over HTTPS, have been designed to be simple and easy to integrate while minimising the time it takes to modify e-commerce platforms.

The operations exposed by the Local Collect API enable customers to:

- Retrieve an up-to-date list of Post Offices that offer Local Collect and have sufficient capacity to receive parcels (for a given location and delivery date)
- Confirm selection in a given Post Office for a delivery to ensure there is sufficient physical space for parcels to be held awaiting collection

4.2 Interface Components

Please see Figure 1 below for a graphical representation of the interface between Royal Mail and customers for the Local Collect API. This document covers what information is to be exchanged, how this information is structured and the means by which it is transferred.

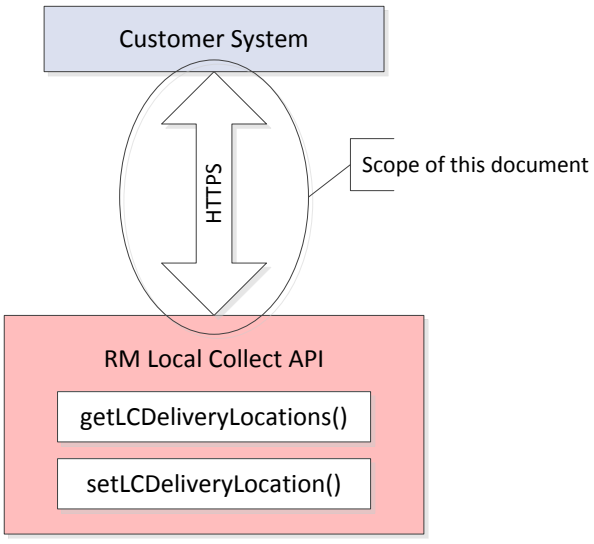


Figure 1 – Local Collect API

5 Integrating with the Local Collect API

The high-level process associated with integrating with the Local Collect API is represented and described in the diagram below.

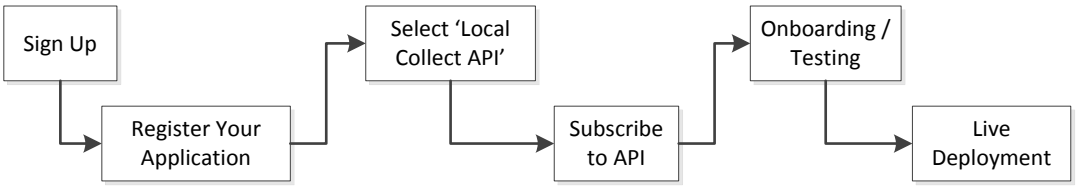


Figure 2 – Process for Integrating with the API

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.
2. Register the '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 is available 24x7 and has the same functionality as live with the exception that all Post Office locations will always have space available and will never be shown as full.
- The onboarding environment may not be used for performance testing. This is a small scale system for functional testing only.

- The API imposes a cap on the number of transactions per second for each customer. Excessive volumes of traffic over the configured throttling limit will result in transactions being rejected.

5.2 API Access

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

https://api.royalmail.net/LocalCollect

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.

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

6 Local Collect Services

6.1 Business Services

The Local Collect service is a delivery enhancement to Royal Mail products that enables customers to benefit from 'Click and Collect' options by having their parcels delivered to participating Post Offices.

The table below provides an overview of the business services that are supported by this interface.

Business Service	Web Service Operation	Description	Technology	Conversation Style
Get Delivery Locations	GetLCDeliveryLocations	Returns a list of Post Offices including (for each location) a flag to indicate if the Post Office has capacity available, and a unique reservation code.	SOAP over HTTPS	Synchronous Request / Response
Set Delivery Location	SetLCDeliveryLocation	Confirm selection in a given Post Office for a delivery to ensure there is sufficient physical space for parcels to be held awaiting collection.	SOAP over HTTPS	Synchronous Request / Response

Table 3 – Business Services

Royal Mail anticipates that customers will present delivery options and available Post Offices to the consumer as part of the order process within their e-commerce systems. The **GetLCDeliveryLocations** operation returns a list of Post Offices including (for each location) a flag to indicate if the Post Office has capacity available, and a unique reservation code.

Once a consumer has placed the order (which will result in one or more parcels being despatched), the customer's system must make a subsequent call to the **SetLCDeliveryLocation** operation (using the correct reservation code returned as part of the GetLCDeliveryLocations call). The response to the SetLCDeliveryLocation includes a confirmation number (which is not required to collect the parcel) and a repeat of the location details (name, address etc) of the Post Office selected.

It is not permitted for customers to cache results from the GetLCDeliveryLocations and it should be noted that reservation codes are only valid for 10 minutes. As such, if a consumer takes longer than 10 minutes from selecting an available Post Office to confirming the order, a further call to GetLCDeliveryLocations must be made to retrieve a new reservation code. If a call is made to the SetLCDeliveryLocation operation with an invalid reservation code the API will return an error (see section 9.2).

