Table of Contents

Introduction

MetaPack Options technical documentation.

Download PDF

MetaPack Options

Developer Guide

`Version: 1.28.0`

Audience

The goal of this document is to enable developers to integrate MetaPack Options (MO) into their website. Developers are assumed to have the following knowledge:

- JSON
- REST APIs

It is advantageous for the developer to have knowledge of:

- JavaScript
- AJAX
- HTML
- CSS
- Debugging within a browser environment

Basic overview

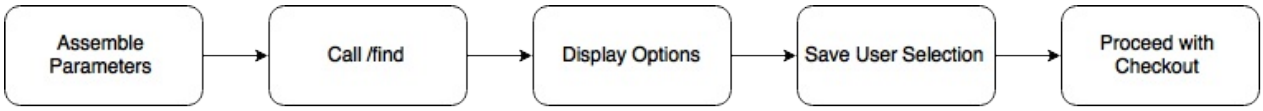
A “delivery option” identifies a **carrier**, its **service**, **charge** and **collection / delivery dates**. On your website, you will want to offer various delivery options, and present them in a way that you find useful. For example, you can present a grid / table of delivery options so that the user can choose their desired delivery date. You could also present them with a list of classes of service (such as “Economy”, “Next day”, “Evening” or “Weekend” etc...).

To determine the possible delivery options effectively, we need the **destination**, **number of parcels**, and **weights** and **dimensions**. If those are not available (and they rarely are), you should use estimates or rely on the default values. The accuracy of the answers will depend on the accuracy of the information you provide, therefore it is recommended you provide as much information as possible.

After you’ve presented the options on the screen, the user will choose one. You'll need to save the shopper’s selection so as to submit the booking at any point later on.

Flow of Work

This is an outline of the workflow. You can start using delivery options before the check-out, if you wish, but the most common place is at the end of the check-out process.



This document doesn't cover the integration with MetaPack Manager (MM) but in general terms, when you do a "create and allocate consignment" call in MM, you'll provide the **booking code** or **service group** which are provided in the MO response as well as the number and characteristics of the items the user confirmed as part of his shopping basket.

The booking code is carrier specific and the service group is more generic and only identifies the service type.

Basic Call Concept

The MetaPack Options find call relies on query string parameters.

Example:

```
GET https://ddo.metapack.com/find?param1=val1&param2=val2&param3=val3 HTTP/1.1
```

We don't require any specific header for this call so a fast and convenient way to test the API is just using the browser.

The response will be a JSON object.

Errors will be identifiable if an HTTP status code different that `200` (OK) is returned. In some cases, the response body will contain a JSON object with details about the error (example: parameter validation errors).

Security

There are 2 different methods with with to authenticate against MetaPack Options:

- API Key
- OAuth2 Bearer Tokens

The first one is simpler but only recommended for back-end integrations. The second one provides the security guarantees needed to enable client-side access to the options `/find` call.

API Key

This key is a Universally Unique Identifier (UUID) and you pass it in the `key` parameter each time you make a call to us. Put simply, it allows us to retrieve the correct configuration for you.

It is recommended you store the key rather than hardcode it into your website, allowing you to configure a different one should you need to.

For billing purposes, we count the number of requests to our service that include your key. You may wish take this information into account when deciding on your integration type. Integrating the API on the client side, via AJAX will expose this key.

OAuth

The OAuth2 Bearer Token Scheme, using JSON Web Tokens was selected as the primary authentication and authorisation method for MetaPack Options.

Using this method, the client does not need to expose secrets in an open channel, if calling the API from the browser directly.

Generate Token

This first stage - which should happen in the backend - involves replacing username / password, using HTTP Basic Authentication, with a JWT token, which represents the user and the permissions that it has in the system, as well as a TTL / expiration time which is parametrised and we recommend should match the typical checkout process duration.

An ill-intentioned end user may still use the browser developer tools to get hold of the token and generate traffic on behalf of the retailer, which we would bill. However, with an appropriate TTL this should be greatly minimised.

This operation is relatively slow, when compared to the actual `/find` call but it's impact on perceived website speed can be completely bypassed by preemptively generating the token before the end user has provided the necessary information to determine delivery options, for example, as soon as the user clicks the "Checkout" button. As soon as the backend process responsible for retrieving the token obtains it, it then needs to be exposed in the user browser, for example in the form of a cookie, so that the client side integration can use it to authorize `/find` calls immediately.

Response

```
HTTP/1.1 200 OK
```


```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczovL2FwaS5zdG9ybXBhdGguY29tL3YxL2FwcGxpY2F0aW9ucy83TWJtdHNZVGFKMFA4RjRuMGx4dUk5Iiwic3ViIjoiL3JldGFpbGVycy80M2ZlMjFlYi04Zjg4LTRkZWEtODUwYy0xNDBhNzI5MDE2ZmQvYXBpS2V5cy8xSUYxQjVZT0FGUjU5QlRWOEY4V0lINkMyIiwic2NvcGUiOiJjYW4tcmV0cmlldmUtbG9jYXRpb25zIGNhbi1zZWFyY2gtbG9jYXRpb25zIGNhbi1tYW5hZ2Utb3BlbmluZy10aW1lcy1ydWxlcyBjYW4tc2VhcmNoLW9wdGlvbnMiLCJqdGkiOiJjZDU2MjhmMS1hY2RmLTQ2NzItYTRlYS0wZWI0NmNhOTRiOGIiLCJpYXQiOjE0ODcyNjE5NzksImV4cCI6MTQ4NzI2NTU3OX0.YTobXCQ1N0EEpqrz4baqnRsMkcAzMoeVpE9DvlRn5T0",
  "token_type": "bearer",
  "expires_in": 3600,
  "scope": "can-retrieve-locations can-search-locations can-manage-opening-times-rules can-search-options"
}
```


The value for `access_token` is what to refer to as the JWT token, the remaining properties are purely informative.

```
POST /oauth HTTP/1.1
Authorization: Basic <base64(apiKeyId:apiKeySecret)>
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
```

```
$ curl -X POST -H "Authorization: Basic <base64(apiKeyId:apiKeySecret)>" -H "Content-Type: application/x-www-form-urlencoded"
-d 'grant_type=client_credentials' "/oauth"
```

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType, "grant_type=client_credentials");
Request request = new Request.Builder()
  .url("/oauth")
  .post(body)
  .addHeader("authorization", "Basic <base64(apiKeyId:apiKeySecret)>")
  .addHeader("content-type", "application/x-www-form-urlencoded")
  .build();

Response response = client.newCall(request).execute();
```

(using OkHttp)

Find Options

With an access token, we can now perform a `/find` call.

Success

See corresponding section.

Failure

Apart from the typical errors, the client will also receive the following, when performing a call after the token expires:

```
HTTP/1.1 401 UNAUTHORIZED
```

```
GET /find?<query string parameters> HTTP/1.1
Authorization: Bearer <bearer token>
```

```
$ curl -X GET -H "Authorization: Bearer <bearer token>" "/find?<query string parameters>"
```

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
  .url("/find?<query string parameters>")
  .get()
  .addHeader("authorization", "Bearer <bearer token>")
  .build();

Response response = client.newCall(request).execute();
```

(using OkHttp)

Error and Outage Handling

You will have to deal with two specific kinds of error:

1. When MetaPack's system cannot be located, is working too slowly or is not working properly.
2. When MetaPack's system is working, but is rejecting requests as they have invalid parameters.

From your customer's perspective, the first one should be irrelevant, seamless. If that situation occurs, there's nothing they can do to correct the situation (and nor can you). Therefore, **reverting to a default delivery option** is the best route forward. It is best that this be configurable (unless the business logic of determining which is applicable is simply too complex).

The second type of error deal with things like invalid postcodes, negative dimensions etc. These should all be corrected (perhaps by the customer).

Options Search

The `/find` call allows you to find delivery options.

URL path	Supported HTTP method
`/find`	`GET`

Request

This section describes the **query string** parameters that the find service makes use of. If an error occurs, an error response is returned instead. See the “Error Responses” section.

MetaPack Options uses one of two option repository systems, designated by *Blue* and *Green*. The following tables also identify which parameters are available in which distribution. Please make sure you use the right parameters which are available for the distribution you are using.

Identification

Query parameter	Format	Required?	Description	Blue	Green
`key`	`string`	yes	Security key (UUID) as provided by MetaPack Support during registration.	✓	✓

Shipping location (sender)

Query parameter	Format	Required?	Description	Default value	Blue	Green
`wh_code`	`string`	yes	Code	`DEFAULT` (*1)	✓	✓
`wh_pc`	`string`	no	Postcode	`EC1R 4PF`	✓	✓
`wh_cc`	`string`	no	ISO 3166 alpha-3 3 letter country code	`GBR`	✓	✓
`wh_l1`	`string`	no	First line of the address	`DEFAULT`	-	✓
`wh_l2`	`string`	no	Second line of the address	`DEFAULT`	-	✓
`wh_l3`	`string`	no	Third line of the address	`DEFAULT`	-	✓
`wh_l4`	`string`	no	Fourth line of the address	`DEFAULT`	-	✓
`wh_street`	`string`	no	Street name	-	✓	-
`wh_house_nr`	`string`	no	House number	-	✓	-
`wh_city`	`string`	no	City	-	✓	-

Notes

1. It is important to identify the shipping location (using the `wh_code` parameter). The default value is only valid if there is effectively a location with code `DEFAULT`.

