



API Specification

Royal Mail Group

Local Collect API V2 (SOAP) Technical User Guide

This API specification details the requirements for integrating with the Local Collect API V2 (SOAP) API. This guide provides information that will assist a retailer's developer in integrating to the local collect API. It specifically covers how the Local Collect V2 API can be used by business customers looking to retrieve Royal Mail and Post Office delivery locations that are convenient to the recipient, and provides the technical information required to build this integration.

20th January 2017

Version 0.1

Contents

1	Document Control	4
1.1	Terms and Abbreviations	4
1.2	Version History	4
2	Overview	5
3	Terms of Use	5
4	Purpose	6
5	What's Functionally Different From the V1 Local Collect API	
?	6	
6	Introduction to Local Collect API Version 2	7
6.1	Overview	7
6.2	Interface Components	7
7	Integrating with the Local Collect API	8
7.1	Terms & Conditions	8
7.2	API Access	9
8	Local Collect Services	10
8.1	Business Services	10
9	Message Structure	10
10	Message Definition and Schemas	12
10.1	Notation	12
10.2	WSDL and Schemas	12
10.3	HTTP Header Information	12
10.3.1	Description	12
10.3.2	Request Message	13
10.3.3	Example Data	13
10.4	GetLocations	13
10.4.1	GetLocationsRequest	14
10.4.2	GeoLocation	14
10.4.3	SOAP Header	15
10.4.3.1 IntegrationHeader	15
10.4.3.2 DebugAudit	16
10.4.4	Example Data	16
10.5	GetLocationsResponse	17

10.5.1	Location	18
10.5.2	Opening Time	18
10.5.3	Lunch Time	19
10.5.4	IntegrationFooter	19
10.5.5	Error Structure	20
10.5.6	Warning Structure	20
10.5.7	Example Data	21
11	Error Handling	25
11.1	Overview	25
11.2	Business Errors	25
11.2.1	Example Data	26
11.3	Technical Errors / Exceptions	27
11.3.1	Example Data	27
12	Non-Functional Characteristics	29
12.1	Availability	29
12.1.1	Service Hours	29
12.1.2	Maintenance Windows	29
12.1.3	Unavailability	29
12.2	Performance	29
12.3	Security	29
14	Frequently Asked Questions	31
14.1	Multipart Deliveries	31
14.2	Application Compatibility	31

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
POL	Post Office Limited
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
0	01/11/2015	Mark Cornforth	Document template created
0.1	20/01/2017	Cliff Dias	Created new version for V2 Local Collect
0.2	25/01/2017	Cliff Dias	Updated based on peer review to remove client secret
0.3	09/01/2017	Cliff Dias	Corrected error response object
1.0	10/01/2017	Cliff Dias	Final Version

Table 2 - Document Version History

2 Overview

The Royal Mail provides a service wherein a retailer's customer can pick up their mail items (Royal Mail Tracked and Special Delivery Guaranteed only) either from a Post Office or Enquiry Office branch. This service is termed Local Collect.

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 or Enquiry Offices.

With over 10,000 Post Offices and 1,500 Enquiry offices providing collection points, consumers have a wide choice of where their parcel is delivered to, and when they collect it.

As part of this service, the Royal Mail exposes a web service API that allows customers to obtain Post Office and Enquiry Office location data.

Local Collect is available as an option with Royal Mail shipping solutions:

1. Despatch Manager Online (DMO)
2. Shipping API (SAPI)
3. Customers own system solution (COSS).

Specifications for these shipping solutions are available from the Royal Mail Customer Solutions team.

3 Terms of Use

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.

Note that, 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 by the customer on their own systems 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.

4 Purpose

This document is to provide Royal Mail customers with guidelines and detailed specifications for integrating with the Local Collect V2 SOAP web service.

The document details:

- The specification for the web service interface for customers who want to query Post Office and Enquiry Office location data
- 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 V2. This document should be read in conjunction with the following artefacts which are available from the Local Collect API V2 page on the [Royal Mail API \(Developer\) Portal](#):

- Local Collect WSDL
- Local Collect XSDs

5 What's Functionally Different From the V1 Local Collect API ?

The differences from the version 1 of the API are listed below:

1. The operation to book a slot is no longer required and has been removed from the service.
2. The operation to retrieve the local collect offices no longer requires an estimated delivery date.
3. The operation to retrieve the locations returns an ordered list of enquiry and post office branches. The list is ordered by distance.
4. The Integration header is part of the SOAP header element as opposed to being part of the SOAP Body as it was in V1.
5. Client-secret is no longer required to invoke this API. The client Id identifies the caller. This decision was taken because the data is publicly available.