7 Message Structure

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

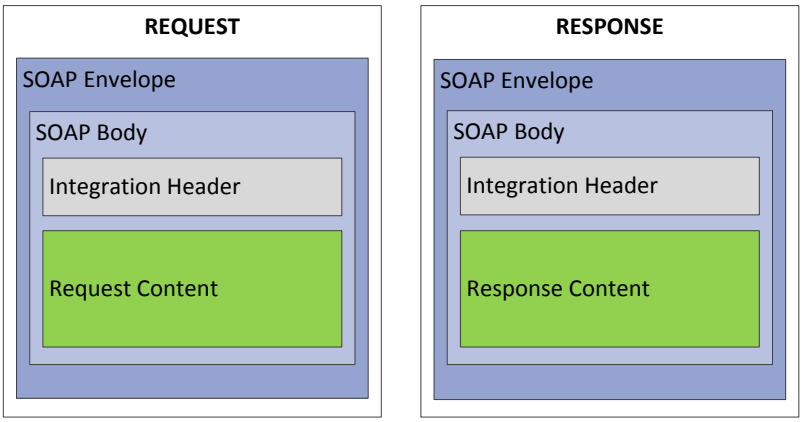


Figure 3 – API Structure

The GetLCDeliveryLocations and SetLCDeliveryLocation operations are defined as SOAP operations, with separate request / response message parts. Technical details 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 which is described in Section 8.4.1.

The definition of the data types used within the elements section of each service call is defined within the schemas. The schemas contain a number of optional elements (e.g. altitude within the searchPosition, geoSpatialPositionType within searchPosition etc) that are not used as part of the Local Collect API and as such should be ignored (and are not included in the tables in the section 8.4).

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.

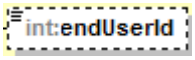
Graphical Representation	Meaning
	A solid line around an element indicates it is a mandatory field that will/must always be present.
	A dashed line around an element indicates the field is optional and may or may not be present. An optional element is one that indicates the data and enclosing XML tags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

Figure 4 – Notation used to describe this API

8.2 WSDL and Schemas

The WSDL and XSDs for the Local Collect API can be found on 'Royal Mail APIs' section of the Royal Mail API (Developer) Portal.

The following table lists the documents that should be referenced for the RMG-defined canonical data types used by Local Collect API. The XML schema documents themselves are provided in supporting files. Each schema file contains descriptions of every type using the "xs:documentation" element.

XML Schema File	Description
CommonClassesV1_2.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.
DatatypesV1_2.xsd	RMG defined simple types that are commonly required such as "date" and "name". This schema is the foundation for the other schema files.
IntegrationHeaderV1_1.xsd	RMG schema which defines a common header definition to be used for request and response messages.
ReferenceDataV1_2.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.
RMLCSchemaV2_4.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.

Table 4 – 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.

Parameter	Optional	Description
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.

Table 5 – HTTP Header Information in the API Request

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

Table 6 – Example HTTP Header Information for API Request

8.4 GetLCDeliveryLocationsRequest

The behaviour of the GetLCDeliveryLocations operation is to return a list of Post Offices that are available and in proximity to the postcode or position provided as part of the request parameters. The number of results returned is determined by the use of the radius parameter (although if this is omitted a pre-determined list of 20 locations will be returned).

Please see diagram below for a representation of the GetLCDeliveryLocations request message:

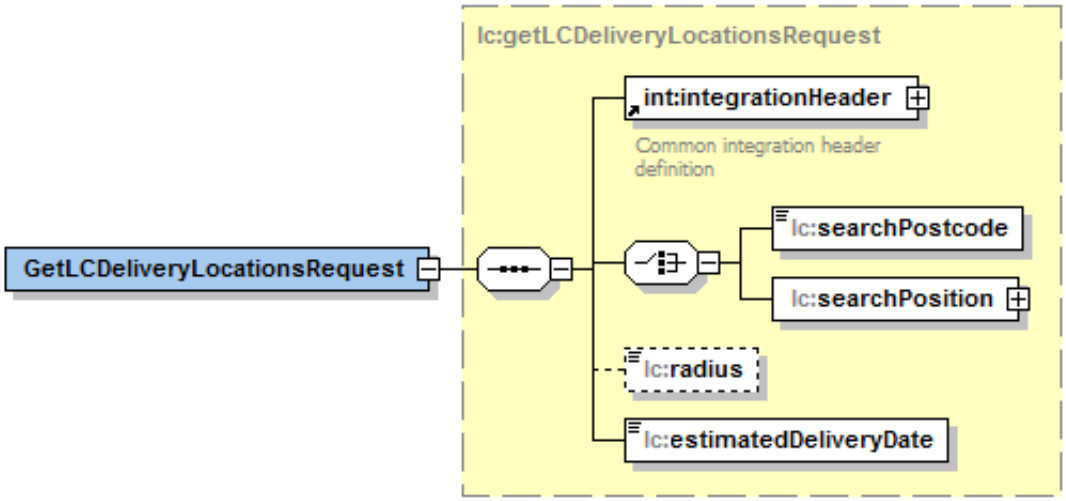


Figure 5 – GetLCDeliveryLocationsRequest Message Structure

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

Element	Optional	Description
integrationHeader	No	Please see section 8.4.1.
searchPostcode	Either a searchPostcode or searchPosition (and all child elements) element must be provided.	This is the Postcode of the location to search for Post Offices near. Only UK postcodes are supported
searchPosition		This is the container element for geospatial information (i.e. the geoDeticSystem, longitude & latitude) to be used for geospatial Post Office searches.
searchPosition.geoDeticSystem.systemNameCode.code	No (if searchPosition element is used)	This is the geodetic system of the position being specified. Only WGS84 is supported and this value must be specified. If other values are specified (e.g. OSGB36) these will be ignored and the longitude and latitude values will be assumed to be in the WGS84 datum
searchPosition.longitude	No (if searchPosition element is used)	This is the longitude of the position to search for Post Offices near
searchPosition.latitude	No (if searchPosition element is used)	This is the latitude of the Position to search for Post Offices near
radius	Yes	By setting this value, only Post Offices that are within the specified radius (in miles up to a maximum of 99.0) are returned. Note: there is a system imposed restriction which will limit the number of results returned (to the 20 nearest Post Offices) regardless of how large the radius is set to

estimatedDeliveryDate	No	This is the date when the parcel is anticipated to arrive at the Post Office (taking into account Royal Mail product used to deliver the item e.g. Tracked 24). If customer's do not know when the parcel will be despatched (e.g. due to consumers ordering out of stock items) customers should specify a delivery date that assumes the order will be despatched on the same day (as the order). Notes: 1. estimatedDeliveryDate must be at least one day in the future 2. estimatedDeliveryDate cannot be more than 30 days in the future