Customer (receiver)

Query parameter	Format	Required?	Description	Default value	Blue	Green
c_lat	double	conditional	Latitude	-	(*1)	(*1) (*3)
c_long	double	conditional	Longitude	-	(*1)	(*1) (*3)
c_pc	string	conditional	Postcode	-	(*1)	(*1) (*3)
c_cc	string	yes	ISO 3166 alpha-3 3 letter country code	GBR	✓	✓
c_l1	string	no	First line of the address	DEFAULT	-	✓
c_l2	string	no	Second line of the address	DEFAULT	-	✓
c_l3	string	no	Third line of the address	DEFAULT	-	✓
c_l4	string	no	Fourth line of the address	DEFAULT	-	✓
c_street	string	no	Street name	-	✓	-
c_house_nr	string	no	House number	-	✓	-
c_city	string	no	City	-	✓	-
c_district	string	no	District	-	✓	-
c_province	string	no	Province	-	✓	-
c_po_box_set	boolean	no	Indicates whether address is PO Box or not	FALSE	✓	-
c_mobile	string	no	Mobile phone number	-	✓	-
c_email	string	no	Email address	-	✓	-
c_name	string	no	Name	-	✓	-

Notes

1. The values of `c_lat` and `c_long`, if provided, will take precedence over `c_pc` and other address fields. This behaviour is **superseded** by (*2) in **UAT** and the new behaviour will be rolled out into production in **Sept. 2019**.
2. When both `c_lat` / `c_long` and `c_pc` provided then `c_lat` / `c_long` will be used for PUDO/OWNSTORE lookup and `c_pc` will be used for HOME delivery.
3. In order to guarantee a consistent PUDO/OWNSTORE lookup when both `c_lat` / `c_long` and `c_pc` provided then `c_lat` / `l_long` must represent the coordinates of `c_pc`.

Parcel / Consignment / Product

Query parameter	Format	Required?	Description	Default value	Blue	Green
`e_maxdim`	`integer`	no	Length (in cm) of the longest dimension of the biggest parcel (*1)	`0`	✓	✓
`e_maxweight`	`float`	no	Weight (in kg) of the heaviest parcel (*1)	`0`	✓	✓
`e_n`	`integer`	no	Estimated number of parcels	`1`	✓	✓
`e_w`	`float`	no	Estimated weight (in kg) of the whole consignment (*2)	`0.01`	✓	✓
`e_v`	`float` (*3)	no	Consignment value	`-`	✓	✓
`e_v_currency`	`string`	no	Consignment value currency	`-`	✓	✓
`parcelHeight`	`string` (*4)	no	Height of each parcel	`-`	✓	✓
`parcelWidth`	`string` (*4)	no	Width of each parcel	`-`	✓	✓
`parcelDepth`	`string` (*4)	no	Depth of each parcel	`-`	✓	✓
`parcelWeight`	`string` (*4)	no	Weight of each parcel	`-`	✓	✓
`consignmentLevelDetailsFlag`	`boolean`	no	Consignment level details flag (*5)	`true`	✓	✓
`cod_amount`	`double`	no	Cash on delivery amount (*6)	`0.0`	✓	✓
`cod_currency`	`string`	no	Cash on delivery currency	`-`	✓	✓
`hazmat_codes`	`string`	no	Hazardous codes (*7)	`-`	✓	✓
`skus`	`string` (*8)	no	SKU list (*9)	`-`	✓	`-`

Notes

1. If the value is set to `0`, it's not considered in retrieving options.
2. Sum of all the parcel weights.
3. When provided the value will be split equally across all parcels. Example: if value provided is `10.0` and number of parcels is `4` then a value of `2.5` will be assigned to each parcel.
4. Comma separated list of single precision floating point numbers. The number of elements in the list should be equal to parcelCount. Example: `2.0,3,9.2`
5. False means that the parcel details are included in the call and the parcelHeight, parcelWidth, parcelDepth and parcelWeight parameters are parsed and used in the calculation.
6. Applicable if the consignment is for cash on delivery. Default value of `0.0` means the consignment is not cash on delivery.
7. Comma-separated list of codes. Applicable if consignment contains any hazardous goods.
8. List of SKUs. Parcel groups are separated by semicolon, products in the same parcel are separated by comma. Ex.: `x,y;z` Products x and y are part of the same parcel, product z is in a parcel of its own.
9. Mutually exclusive with `e_maxdim`, `e_maxweight`, `e_n`, `e_w`, `parcelHeight`, `parcelWidth`, `parcelDepth`, `parcelWeight` and `consignmentLevelDetailsFlag`.

Filtering

Delivery

Query parameter	Format	Required?	Description	Default value	Blue	Green
`r_t`	`string`	yes	Return type (*1)	-	-	✓
`incgrp`	`string`	no	Include groups (*2)	-	-	✓
`excgrp`	`string`	no	Exclude groups (*3)	-	-	✓
`acceptableCollectionSlots`	`string`	no	Acceptable collection slots (*4)	-	-	✓
`acceptableDeliverySlots`	`string`	no	Acceptable delivery slots (*4)	-	-	✓
`limit`	`int`	no	Limit to this many options (*5)	-	✓	✓
`optionType`	`string`	no	Option type (*6)	`ALL`	✓	✓
`multiCountry`	`boolean`	no	Include locations from multiple countries (*7)	`false`	✓	✓

Notes

1. 3-letter string, values are from: ggg, gss, gsc, lgg, lss, lsc or lsd. The return type which indicates what kind of options search you want to perform. See return types for more information.
2. Comma-separated list of service groups to include. Only services in these groups will be considered. By default, all services are considered.
3. Comma-separated list of service groups to exclude. Services in these groups will not be considered. By default, all services are considered.
4. Two timestamps to mark the respective window, separated by a comma.
5. Not applicable to own stores, if `minown` or `maxown` parameters are provided.
6. Comma-separated list of types of the options required. Avaliable option types: `HOME`, `PUDO`, `OWNSTORE`. Example: `PUDO,OWNSTORE`
7. Different to the one specified via the c_cc parameter.

Locations

Query parameter	Format	Required?	Description	Default value	Blue	Green
`radius`	`int`	no	Radius in metres	`1000`	✓	✓
`minown`	`int`	no	Include at least these many own stores	-	✓	✓
`maxown`	`int`	no	Limit to these many own stores (*1)	`100`	✓	✓
`minpudo`	`int`	no	Include at least these many PUDOs (*2)	-	✓	-
`maxpudo`	`int`	no	Limit to these many PUDOs (*1)	`100`	✓	-
`language`	`string`	Lowercase ISO 639-1 two-letter code	Preferred language for locations (*3)	-	-	-

Notes

1. Setting high value for `max*` parameters may increase response time, therefore it is advised to set those to some reasonable values.

2. The number of PUDOs for which options are returned is dependent upon the carrier-configuration of the back-end system.
3. This parameter will take effect only for locations that have data in given language, all the other locations will be returned in the default language they are available.For more information, see

Verbose mode

Verbose mode provides additional information on request processing. For example togling verbose mode `GEOCODING` (verbose=GEOCODING) for requests that require geocoding will provide additional information on the accuracy of geocoding.

Query parameter	Format	Required?	Description	Default value	Blue	Green
`verbose`	`string`	no	Toggles verbose modes	-	✓	-

Available modes

Name	Description
GEOCODING	Provide additional information on geocoding, like geocoding accuracy
ALLOCATION	Provide additional information on response, like if response contains allocation options for all requested PUDOs

Click `here` for more information on verbose mode.

Other

Query parameter	Format	Required?	Description	Default value	Blue	Green
`user_timezone`	`string`	no	User time zone	`-`	✓	`-`
custom1 ... custom10	`string`	no	10 custom fields	`-`	`-`	✓

Verbose Mode

During the integration stage, it is often useful for the integrator to retrieve meta-data about the find calls, as well as the response data.

Currently, there are 2 verbose modes that can be enabled via a request [query string] parameter.

The response data is available as a top level response property (`info`), at the same level as `header` and `results` .

Geocoding

In find calls that include pickup points (PUDOS / own stores), if the user provides an address, we then need to geocode the provided address.

Geocoding is an extremely complex problem to solve and it is not always possible to retrieve accurate coordinates. The lowest common denominator is a set of reference coordinates for the provided country. The geocoding verbose mode makes the information on the accuracy level of the coordinates used to retrieve pickup points transparent to the caller.

The possibilities are: `ADDRESS` and `COUNTRY` .

Request

```
GET ...&verbose=GEOCODING
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "header": { /* ... */ },
  "results": [ /* ... */ ],
  "info": {
    "geocoding": {
      "accuracy": "COUNTRY"
    }
  }
}
```

Allocation

Again, in find calls that include pickup points (PUDOS / own stores), DDO needs to perform multiple simultaneous calls to a backend system to determine the full list of delivery options, corresponding to the number of relevant pick up points and the home delivery itself, if relevant, depending on the provided `optionType` value(s).

There are a few things that can happen to each of these concurrent calls:

- Success (manage to populate options for a specific address);
- Failure
  - Address not covered by the services configured in the backend system (ex.: routing, missing parameters, failed validation, etc...)
  - Network failure (timeout, network partition, routing issues, etc...)
  - Backend system unavailable

If all of the requests succeed, the response is considered `COMPLETE`, if none of them do, we identify the status as `EMPTY`. The other status is `PARTIAL`, when at least one of the requests succeed in generating options.

Request

```
GET ...&verbose=ALLOCATION
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "header": { /* ... */ },
  "results": [ /* ... */ ],
  "info": {
    "allocation": {
      "status": "PARTIAL"
    }
  }
}
```

Response

This identifies a successful response.

Field	Format	Description
`header`	JSON object	Request and monitoring data.
`results`	JSON Array	The delivery options to present to the customer for selection.

Header

Field	Format	Description	Example value
`inputParameters`	JSON object	A map containing all input parameters.	`{"c_pc":["EC1R 4PF"],"r_t":["lsc"],"wh_l2":["London"], "r_t":["ggg"]}`
`requestDate`	`date`	The date & time the request arrived.	`"requestDate":"2014-10-31T15:07:16.241Z"`
`requestId`	`UUID`	A UUID which uniquely identifies the request. This can be used by MetaPack Support.	`"requestId":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"`
`responseDate`	`date`	The date & time the response was ready and started being sent to the client.	`"responseDate":"2014-10-31T15:07:17.221Z"`

Results

Eeach delivery option in the results array is defined by following properties.