6 Introduction to Local Collect API Version 2

6.1 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 and Royal Mail Enquiry offices. With over 10,000 Post Offices and 1500 enquiry office providing collection points, consumers have a wide choice of where their parcel is delivered to, and when they collect it.

The web service, is provided as a SOAP and a REST API using industry standard. The APIs 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 and Royal Mail enquiry offices that offer Local Collect.

6.2 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.

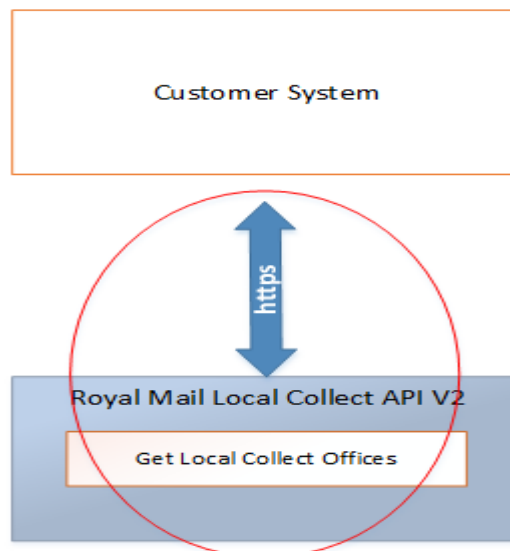


Figure 1 – Local Collect API V2

7 Integrating with the Local Collect API

The high-level process associated with integrating with the Local Collect API V2 is represented and described in the diagram below.

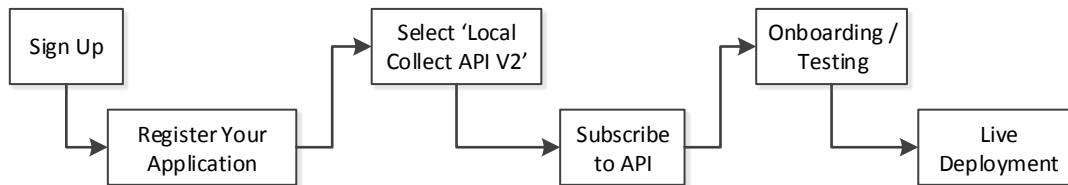


Figure 2 – Process for Integrating with the API

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 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.

7.1 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.

7.2 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 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 is obtained by registering an application on the RMG API Management site.

Please see section 11 for a full list of technical and business error codes which are returned from this API.

8 Local Collect Services

8.1 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	getLocations	Returns a list of Post Offices and Enquiry Offices for a given post code or Geo Location	SOAP over HTTPS	Synchronous Request / Response

Table 3 – Business Services

Royal Mail anticipates that customers will present delivery options and available Post Offices and Enquiry Offices to the consumer as part of the order process within their e-commerce systems. The **GetLocations** operation returns a list of Post Offices and Enquiry Offices ordered by distance from the search location.

Retailers must not cache results from the GetLocations operations as this may result in incorrect information being presented to its customers.

9 Message Structure

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

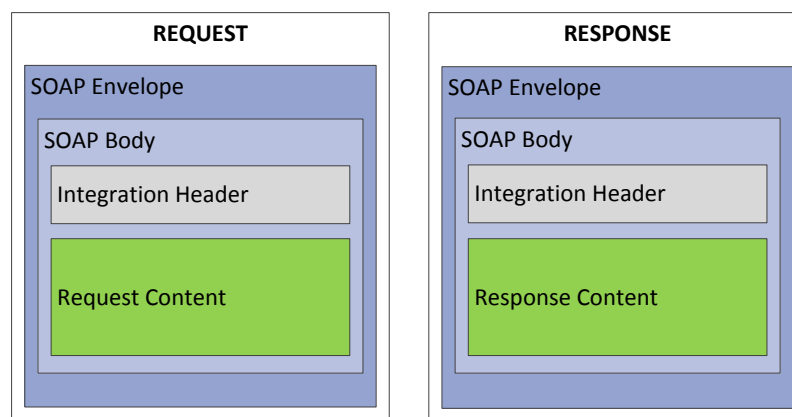


Figure 3 – API Structure

The details of the operation are provided as a WSDL and its related artefacts.