Table 7 – GetLCDeliveryLocationsRequest Structure

8.4.1 integrationHeader

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:

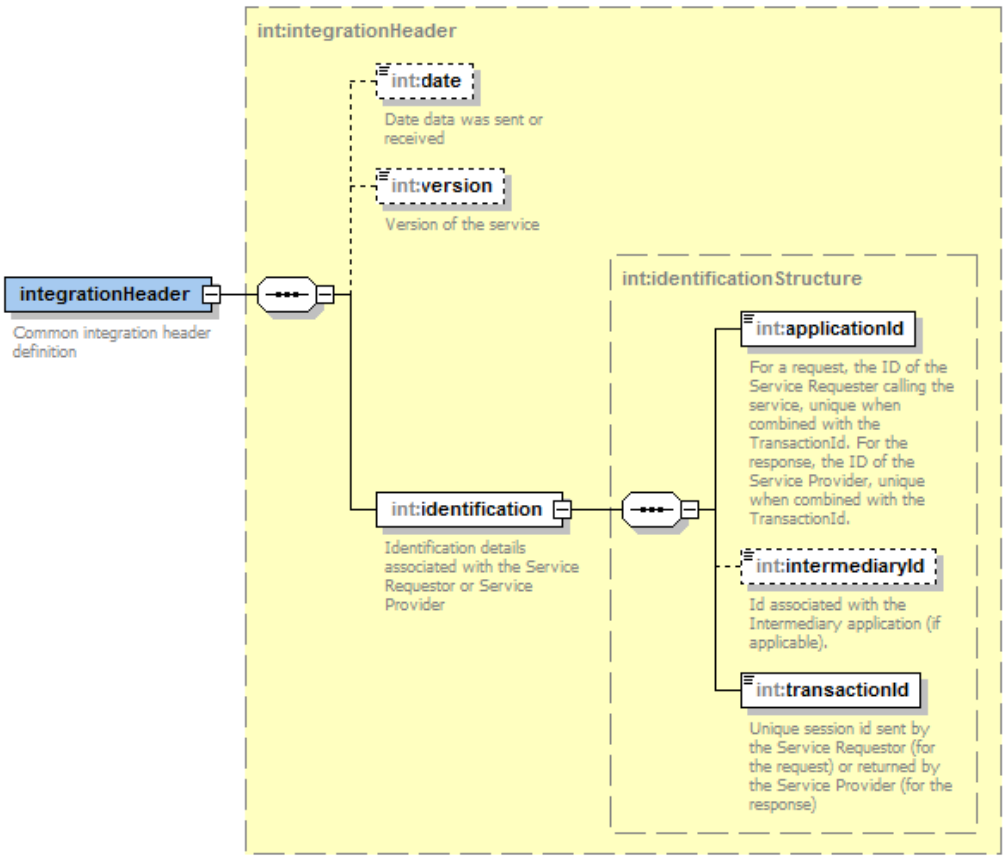


Figure 6 – integrationHeader Structure

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

Element	Optional	Description
Date	Yes	This should be populated with the date the message was generated
Version	Yes	The version of the API currently being used (initially 1.0)
identification.applicationId	No	This is the ten digit Customer Account Number allocated by Royal Mail
identification.intermediaryId	Yes	This is the ID of an intermediary organisation involved in processing this web service
identification.transactionId	No	This is a unique number to identify the transaction number as provided by customer systems

Table 8 – integrationHeader Element (Request)

8.4.2 Example Data

Full XML examples of SOAP requests and responses are provided on the Royal Mail API (Developer) Portal. This section provides a simplified extract to illustrate the GetLCDeliveryLocationsRequest only.

```
<v2:GetLCDeliveryLocationsRequest>
  <v2:integrationHeader>
    <v1:date>2015-08-02</v1:date>
    <v1:version>1</v1:version>
    <v1:identification>
      <v1:applicationId>1234567890</v1:applicationId>
      <v1:transactionId>1</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>
  <v2:searchPostcode>SW3 4SR</v2:searchPostcode>
  <v2:estimatedDeliveryDate>2013-08-31</v2:estimatedDeliveryDate>
</v2:GetLCDeliveryLocationsRequest>
```

8.5 GetLCDeliveryLocationsResponse

The body of the response message contains all the Post Office locations as well as any errors that may have occurred. Please see diagram below for a representation of the GetLCDeliveryLocations response message:

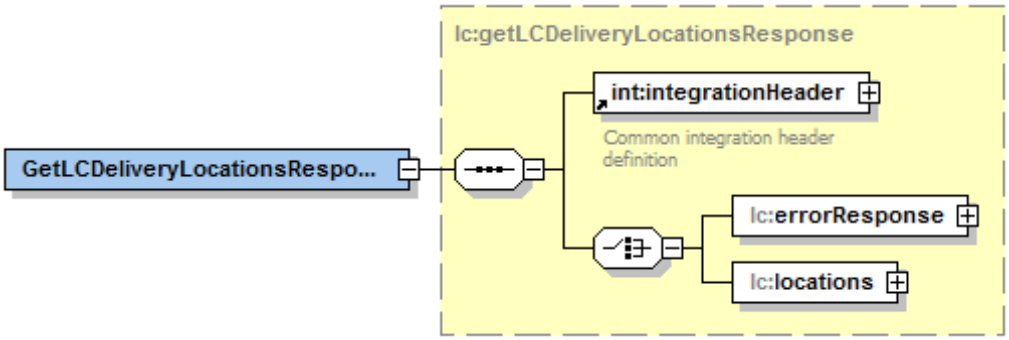


Figure 7 – GetLCDeliveryLocationsResponse Message Structure

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

Element	Optional	Description
integrationHeader	No	Please see section 8.5.1.
locations	No	This element will only be returned in the event that no error condition has occurred.
locations.location.organisationName	No	This is the organisation name for the location (populated with "Post Office Limited")
locations.location.locationPosition.geoDeticSystem.systemNameCode.code	No (if locationPosition element is provided)	The geodetic system of the position for the location (this always will be WGS84)
locations.location.locationPosition.longitude	Yes	Longitude of the location
locations.location.locationPosition.latitude	Yes	Latitude of the location
locations.location.locationName	No	Name of the location
locations.location.address.addressLine1	No	Address line 1 of the location
locations.location.address.addressLine2	Yes	Address line 2 of the location
locations.location.address.addressLine3	Yes	Address line 3 of the location
locations.location.address.addressLine4	Yes	Address line 4 of the location. This will always be the name of the town the location is in
locations.location.address.county.countyCode.name	Yes	Name of county the location is in
locations.location.address.postcode	Yes	Postcode of the location
locations.location.lCAvailability	Yes	Indicates if the Local Collect service is available at this location. Boolean value
locations.location.lCBookingReference	Yes	Unique reference that can be subsequently used to confirm delivery for a location. If lcAvailability is false, the value in this field will be 99999999999999999999999XX99XXXXXX
locations.location.searchDistance	Yes	Distance the location is from postcode or position as provided in the request message (in miles as the crow flies)
For each day the location is open*		
locations.location.openingDay.dayOfWeekType.dayOfWeekCode.name	Yes	Day name
locations.location.openingDay.openingTime	No	Opening time of the location
locations.location.openingDay.closingTime	No	Closing time of the location. Note: It's not possible to collect items when location is closed

		(i.e. outside of opening hours)
locations.location.openingDay.lunchTime.lunchClosingTime	Yes	Time the location closes for lunch. Note: It is not possible to collect items when the location is closed (i.e. during lunch hours)
locations.location.openingDay.lunchTime.lunchOpeningTime	Yes	Time the location opens after lunch
errorResponse	No	This element will only be returned in the event that an error condition has occurred. See section 8.5.2 for details.

Table 9 – GetLCDeliveryLocationsResponse Structure

*Note: On Local or Public Holidays, individual opening times may vary.

8.5.1 integratioHeader

All elements are relative to the integrationHeader element in the response message.

Element	Optional	Description
date	Yes	This is always returned and contains the date when the Royal Mail system processed the transaction.
version	Yes	This is always returned and contains the current version of the Local Collect API (initially 1.0)
identification.applicationId	No	This is always populated with 'RoyalMailGroup'
identification.intermediaryId	Yes	Not currently used.
identification.transactionId	No	This is a unique string provided by Royal Mail to identify the transaction (and does not correlate to the transactionId in the request message).

Table 10 – integrationHeader Element (Response)

8.5.2 errorResponse

The errorResponse contains all business errors which can be returned from calling this API. Please see diagram below for a representation of the errorResponse response element:

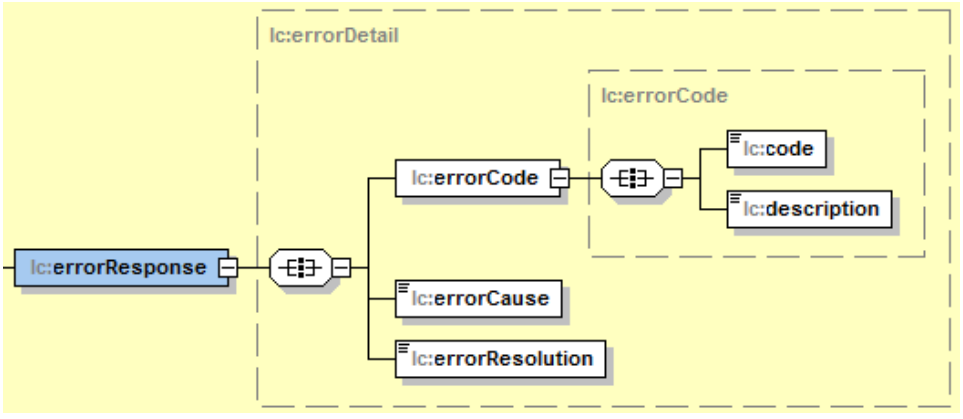


Figure 8 – errorResponse Structure

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

Element	Optional	Description
errorCode.code	No	Code associated with the error condition
errorCode.description	No	Description of the error condition
errorCause	No	Cause of the business error
errorResolution	No	Description of the resolution and action required to correct the error

Table 11 – errorResponse Structure

Please see section 9 for details of the business errors which can be returned from calling this API operation.