Field	Format	Description	Example value
`bookingCode`	`string`	The code to use to identify this specific delivery option, when integrating with MetaPack Options	-
`carrierCode`	`string`	A code that identifies the carrier	-
`carrierServiceCode`	`string`	A code representing the specific service offered by the specific carrier	-
`carrierServiceName`	`string`	The name of the carrier service (as known by MetaPack)	-
`collection`	JSON object with two properties, "from" and "to"	The period of time the collection will be made from the shipping location	-
`collection.from`	`date`	The time the carrier arrives at the shipping location	-
`collection.to`	`date`	The time the carrier departs the shipping location	-
`delivery`	JSON object with two properties, "from" and "to"	The period of time during which the carrier might deliver the consignment	-
`delivery.from`	`date`	The earliest possible delivery date	-
`delivery.to`	`date`	The latest possible delivery date. If it's a nominated day service, this will be the same as the "from"	-
		The full name of the service (which	

`fullName`	`string`	includes the carrier's name). This might be useful for presenting on the screen	-
`groupCodes[]`	array of strings	A list of all the service groups this service is a member of	-
`cutOffDateTime`	`datetime`	The cutoff date and time for the carrier in UTC.	`"2019-08-16T09:30:59.999Z"`
`shippingCharge`	`float`	The amount the customer needs to pay for this service	-
`lat`	double-precision float	The latitude of the location	-
`long`	double-precision float	The longitude of the location	-
`storeTimes`	JSON object with properties "monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday"	The opening and closing times per weekday. If the result references a home delivery option, this value is an empty string	-
`storeTimes.monday`	array of time range formatted strings	The opening and closing times for Monday. Time ranges are formatted as `"hh:mm-hh:mm"`	`"Monday":[ "06:30-18:30" ]`
`storeTimes.tuesday`	array of time range formatted strings	The opening and closing times for Tuesday. Time ranges are formatted as `"hh:mm-hh:mm"`	`"Tuesday":[ "08:00-10:00", “12:00-22:00” ]`
`storeTimes.wednesday`	array of time range formatted strings	The opening and closing times for Wednesday. Time ranges are formatted as `"hh:mm-hh:mm"`	-
`storeTimes.thursday`	array of time range formatted strings	The opening and closing times for Thursday. Time ranges are formatted as `"hh:mm-hh:mm"`	-
`storeTimes.friday`	array of time range formatted strings	The opening and closing times for Friday. Time ranges are formatted as `"hh:mm-hh:mm"`	-
`storeTimes.saturday`	array of time range formatted strings	The opening and closing times for Saturday. Time ranges are formatted as `"hh:mm-hh:mm"`	-
`storeTimes.sunday`	array of time range formatted strings	The opening and closing times for Sunday. Time ranges are formatted as `"hh:mm-hh:mm"`	-
`address`	`string`	The pick up delivery location address (*1)	-
`postcode`	`string`	The postcode of the pick up delivery location	-
`distance`	JSON object with properties "value", "unit"	The approximate distance from customer address to the delivery location	-
`distance.value`	`double`	The approximate distance value (*2)	-
`distance.unit`	`string`	The unit of the distance provided in the response	`m`
`storeId`	`string`	Id of store	-
`storeName`	`string`	Store Name	-
`photoUrls`	array of url strings	Urls for store photos	-
`logoUrl`	`string`	Url for logo	-
`hasDisabledAccess`	`boolean`	Boolean specifying whether store had disabled access or not	`false`

`telephoneNumber`	`string`	Store telephone number	-
`locationProviderId`	`string` (UUID)	Location Provider ID	-
`optionType`	`string`	Type of option (PUDO,OWNSTORE,HOME)	PUDO

Notes

1. If the result references a home delivery option, this value is an empty string.
2. For home deliveries the value will be 0.

Info

Additional information will be returned from system only if any of the verbose modes is togled.

Field	Format	Description	Example value
`geocoding.accuracy`	`string` (*1)	Geocoding accuracy	"ADDRESS"

Notes

1. Geocoding accuracy is classified as `"ADDRESS"`, `"COUNTRY"` and `"NOT_FOUND"`. Accuracy `"COUNTRY"` is returned whenever geocoding can only match country, for all other matches accuracy `"ADDRESS"` is returned. If nothing matched then accuracy `"NOT_FOUND"` is returned.

Success

This is an example of a JSON response as returned from the find call. It contains a header object with the input parameters, request and response timestamps and request id and a simplified response object array, containing an example of a home delivery and a pick up location.

Request

```
GET /find?wh_code=DEFAULT&wh_l1=12%2F16+Laystall+Street&wh_l2=London&wh_pc=EC1R+4PF&wh_cc=GBR&c_phone=0123+3476&c_l1=12%2F16+L
aystall+Street&c_l2=London&c_pc=EC1R+4PF&c_cc=GBR&e_v=51.45&e_n=1&e_w=1.67&e_maxweight=0&e_maxdim=0&r_t=lsc&r_f=json&key=UUID(
Retailer specific)&radius=5000
Host: dmo.metapack.com
```

Response

```
HTTP/1.1 200 OK
```

Option on Green Response:


```
{
  "header": {
    "requestId": "b0ee137a-be28-48eb-9bc0-bf57fbe5e827",
    "requestDate": "2015-08-14T15:28:09.101Z",
    "inputParameters": {
      "c_pc": [
        "N194AS"
      ],
      "orderType": [
        "OWNSTORE"
      ],
      "parcelDepth": [
        "2"
      ],
      "r_t": [
        "lsc"
      ],
      "consignmentLevelDetailsFlag": [
        "false"
      ],
      "parcelHeight": [
        "10"
      ],
      "c_cc": [
        "GBR"
      ],
      "wh_cc": [
        "GBR"
      ],
      "parcelWidth": [
        "15"
      ],
      "parcelWeight": [
        "1"
      ],
      "wh_pc": [
        "WC1X8XZ"
      ],
      "radius": [
        "250000"
      ],
```

```
      "wh_code": [
        "DDOWARETIM"
      ],
      "key": [
        "UUID"
      ]
    },
    "responseDate": "2015-08-14T15:28:13.236Z"
  },
  "results": [
    {
      "delivery": {
        "from": "2015-08-17T05:31:00.000Z",
        "to": "2015-08-17T09:30:59.999Z"
      },
      "cutOffDateTime": "2019-08-16T09:30:59.999Z",
      "groupCodes": [
        "UPS"
      ],
      "address": "",
      "photoUrls": [],
      "telephoneNumber": null,
      "carrierServiceName": "EXPRESS PRE 1030",
      "distance": {
        "unit": "m",
        "value": 0
      },
      "postcode": "",
      "fullName": "UPS EXPRESS PRE 1030",
      "description": null,
      "collection": {
        "from": "2015-08-14T17:30:00.000Z",
        "to": "2015-08-14T18:30:59.999Z"
      },
      "storeId": null,
      "carrierServiceCode": "UPSEXPRESS1030",
      "long": null,
      "logoUrl": null,
      "storeTimes": {},
      "carrierCode": "UPS",
      "hasDisabledAccess": false,
      "storeName": "",
      "shippingCharge": 0,
      "bookingCode": "UPSEXPRESS1030/2015-08-14/*-*/*/*-*",
      "lat": null
    },
    {
      "delivery": {
        "from": "2015-08-15T08:00:00.000Z",
        "to": "2015-08-15T16:00:59.999Z"
      },
      "cutOffDateTime": "2019-08-16T09:30:59.999Z",
      "groupCodes": [
        ""
      ],
      "address": "Premier Food Hall, 774-776 Holloway Road, London",
      "photoUrls": [],
      "telephoneNumber": null,
      "carrierServiceName": "24 Hour Collect+ Delivery to Store 00279m 0.17 miles\nPremier Food Hall\n774-776 Holloway Road  \
nLondon\n\nN19 3JQ",
      "distance": {
        "unit": "m",
        "value": 238
      },
      "postcode": "N19 3JQ",
      "fullName": "Yodel Home 24 Hour Collect+ Delivery to Store 00279m 0.17 miles\nPremier Food Hall\n774-776 Holloway Road
\nLondon\n\nN19 3JQ",
      "description": null,
      "collection": {
        "from": "2015-08-14T17:30:00.000Z",
        "to": "2015-08-14T18:30:59.999Z"
      },
```

```
      "storeId": null,
      "carrierServiceCode": "HDN24MERCHTOSTORE_Y3S5_51557_2001",
      "long": -0.1307003085686866,
      "logoUrl": null,
      "storeTimes": {
        "monday": [
          "08:00-23:00"
        ],
        "tuesday": [
          "08:00-23:00"
        ],
        "wednesday": [
          "08:00-23:00"
        ],
        "thursday": [
          "08:00-23:00"
        ],
        "friday": [
          "08:00-23:00"
        ],
        "saturday": [
          "08:00-23:00"
        ],
        "sunday": [
          "08:00-23:00"
        ]
      },
      "carrierCode": "HDN",
      "hasDisabledAccess": false,
      "storeName": null,
      "shippingCharge": 0,
      "bookingCode": "HDN24MERCHTOSTORE_Y3S5_51557_2001/2015-08-14/*-*/*/*-*",
      "lat": 51.56426648599607
    }
  ]
}
```