The definition of the data types used within the elements section of each service call is defined within the schemas.

10 Message Definition and Schemas

10.1 Notation

The API is defined using the standard SOAP elements.

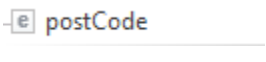
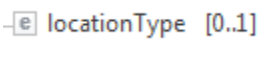


Graphical Representation	Meaning
 <code>postCode</code>	A mandatory element is depicted with no cardinality details.
 <code>locationType [0..1]</code>	An optional element is depicted with the cardinality 0.... E.g. the element depicted in the previous column may have 0 or 1 occurrences.
	The <i>sequence</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

10.2 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 WSDL references several RMG data types in addition to project specific schemas. These are:

XML Schema File	Description
CommonIntegrationSchemaV3_0.xsd	Royal Mail (RMG) schema which defines common objects such as "integration footer".
DatatypesV4.2.xsd	RMG defined simple types that are commonly required such as "date" and "name". This schema is the foundation for the other schema files.
ReferenceDataV4.2.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.

Table 4 – API Schemas

10.3 HTTP Header Information

10.3.1 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.

10.3.2 Request Message

All service requests to this API will be authorised in accordance with the Client ID 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.

Table 5 - HTTP Header Information in the API Request

10.3.3 Example Data

Example request data for the HTTP Header:

Parameter	Value
Accept	application/soap+xml
X-IBM-Client-Id	f0e4f151-2041-4df2-b31d

Table 6 - Example HTTP Header Information for API Request

10.4 GetLocations

The behaviour of the GetLocations operation is to return a list of Post Offices and Enquiry 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).

The branches returned are sorted by distance from the search post code or geo location and will contain a mixture of post offices and enquiry offices.

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

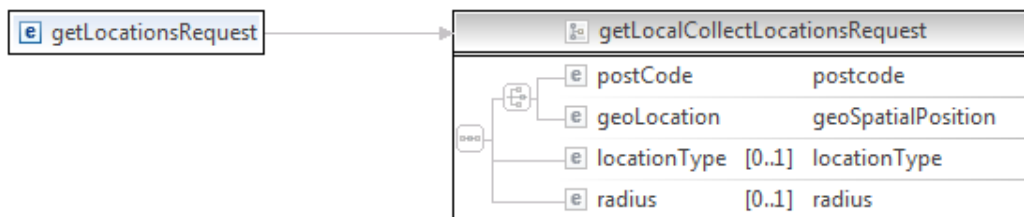


Figure 5 – GetLocationsRequest Message Structure

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

10.4.1 GetLocationsRequest

This element encapsulates the request element with the parameters required to return a valid response.

Element	Optional	Data Type	Constraint	Description
postCode	Choice of either a postCode or a geoLocation	String	[a-zA-Z][0-9a-zA-Z]+\s?[0-9a-zA-Z]*	This is the Postcode of the location to search for Post Offices and Enquiry offices.. Only UK postcodes are supported
geoLocation		geopatialLocation		This is the container element for geospatial information to be used for geospatial searches.
locationType	Yes	String	Enumeration: Post Office Enquiry Office	The type of branch to return. Default is all if element is not specified.
radius	Yes	Integer	Max 20 miles	The radius of the search circle. If not presented defaults to 10 miles

Table 7 – GetLCDeliveryLocationsRequest Structure

10.4.2 GeoLocation

This element encapsulates the latitude and longitude of the branch in the request element

Element	Optional	Data Type	Constraint	Description
latitude	No	geospatial	Base Type = Decimal Fractions = 7 Total Digits = 10	The latitude of the location
longitude	No	geospatial	Base Type = Decimal	The longitude of the location

			Fractions = 7 Total Digits = 10	
--	--	--	---------------------------------------	--