8.5.3 Example Data

Full XML examples of SOAP requests and responses are provided on the Royal Mail API (Developer) Portal. This section provides a simplified extract to illustrate the GetLCDeliveryLocationsResponse only.

```
<NS1:GetLCDeliveryLocationsResponse
xmlns:NS1="http://www.royalmailgroup.com/API/LocalCollect/V2.0">
  <NS1:integrationHeader>
    <NS2:date xmlns:NS2="http://www.royalmailgroup.com/integration/core/V1">2013-08-02</NS2:date>
    <NS3:version xmlns:NS3="http://www.royalmailgroup.com/integration/core/V1">1.0</NS3:version>
    <NS4:identification xmlns:NS4="http://www.royalmailgroup.com/integration/core/V1">
      <NS4:applicationId>RoyalMailGroup</NS4:applicationId>
      <NS4:transactionId>78d106a0-fb84-11e2-97e9-ac14a0b80000</NS4:transactionId>
    </NS4:identification>
  </NS1:integrationHeader>
  <NS1:locations>
    <NS1:location>
      <NS1:organisationName>Post Office Limited</NS1:organisationName>
      <NS1:locationPosition>
        <geoDeticSystem>
          <systemNameCode>
            <code>WGS84</code>
          </systemNameCode>
        </geoDeticSystem>
        <longitude>-0.16317</longitude>
        <latitude>51.49000</latitude>
      </NS1:locationPosition>
      <NS1:locationName>Kings Walk</NS1:locationName>
      <NS1:address>
        <addressLine1>Unit G11</addressLine1>
        <addressLine2>Kings Walk Shopping Centre</addressLine2>
        <addressLine3>122 Kings Road</addressLine3>
        <addressLine4>London</addressLine4>
        <county>
          <countyCode>
            <name>Greater London</name>
          </countyCode>
        </county>
        <postcode>SW34TR</postcode>
      </NS1:address>
      <NS1:lCAvailability>true</NS1:lCAvailability>
      <NS1:lCBookingReference>400832600612015008310813RM002ByslTK</NS1:lCBookingReference>
```

```
        <NS1:openingDay>
          <NS1:dayOfWeekType>
            <dayOfWeekCode>
              <name>Monday</name>
            </dayOfWeekCode>
          </NS1:dayOfWeekType>
          <NS1:openingTime>09:30:00</NS1:openingTime>
          <NS1:closingTime>17:30:00</NS1:closingTime>
        </NS1:openingDay>
        <NS1:openingDay>
          <NS1:dayOfWeekType>
            <dayOfWeekCode>
              <name>Tuesday</name>
            </dayOfWeekCode>
          </NS1:dayOfWeekType>
          <NS1:openingTime>09:30:00</NS1:openingTime>
          <NS1:closingTime>17:30:00</NS1:closingTime>
        </NS1:openingDay>
        <NS1:openingDay>
          <NS1:dayOfWeekType>
            <dayOfWeekCode>
              <name>Wednesday</name>
            </dayOfWeekCode>
          </NS1:dayOfWeekType>
          <NS1:openingTime>09:30:00</NS1:openingTime>
          <NS1:closingTime>17:30:00</NS1:closingTime>
        </NS1:openingDay>
        <NS1:openingDay>
          <NS1:dayOfWeekType>
            <dayOfWeekCode>
              <name>Thursday</name>
            </dayOfWeekCode>
          </NS1:dayOfWeekType>
          <NS1:openingTime>09:30:00</NS1:openingTime>
          <NS1:closingTime>17:30:00</NS1:closingTime>
        </NS1:openingDay>
        <NS1:openingDay>
          <NS1:dayOfWeekType>
            <dayOfWeekCode>
              <name>Friday</name>
            </dayOfWeekCode>
          </NS1:dayOfWeekType>
          <NS1:openingTime>09:30:00</NS1:openingTime>
          <NS1:closingTime>17:30:00</NS1:closingTime>
        </NS1:openingDay>
        <NS1:openingDay>
          <NS1:dayOfWeekType>
            <dayOfWeekCode>
              <name>Saturday</name>
            </dayOfWeekCode>
          </NS1:dayOfWeekType>
          <NS1:openingTime>09:30:00</NS1:openingTime>
          <NS1:closingTime>13:00:00</NS1:closingTime>
        </NS1:openingDay>
        <NS1:searchDistance>0.2</NS1:searchDistance>
      </NS1:location>
    </NS1:locations>
  </NS1:GetLCDeliveryLocationsResponse>
```

8.6 SetLCDeliveryLocationRequest

The SetLCDeliveryLocationRequest operation allows customers to confirm that capacity should be reserved for a given location for an estimated delivery date (originally provided in the parameters supplied to the GetLCDeliveryLocations operation).

Please see diagram below for a representation of the SetLCDeliveryLocation request message:

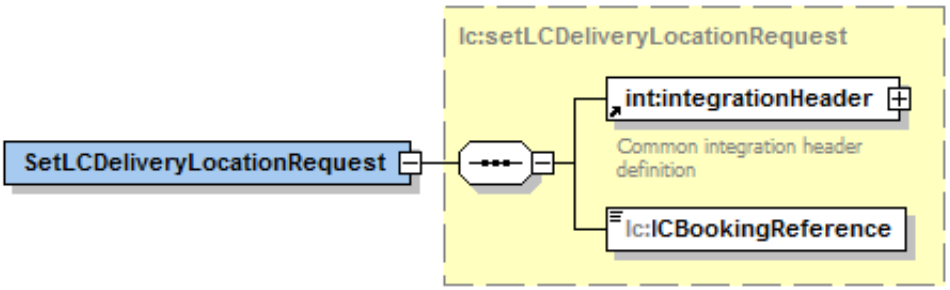


Figure 9 – SetLCDeliveryLocationRequest Message Structure

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

Element	Optional	Description
integrationHeader	No	Please see section 8.4.1.
lCBookingReference	No	This is the reference for the estimated delivery date and location where capacity is to be reserved.