Options On Blue Response:


```
{
   "header":{
      "requestId":"e868e706-2a1d-4efa-b151-d8f7eee21cd0",
      "inputParameters":{
         "c_pc":[
            "10117"
         ],
         "r_t":[
            "lpf"
         ],
         "maxpudo":[
            "1"
         ],
         "c_cc":[
            "DEU"
         ],
         "wh_cc":[
            "GBR"
         ],
         "e_n":[
            "1"
         ],
         "r_f":[
            "json"
         ],
         "wh_pc":[
            "EC1R 4PF"
         ],
         "e_w":[
            "1"
         ],
```

```
            "radius":[
               "10000"
            ],
            "wh_code":[
               "DEV"
            ],
            "e_v":[
               "89.10"
            ],
            "wh_street":[
               "12/16 Laystall Street"
            ],
            "key":[
               "5cab61b4-d4f4-410d-a012-64c0ad51d86d"
            ],
            "wh_city":[
               "London"
            ]
         },
         "requestDate":"2017-02-22T15:27:02.082Z",
         "responseDate":"2017-02-22T15:27:04.089Z"
      },
      "results":[
         {
            "bookingCode":null,
            "carrierCode":"WNC-01",
            "carrierServiceCode":"XSCDD DE",
            "carrierServiceName":"5087572",
            "collection":{
               "from":"2017-02-24T00:00:00.000Z",
               "to":"2017-02-24T23:59:59.999Z"
            },
            "delivery":{
               "from":"2017-03-01T00:00:00.000Z",
               "to":"2017-03-01T23:59:59.000Z"
            },
            "cutOffDateTime": null,
            "fullName":"5087572",
            "groupCodes":[
               "PUDO"
            ],
            "shippingCharge":0.0,
            "storeTimes":null,
            "distance":{
               "value":0.0,
               "unit":"m"
            },
            "storeId":null,
            "storeName":null,
            "photoUrls":null,
            "logoUrl":null,
            "hasDisabledAccess":false,
            "telephoneNumber":null,
            "locationProviderId":null,
            "optionType":"HOME",
            "hazmat":null,
            "cod":null,
            "lat":52.5155098,
            "long":13.3847539,
            "postcode":"",
            "address":""
         },
         {
            "bookingCode":null,
            "carrierCode":"WNC-01",
            "carrierServiceCode":"XSCDD DE",
            "carrierServiceName":"5087572",
            "collection":{
               "from":"2017-02-24T00:00:00.000Z",
               "to":"2017-02-24T23:59:59.999Z"
            },
            "delivery":{
```

```
            "from":"2017-03-01T00:00:00.000Z",
            "to":"2017-03-01T23:59:59.000Z"
         },
         "cutOffDateTime": null,
         "fullName":"5087572",
         "groupCodes":[
            "PUDO"
         ],
         "shippingCharge":0.0,
         "storeTimes":{
            "monday":[
               "09:30-19:00"
            ],
            "tuesday":[
               "09:30-19:00"
            ],
            "wednesday":[
               "09:30-19:00"
            ],
            "thursday":[
               "09:30-19:00"
            ],
            "friday":[
               "09:30-19:00"
            ],
            "saturday":[
               "09:00-13:00"
            ],
            "sunday":null
         },
         "distance":{
            "value":398.0,
            "unit":"m"
         },
         "storeId":"8003-4115119",
         "storeName":"Postfiliale 502",
         "photoUrls":[

         ],
         "logoUrl":null,
         "hasDisabledAccess":true,
         "telephoneNumber":null,
         "locationProviderId":"2f329336-4ebf-4136-86ef-5f0f7c59c000",
         "optionType":"PUDO",
         "hazmat":null,
         "cod":null,
         "lat":52.5156313,
         "long":13.3906333,
         "postcode":"10117",
         "address":"Postfiliale 502 Behrenstr. 29, Mitte, Berlin, DEU"
      }
   ]
}
```

Errors

API parameter validation

```
HTTP/1.1 400 BAD REQUEST
```

Ex.:

`radius=abc`

```
{
   "header":{
      "requestId":"f2f84566-4104-40a5-8040-bd6c41a25acd",
      "inputParameters":{
         "c_pc":[
            "vd65 7we"
         ],
         "r_t":[
            "lsc"
         ],
         "wh_l2":[
            "London"
         ],
         "e_maxdim":[
            "0"
         ],
         "c_l2":[
            "London"
         ],
         "wh_l1":[
            "12/16 Laystall Street"
         ],
         "c_l1":[
            "12/16 Laystall Street"
         ],
         "e_maxweight":[
            "0"
         ],
         "c_phone":[
            "0123 3476"
         ],
         "c_cc":[
            "GBR"
         ],
         "wh_cc":[
            "GBR"
         ],
         "e_n":[
            "1"
         ],
         "r_f":[
            "json"
         ],
         "wh_pc":[
            "EC1R 4PF"
         ],
         "e_w":[
            "1.67"
         ],
         "radius":[
            "abc"
         ],
         "wh_code":[
            "DEFAULT"
         ],
         "e_v":[
            "51.45"
         ],
         "key":[
            "UUID"
         ]
      },
      "requestDate":"2016-12-14T11:12:46.500Z",
      "responseDate":"2016-12-14T11:12:46.508Z"
   },
   "errorMessage":"radius must be an integer"
}
```


`r_t=abc`

```
{
   "header":{
      "requestId":"2ca8a1df-38f3-4c79-be23-38fb2358ae9e",
      "inputParameters":{
         "c_pc":[
            "vd65 7we"
         ],
         "r_t":[
            "abc"
         ],
         "wh_l2":[
            "London"
         ],
         "e_maxdim":[
            "0"
         ],
         "c_l2":[
            "London"
         ],
         "wh_l1":[
            "12/16 Laystall Street"
         ],
         "c_l1":[
            "12/16 Laystall Street"
         ],
         "e_maxweight":[
            "0"
         ],
         "c_phone":[
            "0123 3476"
         ],
         "c_cc":[
            "GBR"
         ],
         "wh_cc":[
            "GBR"
         ],
         "e_n":[
            "1"
         ],
         "r_f":[
            "json"
         ],
         "wh_pc":[
            "EC1R 4PF"
         ],
         "e_w":[
            "1.67"
         ],
         "radius":[
            "1000"
         ],
         "wh_code":[
            "DEFAULT"
         ],
         "e_v":[
            "51.45"
         ],
         "key":[
            "7cda774d-8e64-432c-9781-288530b0e09d"
         ]
      },
      "requestDate":"2016-12-14T11:14:19.516Z",
      "responseDate":"2016-12-14T11:14:19.616Z"
   },
   "errorMessage":"return type value must be either of ggg, gss, gsc, lgg, lss, lsc, lsd, lpf or lpa"
}
```


Authentication

Invalid or missing credentials:

```
HTTP/1.1 401 UNAUTHORIZED
```

Authorisation

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Internal errors (MetaPack)

`HTTP 5XX`

Ex.:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Strategies

Estimating the number of parcels

The number of parcels can affect which carriers are selectable. For example, some carriers will not allow more than 30 parcels to be placed on a single order. However, the number of parcels can also have a direct cost to the retailer. Therefore it could be important to calculate an estimate of the number of parcels that will be needed to ship the order.

It will be dependent on the items, of course. If you know the rough sizes of the items, and you know the sizes of the parcels that the retailer uses, you can estimate the number of parcels. Even with the precise dimensions of both, it is only ever an estimate at this stage. To keep things simple, it's probably worthwhile using an "average number of items per parcel" as a basis.

Total weight of consignment

Similar to the number of parcels, the total weight can also affect the retailer's costs and the available carriers. In the rare event that you actually know the weights of the items, you can work out the total weight. However, it is also just an estimate – packing materials (such as padding) and the boxes themselves) also affect the total shipping weight.

Where the weights are completely unknown, you can supply the "average total consignment weight" to all orders. If you are aware of particularly large products (such as a washing machine) you can add suitably large numbers to it.

Can't estimate anything

Sometimes, it's just not possible to estimate weights, dimensions or the number of parcels. In this case, reverting to the defaults is about as good as you can do. The actual costs to the retailer will be wrong, as might be the actual carrier selection. However, as you've not estimated the postcode, the availability of the classes of service (or "service groups", such as "Next day", "Evening" etc.) will usually be accurate enough.

Charges

The charges associated with services are obtained from MetaPack's Delivery Manager. They are configured by the retailer and can change from time to time. Often, they will be based more on the class of service, rather than the actual cost the carrier charges the retailer. For example, the retailer may choose to offer free delivery for economy services, and a flat rate for Saturday services.

Which shipping location?

With larger retailers, orders might be fulfilled from multiple shipping locations. You can either split the order into different parts (one part for each shipping location), or choose one of them. Another alternative is to have an entirely different, logical shipping location configured that is only used by the delivery options system.

Which delivery address?

If you're presenting options before the customer has entered their required delivery address, you can use a default postcode. It's best to choose a typically small-town postcode, rather than one in a large city or in the middle of the countryside. At this point, it's only an estimate after all.

If you provide both "lat/long" and "postcode" for recipient, "lat/long" will take precedence to find delivery options.

Return Types

The return type affects how the "shipment" related properties of the delivery options (*1) 2 main categories are "grids" (`g**`) and "lists" (`l**`).

Notes

1. As opposed to the "location" related properties which are not dependent of return types.

Grids

Grids represent a distribution of one option / service per day over a **20 day period**.

Code	Meaning
`ggg`	Cheapest option for each service **group** (ordered by group)
`gsc`	First option for each **service** (ordered by **cost**)
`gss`	First option for each **service** (ordered by **score**)

By "expanding groups", we mean that if a service belongs to multiple groups, there will be one option representing each of those combinations.

Lists

Code	Meaning
`lgg`	Cheapest option for each service **group** (ordered by **group**)
`lsc`	First option for each **service** (cheapest delivery **cost** first)
`lsd`	First option for each delivery **date** (cheapest delivery cost first)
`lss`	First option for each **service** (ordered by **score**)

Examples

These examples are an internal illustration of how the different return types affect delivery options generation, ordering and filtering.