Table 8 - GeoLocation Structure

10.4.3 SOAP Header

The SOAP Header encloses the following elements

10.4.3.1 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:

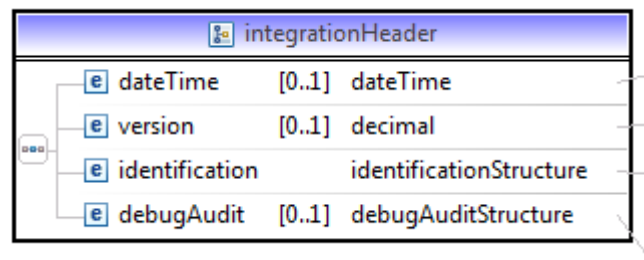


Figure 6 - integrationHeader Structure

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

Element	Optional	Data Type	Constraint	Description
Date	Yes	dateTime		This should be populated with the date the message was generated. The format should be YYYY-MM-DD e.g 2015-08-02
Version	Yes	decimal		The version of the API currently being used (initially 1.0)
identification.applicationId	No	String	Max Length = 50 [a-zA-Z0-9/\-]*	Use the client Id allocated to you when you registered and created your APP within the Royal Mail API Portal.
identification.intermediaryId	Yes	String		This is the ID of an intermediary organisation involved in processing this

				web service
identification.transactionId	No	String	Max Length = 50 [a-zA-Z0-9/\-]*	This is a unique number to identify the transaction number as provided by customer systems

Table 9 – integrationHeader Element (Request)

10.4.3.2 DebugAudit

The purpose of this element is to support the ability to test the service in a live environment. This element is not currently used and exists to future capability

10.4.4 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 GetLocationsRequest for a post code search only.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v3="http://www.royalmailgroup.com/integration/core/V3"
xmlns:v2="http://www.royalmailgroup.com/API/LocalCollect/schema/V2.0">
  <soapenv:Header>
    <v3:integrationHeader>
      <!--Optional:-->
      <dateTime>2017-01-22</dateTime>
      <!--Optional:-->
      <version>2</version>
      <identification>
        <applicationId>1234567890</applicationId>
        <transactionId>24567</transactionId>
      </identification>
    </v3:integrationHeader>
  </soapenv:Header>
  <soapenv:Body>
    <v2:getLocationsRequest>
      <postCode>EC1A1BB</postCode>
      <radius>10</radius>
```



```

    </v2:getLocationsRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

10.5 GetLocationsResponse

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

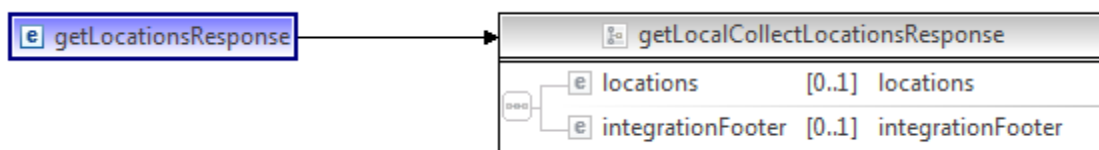


Figure 7 – GeLocationsResponse Message Structure

Element	Optional	Data Type	Constraint	Description
Locations	Yes	Sequence of Location Elements		This element will be returned in the event there is no error
IntegrationFooter	Yes	IntegrationFooter		This element contain any errors or warnings generated

				in the execution of the operation
--	--	--	--	-----------------------------------

Table 10 - GetLocationsResponse Structure

10.5.1 Location

The element encapsulates the branch details.

Element	Optional	Data Type	Constraint	Description
organisationName	No	String	Max Length = 128	Post Office Limited or Royal Mail
locationName	No	String	Max Length = 128	The store name
addressLine1	No	String	Max Length = 256	First line of address
addressLine2	Yes	String	Max Length = 256	Second line of address
addressLine3	Yes	String	Max Length = 256	Third line of address
city	Yes	String	Max Length = 64	The city - populated for Enquiry offices
county	Yes	String	Max Length = 64	
postcode	No	String	[a-zA-Z][0-9a-zA-Z]+\s?[0-9a-zA-Z]*	The post code of the branch
openingHours	No	Sequence of openingTime		The opening hours of the branch
geolocation	No	geopatialLocation		
distance	No	Integer		
facilities	Yes	Facilities		The facilities provided by the branch. Currently, whether customer parking and disability access exists.
locationType	No	String	Enumeration of: Post Office Enquiry Office	

Table 11 - GetLocationsResponse Structure

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

10.5.2 Opening Time

The element encapsulates the opening time for each day of the week.