Table 12 – SetLCDeliveryLocationRequest Structure

8.6.1 Example Data

Full XML examples of SOAP requests and responses are provided on the Royal Mail API (Developer) Portal. This section provides a simplified extract to illustrate the SetLCDeliveryLocationRequest only.

```
<v2:SetLCDeliveryLocationRequest>
  <v2:integrationHeader>
    <v1:date>2015-08-02</v1:date>
    <v1:version>1</v1:version>
    <v1:identification>
      <v1:applicationId>1234567890</v1:applicationId>
      <v1:transactionId>3</v1:transactionId>
    </v1:identification>
  </v2:integrationHeader>

<v2:lCBookingReference>70145686388405111203081 3RM003BmF239</v2:lCBookingReference>
  </v2:SetLCDeliveryLocationRequest>
```

8.7 SetLCDeliveryLocationResponse

The SetLCDeliveryLocationResponse operation returns the full details of the chosen location, which is provided as a recap of the information originally provided in the GetLCDeliveryLocations response.

Please see diagram below for a representation of the SetLCDeliveryLocation response message:

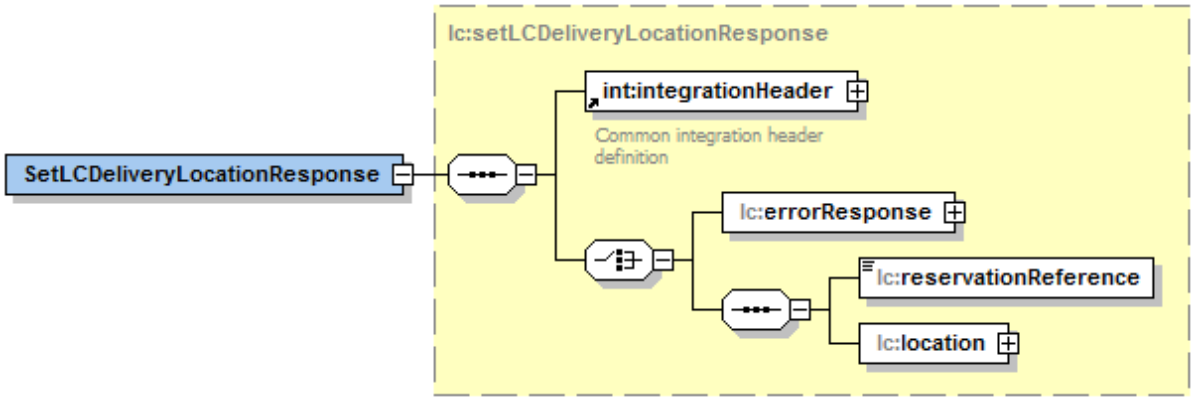


Figure 10 – SetLCDeliveryLocationResponse Message Structure

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

Element	Optional	Description
integrationHeader	No	Please see section 8.5.1.
reservationReference	No	This is the reference for the capacity reservation
location.organisationName	No	This is the organisation name for the location (populated with "Post Office Limited")
location.locationPosition.geoDeticSystem.systemNameCode.code	No (if locationPosition is specified)	The Geodetic system of the position (this always will be WGS84)
location.locationPosition.locationPosition.longitude	Yes	Longitude of the location
location.locationPosition.locationPosition.latitude	Yes	Latitude of the location
location.locationName	No	Name of the location
location.address.addressLine1	No	Address line 1 of the location
location.address.addressLine2	Yes	Address line 2 of the location
location.address.addressLine3	Yes	Address line 3 of the location
location.address.addressLine4	Yes	Address line 4 of the location. This will always be the name of the town the location is in
location.address.county.countyCode.name	Yes	Name of county the location is in
location.address.postcode	Yes	Postcode of the location
location.lCAvailability	Yes	Indicates if the Local Collect service is available at this location. Boolean value

location.ICBookingReference	Yes	Returns the booking reference
location.searchDistance	Yes	Distance the location is from postcode or position as provided in the getLCDeliveryLocationsRequest message (in miles as the crow flies)
For each day the location is open*		
location.openingDay.dayOfWeekType.dayOfWeekCode.name	Yes	Day name
location.openingDay.openingTime	No	Opening time of the location
location.openingDay.closingTime	No	Closing time of the location
location.openingDay.lunchTime.lunchClosingTime	Yes	Time the location closes for lunch
location.openingDay.lunchTime.lunchOpeningTime	Yes	Time the location opens after lunch
errorResponse	No	This element will only be returned in the event that an error condition has occurred. See section 8.5.2 for details.

Table 13 – SetLCDeliveryLocationResponse Structure

*Note: On Local or Public Holidays, individual opening times may vary.