Different backend systems may have variations in how the methods are implemented and also may only provide a subset of these types.

When making an implementation / product / UX decision related to the return type parameter, please make sure you seek advise and discuss it with a MetaPack implementation analyst.

Service Definitions

Group (*1)	Service	Cost	Score	Type (*2)
X	A	9	1	24h
X	B	8	2	48h
X	C	7	3	24h
Y	C	7	3	24h
Y	D	6	4	48h
Y	E	5	5	48h

Notes

1. Carrier Service Group
2. Carrier Service Type

Grids

Services by Cost / Score

`gsc` / `gss`

Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
-	TODAY	A [X] C [X, Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]
A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]
A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]	A [X] B [X] C [X, Y] D [Y] E [Y]

Groups by Cost

`ggg`

Groups are expanded: if a service belongs to multiple groups, there will be one option per service + unique group combination which is then filtered by selecting the cheapest one, per group in each day.

Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
-	TODAY	`A [X]` `C [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`
`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`
`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`	`C [X]` `E [Y]`

Lists

Groups by Cost

`lgg`

`C [X]` , `E [Y]` (earliest collection slots)

Services by Cost

`lsc`

`E [Y]` , `D [Y]` , `C [X]` , `C [Y]` , `B [X]` , `A [X]` (earliest collection slots)

Services by Date

`lsd`

`A [X]` , `C [X, Y]` , `B [X]` , `D [Y]` , `E [Y]` (earliest **delivery** slots)

Services by Score

`lss`

`E [Y]` , `D [Y]` , `C [Y]` , `C [X]` , `A [X]` , `B [X]` (earliest collection slots)

Internal API

Managing Retailer Configurations

Retailer configurations link a ddo find call client, which can be generalised as a "retailer" to the type of integration and the credentials ddo needs to provide to the corresponding backend system to be able to retrieve delivery options on behalf of the client.

Creating Retailer Configurations

To create a new retailer configuration we need to do a `POST` request to `/configurations` with the `application/json` content-type request header and a JSON object containing the specified properties in the request body.

Required Permission

`can-create-configurations`

See Auth* (scopes).

Object Properties

Name	Mandatory	Type	Length
`retailerId`	`false`	`UUID`	36
`apiKey`	`false`	`UUID`	36
`repositoryType`	`true`	Enumeration (*1)	10
`user`	`true`	`string`	200
`password`	`true`	`string`	200
`url`	`true`	`string`	200
`enabled`	`true`	`boolean`	-
`oauthAuthentication` (*2)	`false`	`boolean`	-

Notes

1. Possibilities: `DM` , `SMR` , `SMRREST` .
2. If not set or false then retailer will be configured to use apiKey authentication.

Request

```
POST /configurations
Authorization: Bearer <token>
Content-Type: application/json
```

```
{
  "retailerId": "ed3dd9a9-4abd-4d1e-b2f1-0eecbb600ec9",
  "apiKey": "cb3dbb84-23a7-4c19-8705-4ff418356de8",
  "repositoryType": "SMR",
  "user": "test",
  "password": "test",
  "url": "https://test.mpm.metapack.com/BlackBox",
  "enabled": true
}
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
  "rel": "self",
  "href": "/configurations/<configuration ID>"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Retrieving Retailer Configurations

Required Permission

`can-read-configurations`

See Auth* (scopes).

Request

```
GET /configurations/<configuration ID>
Authorization: Bearer <token>
```

Response

```
HTTP/1.1 200 OK
```

Example response body:

```
{
  "configurationId": "ff99f5be-53c7-48b0-831c-82efd3cadf19",
  "user": "test",
  "url": "http://test1.metapack.com/api/4.x/services",
  "enabled": true,
  "repositoryType": "DM",
  "retailerId": "ff99f5be-53c7-48b0-831c-82efd3cadf19",
  "apiKey": "ff99f5be-53c7-48b0-831c-82efd3cadf19",
  "dateCreated": "2014-08-15T15:55Z",
  "links": [
    {
      "rel": "self",
      "href": "/configurations/ff99f5be-53c7-48b0-831c-82efd3cadf19"
    }
  ]
}
```

Failure

Unexistent configuration ID:

```
HTTP/1.1 404 NOT FOUND
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Listing Retailer Configurations

Required Permission

`can-list-configurations`

See Auth* (scopes).

Request

```
GET /configurations
Authorization: Bearer <token>
```

Response

```
HTTP/1.1 200 OK
```

Example response body:

```
[
  {
    "configurationId": "ff93445d-ade3-457d-aba2-b0139b7799cf",
    "user": "test-deleteme",
    "url": "http://dm-delta.metapack.com/api/5.x/services",
    "enabled": true,
    "repositoryType": "DM",
    "retailerId": "ff93445d-ade3-457d-aba2-b0139b7799cf",
    "apiKey": "ff93445d-ade3-457d-aba2-b0139b7799cf",
    "dateCreated": "2016-05-17T13:33Z",
    "links": [
      {
        "rel": "self",
        "href": "/configurations/ff93445d-ade3-457d-aba2-b0139b7799cf"
      }
    ]
  },
  {
    "configurationId": "ff99f5be-53c7-48b0-831c-82efd3cadf19",
    "user": "",
    "url": "http://test1.metapack.com/api/4.x/services",
    "enabled": true,
    "repositoryType": "DM",
    "retailerId": "ff99f5be-53c7-48b0-831c-82efd3cadf19",
    "apiKey": "ff99f5be-53c7-48b0-831c-82efd3cadf19",
    "dateCreated": "2014-08-15T15:55Z",
    "links": [
      {
        "rel": "self",
        "href": "/configurations/ff99f5be-53c7-48b0-831c-82efd3cadf19"
      }
    ]
  },
  /* ... */
]
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Updating Retailer Configurations

Required Permission

`can-create-configurations`

See Auth* (scopes).

Request

```
PUT /configurations/<configuration ID>
Authorization: Bearer <token>
Content-Type: application/json
```

Example request body:

```
{
  "configurationId": "0057c39d-f404-4de4-a527-1250a8935fed",
  "user": "test",
  "url": "http://dm-delta.metapack.com/api/5.x/services",
  "enabled": true,
  "repositoryType": "SMR",
  "retailerId": "0057c39d-f404-4de4-a527-1250a8935fed",
  "apiKey": "0057c39d-f404-4de4-a527-1250a8935fed",
  "password": "test"
}
```

Response

```
HTTP/1.1 200 OK

Configuration successfully updated
```

Failure

Unexistent configuration ID:

```
HTTP/1.1 404 NOT FOUND
```

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Location Services

Location Services

Introduction

This document provide practical examples on how to use the DDO Locations API to manage locations.

This is a RESTful API which forms a different paradigm to the RPC-style we can see, for example in SOAP APIs.

The main premise is that we're not invoking functions but interacting with resources and we use the HTTP protocol's operations / verbs and standard HTTP Status codes to signal success or failure.

HTTP Request Methods

The resource in our case is locations with which we interact by using the appropriate HTTP request methods:

- `POST` create a location or opening times rules

```
/locations
```

- `GET` retrieve the list of available locations or a specific one or the opening times rules for a specific location

```
/locations/<id>
/locations/<id>/openingTimesRules
```

- `DELETE` remove a location

```
/locations/<id>
```

- `PUT` update location or opening times rules (use case: change the store name or disable a location for a given day)

```
/locations/<id>
```

HTTP Status Codes

Another guideline for RESTful APIs which we follow is in using the appropriate HTTP Status Code to provide a status about the operation which the user has attempted. Additional information is frequently included as part of the response body but the overall status (created, server error, partial content, ...) can be obtained directly.

We are currently using the following codes listed below.

Location or opening times rules created:

```
HTTP/1.1 201 CREATED
```

Retrieved, deleted or updated a location or opening times rules, successfully:

```
HTTP/1.1 200 OK
```

Request error; e.g. invalid JSON object or invalid UUID string:

```
HTTP/1.1 400 BAD REQUEST
```

Entity with provided id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

HATEOAS

HATEOAS stands for: Hypermedia As The Engine Of Application State. We incorporate this concept in our API by providing links to resources in the response for some operations. As an example, when creating a new Location, you should get a JSON object in the response body which contains the link for the Location which was just created.

Example:

```
{
  "rel": "self",
  "href": "/locations/e7f936c"
}
```

We can then retrieve the opening times rules for this location by performing a GET request on /locations/e7f936c

Authentication and Authorization

All of the requests described in this document must contain a valid OAuth 2 bearer token (see Authentication and Authorization).

Creating a New Location

To create a new location we need to do a `POST` request to `/locations` with the `application/json` content-type request header and a JSON object containing the specified properties in the request body.

Properties

Name	Mandatory	Type	Length	
`storeId`	`true`	`string`	36	`"030HHE"`
`storeName`	`true`	`string`	128	`"Haywards Heath"`
`supplementaryName`	`false`	`string`	128	`"Provider Store #10"`
`structuredAddress`	`cond` (*1)	`object`		
`structuredAddress.street`	`true`	`string`	256	`Commercial Square`
`structuredAddress.buildingName`	`false`	`string`	64	`Palmerston House`
`structuredAddress.houseNumber`	`false`	`string`	64	`4`
`structuredAddress.city`	`true`	`string`	64	`London`
`structuredAddress.district`	`false`	`string`	64	`Greater London`
`structuredAddress.province`	`false`	`string`	64	`Hogwarts`
`structuredAddress.postCode`	`true`	`string`	16	`RH16 1DR`
`structuredAddress.countryCode`	`true`	`string`	16	`GBR`
`city`	`cond`	`string`	64	`"London"`
`postCode`	`cond`	`string`	16	`"RH16 1DR"`
`countryCode`	`cond`	`string` (*2)	3	`"GBR"`
`address`	`cond`	`string`	256	`"4 Palmerston House; Comm`
`latitude`	`true`	`signed floating point number`	11,8	`51.006035`
`longitude`	`true`	`signed floating point number`	11,8	`-0.102897`
`locationProvider.id`	`true`	`UUID` (*3)	-	`"2a2b72ca-980b-4c24-8e33-`
`telephoneNumber`	`false`	`string`	16	`"+44 7894 465 4561"`
`email`	`false`	`string`	127(*4)	`"test@metapack.com"`
`logoUrl`	`false`	`string`	128	`"http://logodomain.com/ni`
`photoUrls`	`false`	`string`	512	`"http://logodomain.com/ni`
`hasDisabledAccess`	`false`	`boolean`	-	`true`
`description`	`false`	`string`	512	`"Nice Store"`
`tags`	`false`	`string array`	-	`["a","b","c"]`
`translations`	`false`	`map<string,translation>` (*5)	-	-

Notes

1. Structured address (`structuredAddress.*`) fields are mutually exclusive with `address` , `city` , `postCode` , `countryCode`
2. Valid 3 character / 2 character / 3 digit country code
3. Existing Location Provider identifier
4. Must be a well-formed email address. Both local part and domain part of the email address are limited to 63 characters.
5. See Translations

Required Permission

`can-create-locations`

See Auth* (scopes).

Request

Structured Address Format

```
POST /locations
Content-Type: application/json
```

```
{
   "storeId": "001MKC",
   "storeName": "Milton Keynes",
   "supplementaryName": "Nice Store 1",
   "structuredAddress": {
     "street": "Elder Gate",
     "buildingName": "DODO Estate",
     "houseNumber": "202",
     "city": "Milton Keynes",
     "district": "Greater London",
     "province": "Hogwarts",
     "postCode": "MK9 1GW",
     "countryCode": "GBR"
   },
   "latitude":52.035538,
   "longitude":-0.774214,
   "telephoneNumber": "+441908237494",
   "email": "mkstore@askjeeves.com",
   "tags":[ "a", "b", "c" ],
   "locationProvider": { "id": "2a2b72ca-980b-4c24-8e33-a26b2820a3db" }
}
```

Legacy Address Format

```
POST /locations
Content-Type: application/json
```

```
{
   "storeId":"001MKC",
   "storeName":"Milton Keynes",
   "city":"Milton Keynes",
   "postCode":"MK9 1GW",
   "countryCode":"GBR",
   "address":"202 Elder Gate",
   "latitude":52.035538,
   "longitude":-0.774214,
   "telephoneNumber":"+441908237494",
   "email":"mkstore@askjeeves.com",
   "tags":[
       "a",
       "b",
       "c"
   ],
   "locationProvider":{
      "id":"2a2b72ca-980b-4c24-8e33-a26b2820a3db"
   }
}
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
   "rel":"self",
   "href":"/locations/c06c2305-1b33-43d2-8695-72a88a9cda61"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

```
{
   "error":"Invalid property: 'addr ess'"
}
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Reading Location Data

Required Permission

`can-retrieve-locations`

See Auth* (scopes).

Request

```
GET /locations/<id>
```

(As for all other GET requests, there is no need to specify the "Accepts" header as we only support JSON at the moment)

Response

If the `structuredAddress` fields are populated, the top level `address`, `city`, `postCode`, `countryCode` ones are as well, for backward compatibility reasons. In this scenario, `address` is a space separated concatenation of `structuredAddress.houseNumber`, `structuredAddress.street`, `structuredAddress.district` and `structuredAddress.province`.

```
HTTP/1.1 200 OK
```

```
{
  "storeId": "001MKC",
  "storeName": "Milton Keynes",
  "supplementaryName": "Nice Store 1",
  "city": "Milton Keynes",
  "postCode": "MK9 1GW",
  "countryCode": "GBR",
  "address": "202 Elder Gate Greater London Hogwarts",
  "structuredAddress": {
    "street": "Elder Gate",
    "buildingName": "DODO Estate",
    "houseNumber": "202",
    "city": "Milton Keynes",
    "district": "Greater London",
    "province": "Hogwarts",
    "postCode": "MK9 1GW",
    "countryCode": "GBR"
  },
  "latitude": 52.035538,
  "longitude": -0.774214,
  "telephoneNumber": "+441908237494",
  "email": "mkstore@askjeeves.com",
  "logoUrl": "http://logo.metapack.com/testlogo.gif",
  "photoUrls": [
    "http://logo.metapack.com/testlogo.gif",
    "http://logo.metapack.com/testlogo.gif"
  ],
  "hasDisabledAccess": true,
  "description": null,
  "id": "863086bf-0771-44aa-bb2a-14686a1211fa",
  "tags": ["a", "b", "c"],
  "locationProvider": {
    "id": "2a2b72ca-980b-4c24-8e33-a26b2820a3db",
    "name": "DoDo Test"
  },
  "links": [{
      "rel": "self",
      "href": "/locations/863086bf-0771-44aa-bb2a-14686a1211fa"
    },
    {
      "rel": "openingTimesRules",
      "href": "/locations/863086bf-0771-44aa-bb2a-14686a1211fa/openingTimesRules"
    },
    {
      "rel": "customData",
      "href": "/locations/863086bf-0771-44aa-bb2a-14686a1211fa/customData"
    }
  ]
}
```


The response contains a link to the entity itself and to its opening times rules.

Fields (apart from the ones already detailed):

- id - string - UUID for the location in DDO
- links (HATEOAS)
  - self
    - href - link for same resource
  - openingTimesRules
    - href - link for the location opening times rules
  - customData
    - href - link for the location custom data

Failure

Request error; e.g. invalid UUID string (id):

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Updating a Location

Required Permission

`can-update-locations`

See Auth* (scopes).

Properties

See Creating a New Location

Request

```
PUT /locations/<id>
Content-Type: application/json
```

Structured Address Format

```
{
   "storeId":"001MKC",
   "storeName":"Milton Keynes",
   "structuredAddress": {
     "street": "Elder Gate",
     "buildingName": "DODO Estate",
     "houseNumber": "202",
     "city": "Milton Keynes",
     "district": "Greater London",
     "province": "Hogwarts",
     "postCode": "MK9 1GW",
     "countryCode": "GBR"
   },
   "latitude":52.035538,
   "longitude":-0.774214,
   "telephoneNumber":"+441908237494",
   "email":"mkstore@askjeeves.com",
   "logoUrl":"http://logo.metapack.com/testlogo.gif",
   "photoUrls":[
      "http://logo.metapack.com/testlogo.gif",
      "http://logo.metapack.com/testlogo.gif"
   ],
   "hasDisabledAccess":true,
   "description":null,
   "tags":[
       "a",
       "b",
       "c"
   ],
   "locationProvider":{
      "id":"2a2b72ca-980b-4c24-8e33-a26b2820a3db"
   }
}
```

Legacy Address Format

```
{
   "storeId":"002MKC",
   "storeName":"Milton Keynes",
   "city":"Milton Keynes",
   "postCode":"MK9 1GW",
   "countryCode":"GBR",
   "address":"Elder Gate",
   "latitude":52.035538,
   "longitude":-0.774214,
   "telephoneNumber":"+441908237494",
   "email":"mkstore@askjeeves.com",
   "logoUrl":"http://logo.metapack.com/testlogo.gif",
   "photoUrls":[
      "http://logo.metapack.com/testlogo.gif",
      "http://logo.metapack.com/testlogo.gif"
   ],
   "hasDisabledAccess":true,
   "description":null,
   "tags":[
       "a",
       "b",
       "c"
   ],
   "locationProvider":{
      "id":"2a2b72ca-980b-4c24-8e33-a26b2820a3db"
   }
}
```


Response

```
HTTP/1.1 200 OK
```


```
{
   "rel":"self",
   "href":"/locations/c06c2305-1b33-43d2-8695-72a88a9cda61"
}
```


Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Deleting Locations

Required Permission

`can-delete-own-locations`

See Auth* (scopes).

Request

```
DELETE /locations/<id>
```

Response

```
HTTP/1.1 200 OK
```

(empty response body)

Note

Deleting a location also deletes it's sub-resources, namingly, the associated Opening Times Rules.

Failure

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Location Translations

Locations services has the capability of storing location(s) data in multiple translations and searching for location(s) data in a specific translations.

Location Services has the convention of default translations and explicitly provided translations.

Default Translations

Location data that should be considered as default translation must be provided as part of the actual location and it is not required to be explicitly provided with the translations.

This translation data will be returned on search queries whenever no translation exists in the requested language.

Storing Translations

Translations for a location can be provided (in the translations field) during create or update operation as part of location data.

Translations field:

Name	Mandatory	Type	Length	Example
`translations`	`false`	`map<string,translation>` (*1)	(*2)	-

Notes

1. Key for the map must be ISO 639-1 two-letter code in lowercase
2. Each location can have at maximum `10` translations

Translation

Location Data

Name	Mandatory	Type	Length	Example
`storeName`	`true`	`string`	128	`Haywards Heath`
`supplementaryName`	`false`	`string`	128	`Provider Store #10`
`description`	`false`	`string`	512	`Nice Store`

Location Address Data

Location address translation data can be provided either in structured address or unstructured address data format. Providing both will cause the request to be rejected.

Additionally location address translation data must have the same structure as the address data provided as part of the location. Any request violating this will be rejected.

Unstructured Address Data

Name	Mandatory	Type	Length	Example
`address`	`true`	`string`	256	`4 Palmerston House; Commercial Square`
`city`	`true`	`string`	64	`London`

Structured Address Data

Name	Mandatory	Type	Length	Example
`structuredAddress.street`	`true`	`string`	256	`Commercial Square`
`structuredAddress.buildingName`	`false`	`string`	64	`Palmerston House`
`structuredAddress.houseNumber`	`false`	`string`	64	`4`
`structuredAddress.city`	`true`	`string`	64	`London`
`structuredAddress.district`	`false`	`string`	64	`Greater London`
`structuredAddress.province`	`false`	`string`	64	`Hogwarts`

Example Request Body

This is an example location object which has data in French and Flemish. French is the default translation as it is provided in the location, but not in the translations.