Element	Optional	Data Type	Constraint	Description
dayOfWeek	No	String	Enumeration of: Monday Tuesday Wednesday Thursday Friday Saturday Sunday	The name of the day of the week
openingTime	No	Time		The opening time of the branch
closingTime	No	Time		The closing time of the branch
lunchHours	Yes	lunchTime		The mid day closing hours if exists.

Table 12 - OpeningTime Structure

10.5.3 Lunch Time

This element provides the details of the mid-day closing hours

Element	Optional	Data Type	Constraint	Description
lunchClosingTime	No	Time		The mid-day closing time of the branch
lunchOpeningTime	No	Time		The mid-day opening time of the branch

Table 13 - LunchTime Structure

10.5.4 IntegrationFooter

All elements are relative to the integration footer element in the response message.

Element	Optional	Data Type	Constraint	Description
errors	Yes	errorStructure		A sequence of errors if exists
warnings	Yes	warningStructure		A sequence of warnings if exists

Table 14 - IntegrationFooter Element (Response)

10.5.5 Error Structure

The error structure element provides the details of the exceptions raised by the service.

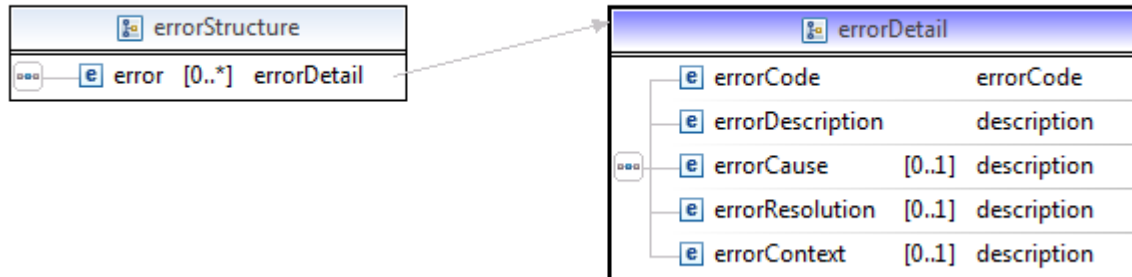


Table 15 - ErrorStructure (Response)

Element	Optional	Data Type	Constraint	Description
error.errorCode	No	String	[E]{1}[0-9]{4}	The error code
error.errorDescription	No	String	Max Length = 128	The error description
error.errorCause	Yes	String	Max Length = 128	The cause of error
error.errorResolution	Yes	String	Max Length = 128	The resolution
error.errorContext	Yes	String	Max Length = 128	The context of the error

Table 16 - ErrorStructure Element (Response)

10.5.6 Warning Structure

The warning structure element provides the details of the warnings raised by the service.

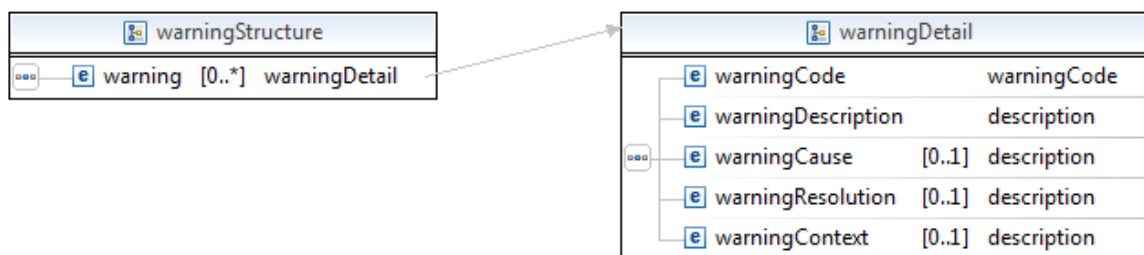


Table 17 – WarningStructure (Response)