8.7.1 Example Data

Full XML examples of SOAP requests and responses are provided on the Royal Mail API (Developer) Portal. This section provides a simplified extract to illustrate the SetLCDeliveryLocationRequest only.

```
<NS1:SetLCDeliveryLocationResponse
xmlns:NS1="http://www.royalmailgroup.com/API/LocalCollect/V2.0">
  <NS1:integrationHeader>
    <NS2:date xmlns:NS2="http://www.royalmailgroup.com/integration/core/V1">2013-08-02</NS2:date>
    <NS3:version xmlns:NS3="http://www.royalmailgroup.com/integration/core/V1">1.0</NS3:version>
    <NS4:identification xmlns:NS4="http://www.royalmailgroup.com/integration/core/V1">
      <NS4:applicationId>RoyalMailGroup</NS4:applicationId>
      <NS4:transactionId>844ef096-fb70-11e2-97e9-ac14a0b80000</NS4:transactionId>
    </NS4:identification>
  </NS1:integrationHeader>
  <NS1:reservationReference>BmF239-Bfl76</NS1:reservationReference>
  <NS1:location>
    <NS1:organisationName>Badwell Ash</NS1:organisationName>
    <NS1:locationPosition>
      <geoDeticSystem>
        <systemNameCode>
          <code>WGS84</code>
        </systemNameCode>
      </geoDeticSystem>
      <longitude>0.91698</longitude>
      <latitude>52.28368</latitude>
    </NS1:locationPosition>
    <NS1:locationName>Badwell Ash</NS1:locationName>
    <NS1:address>
      <addressLine1>The Street</addressLine1>
      <addressLine2>Badwell Ash</addressLine2>
      <addressLine4>Bury St Edmunds</addressLine4>
```

```
      <county>
        <countyCode>
          <name>Suffolk</name>
        </countyCode>
      </county>
      <postcode>IP313DG</postcode>
    </NS1:address>
    <NS1:lCAvailability>true</NS1:lCAvailability>
    <NS1:openingDay>
      <NS1:dayOfWeekType>
        <dayOfWeekCode>
          <name>Monday</name>
        </dayOfWeekCode>
      </NS1:dayOfWeekType>
      <NS1:openingTime>06:00:00</NS1:openingTime>
      <NS1:closingTime>20:00:00</NS1:closingTime>
    </NS1:openingDay>
    <NS1:openingDay>
      <NS1:dayOfWeekType>
        <dayOfWeekCode>
          <name>Tuesday</name>
        </dayOfWeekCode>
      </NS1:dayOfWeekType>
      <NS1:openingTime>06:00:00</NS1:openingTime>
      <NS1:closingTime>20:00:00</NS1:closingTime>
    </NS1:openingDay>
    <NS1:openingDay>
      <NS1:dayOfWeekType>
        <dayOfWeekCode>
          <name>Wednesday</name>
        </dayOfWeekCode>
      </NS1:dayOfWeekType>
      <NS1:openingTime>06:00:00</NS1:openingTime>
      <NS1:closingTime>20:00:00</NS1:closingTime>
    </NS1:openingDay>
    <NS1:openingDay>
      <NS1:dayOfWeekType>
        <dayOfWeekCode>
          <name>Thursday</name>
        </dayOfWeekCode>
      </NS1:dayOfWeekType>
      <NS1:openingTime>06:00:00</NS1:openingTime>
      <NS1:closingTime>20:00:00</NS1:closingTime>
    </NS1:openingDay>
    <NS1:openingDay>
      <NS1:dayOfWeekType>
        <dayOfWeekCode>
          <name>Friday</name>
        </dayOfWeekCode>
      </NS1:dayOfWeekType>
      <NS1:openingTime>06:00:00</NS1:openingTime>
      <NS1:closingTime>20:00:00</NS1:closingTime>
    </NS1:openingDay>
    <NS1:openingDay>
      <NS1:dayOfWeekType>
        <dayOfWeekCode>
          <name>Saturday</name>
        </dayOfWeekCode>
      </NS1:dayOfWeekType>
      <NS1:openingTime>06:00:00</NS1:openingTime>
      <NS1:closingTime>20:00:00</NS1:closingTime>
    </NS1:openingDay>
    <NS1:openingDay>
      <NS1:dayOfWeekType>
        <dayOfWeekCode>
          <name>Sunday</name>
        </dayOfWeekCode>
```

```
            </NS1:dayOfWeekType>
            <NS1:openingTime>06:00:00</NS1:openingTime>
            <NS1:closingTime>20:00:00</NS1:closingTime>
          </NS1:openingDay>
          <NS1:searchDistance>0.3</NS1:searchDistance>
        </NS1:location>
      </NS1:SetLCDeliveryLocationResponse>
```

9 Error Handling

9.1 Overview

There are two types of errors produced by the Local Collect API:

- **Business Errors** (e.g. invalid postcode, delivery date submitted is in the past etc)
- **Technical Errors / Exceptions** (e.g. database unavailable, failed schema validation etc)

Both sets of errors should be appropriately handled by your systems, and technical details of the error should not be displayed directly to consumers.

9.2 Business Errors

All client data errors are provided in the response part of the GetLCDeliveryLocations or SetLCDeliveryLocation operations, and as such will be accompanied with the standard HTTP successful response code of 200. Section 8.5.2 defines the structure of the business errors which can be returned from calling this API.

Table 14 below lists all business errors which may be returned from the API, and their corresponding HTTP response codes and error messages. Where an X or Y is shown, this will be substituted with the relevant element from the request message (e.g. the description "Postcode X invalid" for E1001 would be returned as "Postcode XX99 9XY invalid").

errorCode		errorCause	errorResolution
code	description		
E1001	Postcode X invalid.	Postcode is not a recognised UK Postcode	Resubmit query with a valid Postcode. Alternatively submit with valid Latitude/Longitude coordinates.
E1002	Latitude X or Longitude Y invalid.	Either Latitude or Longitude is outside UK boundaries.	Resubmit query with location within UK boundaries. Note the UK Bounding Box is taken as: NE 60.85469, 1.76896 SW 49.16209, -8.638 Alternatively submit request with a valid Postcode.
E1003	Delivery Date X invalid.	The delivery date submitted is in the past	Resubmit query with a delivery date in the future. Note that "today's date" is considered invalid.
E1005	Radius X invalid.	The radius must fall in the range 1-99 inclusive,	The radius must fall in the range 1-99 inclusive.
E1007	Invalid request for setLCDeliveryLocationRequest	The LCBookingReference is invalid.	Resubmit the GetLCDeliveryLocationsRequest to obtain a valid set of data
E1008	Invalid request for setLCDeliveryLocationR	This LCBookingReference	Resubmit the GetLCDeliveryLocationsRequest to

	equest	code has already been used.	obtain a new LCBookingReference
E1011	Delivery Date should be on or before X.	The Delivery Date submitted is too far into the future.	Resubmit query with a Delivery Date on or before X.