```
{
  "storeId": "TST001",
  "storeName": "LIBRAIRIE ESCAUT MOLENBEEK",
  "structuredAddress": {
    "street": "BOULEVARD DU JUBILÉ",
    "houseNumber": "202",
    "city": "MOLENBEEK-SAINT-JEAN",
    "postCode": "1080",
    "countryCode": "BEL"
  },
  "translations": {
    "nl": {
      "storeName": "DAGBL L ESCAUT MOLENBEEK",
      "structuredAddress": {
        "street": "JUBELFEESTLAAN",
        "houseNumber": "202",
        "city": "SINT-JANS-MOLENBEEK"
      }
    }
  },
  "latitude": 50.852452,
  "longitude": 4.330518,
  "telephoneNumber": "+321908237494",
  "email": "mkstore@askjeeves.com",
  "tags": ["a", "b", "c" ],
  "locationProvider": {
    "id": "f65e6dea-1d3c-4734-8306-095da0955f48"
  }
}
```

Reading Translations

Location services API's are supporting either retrival of location(s) with all translations data or location data in a spefic translation.

Search/Retrieve Location Data With All Translations

The following table details the calls that return all translations:

Endpoint	Parameters	Description	Reference
/locations/{id}	-	Retrieve location	Reading Location Data
/locations	location_provider_id&countrycode	Retrieve locations for providers	Generic search
/locations	consolidator_id&countrycode	Retrieve locations for consolidator	Generic search
/locationProviders/{providerId}/locations	-	Retrieve locations for providers	-

Search/Retrieve Location Data In A Specific Translation

In order to retrieve/search location(s) in a specific translation the parameter `language=<ISO 639-1 two-letter code>` must be provided in the request.

The translation in which the location(s) data is returned depends on whether `language` parameter is passed and also on whether a translation in the requested language exists. The following table summarises possible scenarios.

language parameter provided	Translation in specified language exists	Translation
No	N/A	DEFAULT
Yes	No	DEFAULT
Yes	Yes	In provided language

The following table details the calls that return location(s) in requested language:

Endpoint	Description	Reference
/locations	Generic location search	Generic search

Opening Times Rules

Adding Opening Times Rules to a Location

The openingTimesRules endpoint allows for the configuration of opening times, per weekday for a given store.

These rules are optional. If none is specified, the assumption is that the location or store is never available.

Properties

- `openingTimesRules[i].rule` - string - the currently supported rule format is " `every <weekday>` " (weekday in monday, tuesday ... sunday)
- `openingTimesRules[i].openingClosingTimes` - array of time periods for the given weekday for which the location is open
  - `openingTimesRules[i].openingClosingTimes[j].openingAt` : string - opening time in the following format: " `<2 digit hours>:<2 digit minutes>` "
  - `openingTimesRules[i].openingClosingTimes[j].closingAt` : string - closing time in the following format: " `<2 digit hours>:<2 digit minutes>` "

Notes

1. Opening / closing times values containing spaces or nulls or not conforming to the `<2 digit>:<2 digit>` format are not supported.

Required Permission

`can-create-locations`

See Auth* (scopes).

Request

```
POST /locations/<id>/openingTimesRules
Content-Type: application/json
```

```
{
   "openingTimesRules":[
      {
         "rule":"every Monday",
         "openingClosingTimes":[
            {
               "openingAt":"08:00",
               "closingAt":"18:00"
            }
         ]
      },
      {
         "rule":"every Tuesday",
         "openingClosingTimes":[
            {
               "openingAt":"08:00",
               "closingAt":"18:00"
            }
         ]
      }
   ]
}
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
   "rel":"self",
   "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14/openingTimesRules"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Reading Opening Times Rules

Required Permission

`can-retrieve-locations`

See Auth* (scopes).

Request

```
GET /locations/<id>/openingTimesRules
```

Response

```
HTTP/1.1 200 OK
```


```
{
   "openingTimesRules":[
      {
         "rule":"every Monday",
         "openingClosingTimes":[
            {
               "openingAt":"09:00",
               "closingAt":"18:00"
            }
         ]
      },
      {
         "rule":"every Tuesday",
         "openingClosingTimes":[
            {
               "openingAt":"09:00",
               "closingAt":"18:00"
            }
         ]
      }
   ],
   "links":[
      {
         "rel":"self",
         "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14/openingTimesRules"
      },
      {
         "rel":"location",
         "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14"
      }
   ]
}
```


Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Updating Opening Times Rules

The original opening times rules will be swapped with the newly provided data. An example use case for this would be to disable a location (identified by the ID field on the URL) on a certain weekday - from the previous example, where we had the store open on Mondays and Tuesdays, this example disables it on Tuesdays.

Required Permission

`can-update-locations`

See Auth* (scopes).

Request

```
PUT /locations/<id>/openingTimesRules
Content-Type: application/json
```

```
{
   "openingTimesRules":[
      {
         "rule":"every Monday",
         "openingClosingTimes":[
            {
               "openingAt":"08:00",
               "closingAt":"18:00"
            }
         ]
      }
   ]
}
```

Response

```
HTTP/1.1 200 OK
```

```
{
   "rel":"self",
   "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14/openingTimesRules"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Deleting Opening Times Rules

Required Permission

`can-update-locations`

See Auth* (scopes).

Request

```
DELETE /locations/<id>/openingTimesRules
```

Response

```
HTTP/1.1 200 OK
```

Empty response body.

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Custom Data

For use cases that are not in the scope of the locations services and can not be generalised, location services allow to store `custom data` on location by location basis.

`custom data` must be in `json` and must comply with RFC7159.

Adding Custom Data to a Location

In order to add `custom data` a location must already exist.

Required Permission

Any of `can-create-locations` , `can-update-locations`

See Auth* (scopes).

Request

```
POST /locations/<id>/customData
Content-Type: application/json
```

```
{
   "type": "Shop"
}
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
   "rel":"self",
   "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14/customData"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 (internal server error)
```

Updating Custom Data for a Location

Required Permission

Any of `can-create-locations`, `can-update-locations`

See Auth* (scopes).

Request

```
PUT /locations/<id>/customData
Content-Type: application/json
```

```
{
   "type": "Shop"
}
```

Response

```
HTTP/1.1 200 OK
```

```
{
   "rel":"self",
   "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14/customData"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 (internal server error)
```

Retrieving Custom Data for a Location

Costum data can be retrieved independent of location or might be returned as part of location data.

Required Permission

`can-retrieve-locations`

See Auth* (scopes).

Standalone custom data

Request

```
GET /locations/<id>/customData
```

Response when custom data is `json` object

Given custom data is following `json` object

```
{
    "type": "shop",
    "capacity" : 5000
}
```

then response is going to be

```
HTTP/1.1 200 OK
```

```
{
   "type": "shop",
   "capacity": 5000,
   "links":[
      {
         "rel":"self",
         "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14/customData"
      },
      {
         "rel":"location",
         "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14"
      }
   ]
}
```

Response when custom data is not `json` object

Given custom data is a quoted string

```
"shop"
```

then response is going to be

```
HTTP/1.1 200 OK
```

```
{
   "value": "shop",
   "links":[
      {
         "rel":"self",
         "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14/customData"
      },
      {
         "rel":"location",
         "href":"/locations/6b2e279d-8afd-4785-8ea4-b06739fb0f14"
      }
   ]
}
```

Custom data as part of location data

Some of the location searchs will include custom data with the the other location data. Those searches mainly accept a `retailer id` as request parameter. If present then custom data will be attached to the field `customData`.

Given location search finds a location with the following custom data

```
{
    "type": "shop",
    "capacity" : 5000
}
```

then the response is going to look like following(lots of location properties removed for brevity)

```
[{
    "storeId":"002MKC",
    "storeName":"Milton Keynes",
    "customData": {
        "type": "shop",
        "capacity" : 5000
    }
}]
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 (internal server error)
```

Deleting Custom Data from a Location

`can-update-locations`

See Auth* (scopes).

Request

```
DELETE /locations/<id>/customData
```

Response

```
HTTP/1.1 200 OK
```

Empty response body.

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 (internal server error)
```

Internal API

This section lists APIs exposed to internal MetaPack services. If you use `retailer` or `location provider` credentials, you shoud get a `HTTP 401 (unauthorized)` or `HTTP 403 (forbidden)` status code.

Locations Search

The generic search method `/locations?...` is an internal low level search API which supports searching for PUDOs and own stores simultaneously, given a considerably large set of incremental and mutually exclusive parameters. It should be deprecated in favour of the simpler and more efficient `/locations/pudos?...` and `/locations/ownstores?...` endpoints.

PUDOs

Required Permission

`can-search-locations`

See Auth* (scopes).