Element	Optional	Data Type	Constraint	Description
warning.warningCode	No	String	[E]{1}[0-9]{4}	The warning code
warning.warningDescription	No	String	Max Length = 128	The warning description
warning.warningCause	Yes	String	Max Length = 128	The cause of warning
warning.warningResolution	Yes	String	Max Length = 128	The resolution
Warning.warningContext	Yes	String	Max Length = 128	The context of the warning

Table 18 – WarningStructure Element (Response)

10.5.7 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 GetLocationsResponse only.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <integrationHeader>
      <!--Optional:-->
      <dateTime>2017-01-22</dateTime>
      <!--Optional:-->
      <version>2</version>
      <identification>
        <applicationId>123456789</applicationId>
        <transactionId>1</transactionId>
      </identification>
    </integrationHeader>
  </soapenv:Header>

```

```

<soapenv:Body>
  <getLocationsResponse>
    <!--Optional:-->
    <locations>
      <!--0 to 20 repetitions:-->
      <location>
        <organisationName>Royal Mail</organisationName>
        <locationName>Mount Pleasant Mail Centre</locationName>
        <addressLine1>Farringdon Road</addressLine1>
        <city>London</city>
        <postcode>EC1A1BB</postcode>
        <!--1 to 7 repetitions:-->
        <openingHours>
          <dayOfWeek>Monday</dayOfWeek>
          <openingTime>07:00</openingTime>
          <closingTime>17:00</closingTime>
          <!--Optional:-->
          <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
          </lunchHours>
        </openingHours>
        <openingHours>
          <dayOfWeek>Tuesday</dayOfWeek>
          <openingTime>07:00</openingTime>
          <closingTime>17:00</closingTime>
          <!--Optional:-->
          <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
          </lunchHours>
        </openingHours>
        <openingHours>
          <dayOfWeek>Wednesday</dayOfWeek>
          <openingTime>07:00</openingTime>
          <closingTime>17:00</closingTime>
          <!--Optional:-->
          <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
          </lunchHours>
        </openingHours>
        <openingHours>
          <dayOfWeek>Thursday</dayOfWeek>
          <openingTime>07:00</openingTime>
          <closingTime>17:00</closingTime>
          <!--Optional:-->
          <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
          </lunchHours>
        </openingHours>
        <openingHours>
          <dayOfWeek>Friday</dayOfWeek>
          <openingTime>07:00</openingTime>
          <closingTime>17:00</closingTime>
          <!--Optional:-->
          <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
          </lunchHours>
        </openingHours>
        <openingHours>
          <dayOfWeek>Saturday</dayOfWeek>
          <openingTime>07:00</openingTime>
          <closingTime>17:00</closingTime>
          <!--Optional:-->
          <lunchHours>

```

```

        <lunchClosingTime>13:00</lunchClosingTime>
        <lunchOpeningTime>14:00</lunchOpeningTime>
    </lunchHours>
</openingHours>
<geoLocation>
    <latitude>51.5244569</latitude>
    <longitude>-0.1121266000001051</longitude>
</geoLocation>
<distance>1</distance>
<!--Optional:-->
<facilities>
    <customerParkingIndicator>false</customerParkingIndicator>
    <disabilityIndicator>true</disabilityIndicator>
</facilities>
    <locationType>Enquiry Office</locationType>
</location>
</locations>

<location>
    <organisationName>Post Office Limited</organisationName>
    <locationName>Mount Pleasant</locationName>
    <addressLine1>Rosebery Avenue</addressLine1>
    <city>London</city>
    <postcode>EC1R 4SQ</postcode>
    <!--1 to 7 repetitions:-->
    <openingHours>
        <dayOfWeek>Monday</dayOfWeek>
        <openingTime>07:00</openingTime>
        <closingTime>17:00</closingTime>
        <!--Optional:-->
        <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
        </lunchHours>
    </openingHours>
    <openingHours>
        <dayOfWeek>Tuesday</dayOfWeek>
        <openingTime>07:00</openingTime>
        <closingTime>17:00</closingTime>
        <!--Optional:-->
        <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
        </lunchHours>
    </openingHours>
    <openingHours>
        <dayOfWeek>Wednesday</dayOfWeek>
        <openingTime>07:00</openingTime>
        <closingTime>17:00</closingTime>
        <!--Optional:-->
        <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
        </lunchHours>
    </openingHours>
    <openingHours>
        <dayOfWeek>Thursday</dayOfWeek>
        <openingTime>07:00</openingTime>
        <closingTime>17:00</closingTime>
        <!--Optional:-->
        <lunchHours>
            <lunchClosingTime>13:00</lunchClosingTime>
            <lunchOpeningTime>14:00</lunchOpeningTime>
        </lunchHours>
    </openingHours>
    <openingHours>
        <dayOfWeek>Friday</dayOfWeek>
        <openingTime>07:00</openingTime>

```