Table 14 – API Business Errors

9.2.1 Example Data

Please see below for an example of a business error which is returned from calling the GetLCDeliveryLocationsRequest operation with an invalid postcode. Full XML examples of SOAP requests and responses are provided on the Royal Mail API (Developer) Portal.

```
<NS1:GetLCDeliveryLocationsResponse xmlns:NS1="http://www.royalmailgroup.com/API/LocalCollect/V2.0">
        <NS2:integrationHeader xmlns:NS2="http://www.royalmailgroup.com/integration/core/V1">
                <NS2:date>2013-11-25</NS2:date>
                <NS2:version>1.0</NS2:version>
                <NS2:identification>
                        <NS2:applicationId>RoyalMailGroup</NS2:applicationId>
                        <NS2:transactionId>ec883cdc-5609-11e3-9b09-
ac14a0b80000</NS2:transactionId>
                </NS2:identification>
        </NS2:integrationHeader>
        <NS1:errorResponse>
                <NS1:errorCode>
                        <NS1:code>E1001</NS1:code>
                        <NS1:description>Postcode XX99 9XX invalid</NS1:description>
                </NS1:errorCode>
                <NS1:errorCause>Postcode is not a recognised UK Postcode</NS1:errorCause>
                <NS1:errorResolution>Resubmit query with a valid Postcode. Alternatively submit with valid
Latitude/Longitude coordinates</NS1:errorResolution>
        </NS1:errorResponse>
</NS1:GetLCDeliveryLocationsResponse>
```

9.3 Technical Errors / Exceptions

All technical errors will be thrown as a SOAP fault message. All SOAP faults will be accompanied with an HTTP Error Code of 500 along with the information defined in the Table below.

faultcode	faultstring	detail	
		code	description
Server	Internal Error	E0000	Internal Exception Occurred
Server	Service Unavailable	E0001	Service Unavailable
Server	Service Temporarily Unavailable	E0002	Service Temporarily Unavailable
Server	Unknown Service Error	E0003	Service is unavailable due to an unknown reason. Contact RMG Customer Experience Team.
Client	Invalid Request	E0004	Failed Schema Validation

Server	Unknown Service Error	E0005	No Response Received from Business Fulfilment System Web Service (Service is Unavailable or Timeout)
Server	Internal Error	E0009	Business Fulfilment System Returned an Error Response
Server	Service Unavailable	E0010	Configured Throttling Rate for Service Exceeded. Please try again later.

Table 15 – API Technical Errors

9.3.1 Example Data

Please see below for an example of a technical error which is returned from sending in invalid XML. Full XML examples of SOAP requests and responses are provided on the Royal Mail API (Developer) Portal.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lc="http://www.royalmailgroup.com/API/LocalCollect/V2.0">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>Client</faultcode>
      <faultstring>Invalid Request</faultstring>
      <detail>
        <lc:details>
          <lc:code>E0004</lc:code>
          <lc:description>Failed Schema Validation</lc:description>
        </lc:details>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

10 Non-Functional Characteristics

10.1 Availability

10.1.1 Service Hours

The Local Collect 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 Local Collect API being unavailable, customer systems must be able to display an appropriate message to direct consumers to alternative delivery address options. Royal Mail will endeavour to proactively contact customers in the event of an outage to this API.

If you experience issues with the availability of this API please contact a Royal Mail Support representative by visiting the Royal Mail API (Developer) Portal Support pages.

10.2 Performance

Performance testing has validated that the Local Collect API responds to SOAP calls in less than 2 seconds on average when invoked from the edge of Royal Mail's UK data centre. The API does provide a quicker response to GetLCDeliveryLocations calls when Latitude & Longitude (instead of the post code) is used to search for locations. Additionally, a smaller radius in the search criteria, will render a faster response time.

The API does not support caching the results returned. Customers are required to make a new request each time a consumer intends to send the item with the Local Collect enhancement.

10.3 Security

All API service calls will be made using mutually authenticated HTTPS bound SOAP web services. The Local Collect 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.

12 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.

All FAQs specific to the API described in this document are listed below.

12.1 Multipart Deliveries

Question: How do I use the API with multiple parcels for a single order?

Answer: Ideally, customer systems should make one GetLCDeliveryLocations and one SetLCDeliveryLocation call for each parcel within a single order. If it is not possible (e.g. the number of parcels is determined later in the supply chain) for this behaviour to be implemented, it is acceptable for customer systems to only make one GetLCDeliveryLocations and one SetLCDeliveryLocation call for all of the parcels (even if each parcel is expected to arrive on a different day).

12.2 Cancelling Capacity

Question: How do I un-reserve previously reserved capacity (e.g. if a consumer cancels their order prior to it being despatched)?

Answer: It is not possible or necessary to do this – the system automatically adjusts capacity for items that are not delivered.

12.3 Application Compatibility

Question: What Software Development Kits or tools have been proven to work with the Local Collect API?

Answer: The following applications are known to be compatible with the Royal Mail Local Collect API: SoapUI.