```
<closingTime>17:00</closingTime>
<!--Optional:-->
<lunchHours>
  <lunchClosingTime>13:00</lunchClosingTime>
  <lunchOpeningTime>14:00</lunchOpeningTime>
</lunchHours>
</openingHours>
<openingHours>
  <dayOfWeek>Saturday</dayOfWeek>
  <openingTime>07:00</openingTime>
  <closingTime>17:00</closingTime>
  <!--Optional:-->
  <lunchHours>
    <lunchClosingTime>13:00</lunchClosingTime>
    <lunchOpeningTime>14:00</lunchOpeningTime>
  </lunchHours>
</openingHours>
<geoLocation>
  <latitude>51.5244569</latitude>
  <longitude>-0.11212660000001051</longitude>
</geoLocation>
<distance>1</distance>

  <locationType>Post Office</locationType>
</location>

</getLocationsResponse>
</soapenv:Body>
</soapenv:Envelope>
```


11 Error Handling

11.1 Overview

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

- **Business Errors** (e.g. invalid postcode)
- **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.

11.2 Business Errors

All client data errors are provided in the response part of the GetLocations operation, and as such will be accompanied with the standard HTTP successful response code of 200.

Section 10.5.5 defines the structure of the business errors which can be returned from calling this API.

Table 19 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.
E0010	Configured Throttling Rate for Service Exceeded.	Too many requests	Please contact Customer solutions team to discuss throttling rate

Table 19 - API Business Errors

11.2.1 Example Data

Please see below for an example of a business error which is returned from calling the `GetLocationsRequest` operation with an invalid postcode. 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/">
  <soapenv:Header>
    <integrationHeader>
      <!--Optional:-->
      <dateTime>2017-01-22</dateTime>
      <!--Optional:-->
      <version>2</version>
      <identification>
        <applicationId>123456789</applicationId>
        <transactionId>1</transactionId>
      </identification>
    </integrationHeader>
  </soapenv:Header>
  <soapenv:Body>
    <getLocationsResponse>
      <!--Optional:-->
      <integrationFooter>
        <!--Optional:-->
        <errors>
          <!--Zero or more repetitions:-->
          <error>
            <errorCode>E1001</errorCode>
            <errorDescription>Postcode XX99 9XX
invalid</errorDescription>
            <!--Optional:-->
            <errorCause>Postcode is not a recognised UK
Postcode</errorCause>
            <!--Optional:-->
            <errorResolution>Resubmit query with a
valid Postcode. Alternatively submit with valid
Latitude/Longitude coordinates</errorResolution>
          </error>
        </errors>

      </integrationFooter>
    </getLocationsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

11.3 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. E,g, Postcode structure invalid or the radius is outside of range specified in the XSD
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 20 - API Technical Errors

11.3.1 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/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>Client</faultcode>
      <faultstring xml:lang="?">Invalid Request</faultstring>

      <detail>
        <exceptionDetails>

<exceptionApplicationId>123456789</exceptionApplicationId>
        <exceptionTransactionId>1</exceptionTransactionId>
```

```
        <exceptionCode>E0004</exceptionCode>
        <exceptionText>?</exceptionText>
    </exceptionDetails>

</detail>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>
```

12 Non-Functional Characteristics

12.1 Availability

12.1.1 Service Hours

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

12.1.2 Maintenance Windows

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

12.1.3 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.

12.2 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. 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.

12.3 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 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.

14 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.

14.1 Multipart Deliveries

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

Answer: As there is no capacity constraints on the branches returned, the retailer should perform one GetLocations call and use the results throughout the session for the customer.

14.2 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.

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