Request

`GET /locations/pudos?<query string>`

Parameters

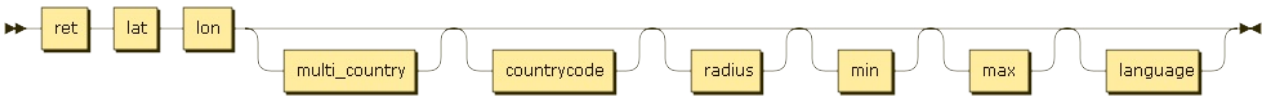
Parameter	Example	Default	Type	Description
`correlation_id`	`9180b450-1e4c-4dd1-b0e5-37d0bfaaf185`	-	string	identifier for request tracking
`ret`	`6f3c2d96-fccb-46f5-9dd7-7e0a478dfdfd`	-	uuid	retailer id (*1)
`lon`	`-0.1274`	-	signed floating point	longitude
`lat`	`51.5197`	-	signed floating point	latitude
`multi_country`	`true`	`false`	boolean	should include locations from multiple countries (*2)
`countrycode`	`GBR`	-	ISO 3166-1 country code (*3)	country code
`radius`	`10000`	`1000`	integer	radius
`min`	`10000`	`-`	integer	minimum number of locations to retrieve (*4)
`max`	`10000`	`100`	integer	maximum number of locations to retrieve
`language`	`en`	-	`string`	prefer location data in this language (*5)

Notes

1. Manually generated and provided by Metapack for each retailer.
2. Different to the one specified via the `countrycode` parameter.
3. Valid 3 digit, 2 characters or 3 characters.
4. Will search beyond radius if initial search doesn't find enough locations.
5. Must be a valid lowercase ISO 639-1 two-letter code (see)

Syntax Diagram

The following diagram lists all of the valid combinations of parameters. Read from left to right, choosing a single path; loops represent optional parameters; forks represent alternatives; a parameter may appear more than once in which case, ignore the subsequent ocurrences if already added:




```
locations.pudos
        ::= ret lat lon multi_country? countrycode? radius? min? max? language?
```

Response

```
HTTP/1.1 200 OK
```


```
[
  {
    "storeId": "0002",
    "storeName": "MetaPack Limited",
    "city": "London",
    "postCode": "WC1X 8XZ",
    "countryCode": "GBR",
    "address": "201 Gray’s Inn Road",
    "latitude": 51.523402,
    "longitude": -0.114333,
    "telephoneNumber": "+442078436720",
    "email": "mkstore@askjeeves.com",
    "logoUrl": null,
    "photoUrls": [],
    "hasDisabledAccess": false,
    "description": null,
    "id": "ec30cb2b-9c64-46ee-a6b4-2db3238901d9",
    "tags": [
       "potato",
       "pharmacy",
       "refrigeration"
    ],
    "locationProvider": {
      "id": "ca0e1f97-dc57-4df5-9430-c93264603197",
      "name": "DoDo Test",
      "ownStores": false
    },
    "locationOpeningTimesRules": {
      "openingTimesRules": [
        {
          "rule": "every Sunday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Monday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Tuesday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
```

```
        },
        {
          "rule": "every Wednesday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Thursday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Friday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Saturday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        }
      ]
    },
    "links": [
      {
        "rel": "self",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9"
      },
      {
        "rel": "openingTimesRules",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/openingTimesRules"
      },
     {
        "rel": "customData",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/customData"
       }
    ]
  }
]
```


Failure

Invalid parameter combination or data types submitted:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer or location provider does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Own Stores

Required Permission

`can-search-locations`

See Auth* (scopes).

Request

```
GET /locations/ownstores?<query string>
```

Parameters

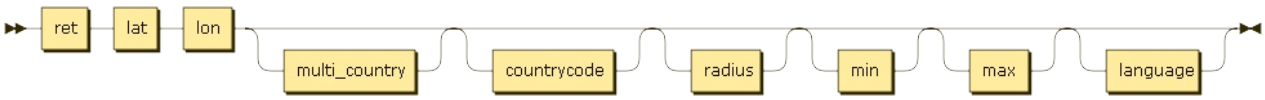
Parameter	Example	Default	Type	Description
`correlation_id`	`9180b450-1e4c-4dd1-b0e5-37d0bfaaf185`	-	string	identifier for request tracking
`ret`	`6f3c2d96-fccb-46f5-9dd7-7e0a478dfdfd`	-	uuid	retailer id (*1)
`lon`	`-0.1274`	-	signed floating point	longitude
`lat`	`51.5197`	-	signed floating point	latitude
`multi_country`	`true`	`false`	boolean	should include locations from multiple countries (*2)
`countrycode`	`GBR`	-	ISO 3166-1 country code (*3)	country code
`radius`	`10000`	`1000`	integer	radius
`min`	`10000`	`-`	integer	minimum number of locations to retrieve (*4)
`max`	`10000`	`100`	integer	maximum number of locations to retrieve
`language`	`en`	-	`string`	prefer location data in this language (*5)

Notes

1. Manually generated and provided by Metapack for each retailer.
2. Different to the one specified via the `countrycode` parameter.
3. Valid 3 digit, 2 characters or 3 characters.
4. Will search beyond radius if initial search doesn't find enough locations.
5. Must be a valid lowercase ISO 639-1 two-letter code (see)

Syntax Diagram

The following diagram lists all of the valid combinations of parameters. Read from left to right, choosing a single path; loops represent optional parameters; forks represent alternatives; a parameter may appear more than once in which case, ignore the subsequent ocurrences if already added:




```
locations.ownstores
        ::= ret lat lon multi_country? countrycode? radius? min? max? language?
```

Response

```
HTTP/1.1 200 OK
```


```
[
  {
    "storeId": "0002",
    "storeName": "MetaPack Limited",
    "city": "London",
    "postCode": "WC1X 8XZ",
    "countryCode": "GBR",
    "address": "201 Gray’s Inn Road",
    "latitude": 51.523402,
    "longitude": -0.114333,
    "telephoneNumber": "+442078436720",
    "email": "mkstore@askjeeves.com",
    "logoUrl": null,
    "photoUrls": [],
    "hasDisabledAccess": false,
    "description": null,
    "id": "ec30cb2b-9c64-46ee-a6b4-2db3238901d9",
    "tags": [
       "potato",
       "pharmacy",
       "refrigeration"
    ],
    "locationProvider": {
      "id": "ca0e1f97-dc57-4df5-9430-c93264603197",
      "name": "DoDo Test",
      "ownStores": false
    },
    "locationOpeningTimesRules": {
      "openingTimesRules": [
        {
          "rule": "every Sunday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Monday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Tuesday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
```

```
        },
        {
          "rule": "every Wednesday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Thursday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Friday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Saturday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        }
      ]
    },
    "links": [
      {
        "rel": "self",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9"
      },
      {
        "rel": "openingTimesRules",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/openingTimesRules"
      },
      {
       "rel": "customData",
       "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/customData"
      }
    ]
  }
]
```


Failure

Invalid parameter combination or data types submitted:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer or location provider does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Generic Search

Required Permission

`can-search-locations`

See Auth* (scopes).

Request

`GET /locations?<query string>`

Parameters

Parameter	Example	Default	Type	Description
`correlation_id`	`9180b450-1e4c-4dd1-b0e5-37d0bfaaf185`	-	string	identifier for request tracking
`ret`	`6f3c2d96-fccb-46f5-9dd7-7e0a478dfdfd`	-	uuid	retailer id (*1)
`location_provider_id`	`e8ec5293-449b-45e5-9435-cd90ca942e3a`	-	uuid	location provider id
`consolidator_id`	`40251966-8252-41aa-a888-1f1953fe27d5`	-	uuid	consolidator id
`store_id`	`1234`	-	string	store (external) id
`lon`	`-0.1274`	-	signed floating point	longitude
`lat`	`51.5197`	-	signed floating point	latitude
`postcode`	`N19 1AA`	-	string	postcode
`countrycode`	`GBR`	-	ISO 3166-1 country code (*2)	country code
`radius`	`10000`	`1000`	integer	radius
`include_own_stores`	`true`	`true`	boolean	include own stores?
`own_stores`	`false`	`false`	boolean	own stores only
`own_stores_qty`	`1`	`3`	integer	number of closest own stores to retrieve (beyond radius)
`pudo_qty`	`1`	`0`	integer	number of pudos + own stores to retrieve (beyond radius)
`multi_country`	`true`	`false`	boolean	should include locations from multiple countries (*3)
`language`	`en`	-	`string`	prefer location data in this language (*4)

Notes

1. Manually generated and provided by Metapack for each retailer.
2. Valid 3 digit, 2 characters or 3 characters.

3. Different to the one specified via the `countrycode` parameter.
4. Must be a valid lowercase ISO 639-1 two-letter code (see)

Syntax Diagram

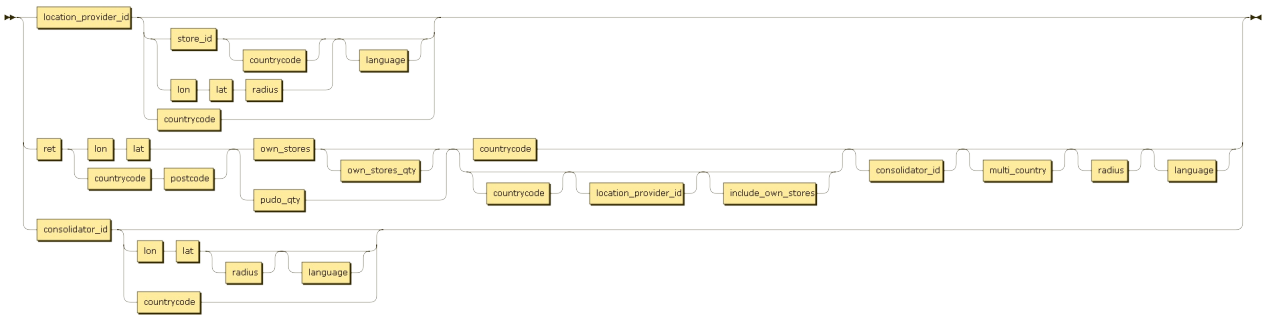
Some of the parameters are mutually exclusive (e.g. lat + lon / postcode + countrycode).

1. Search by location provider
   i. Retrieve all locations optionally filtered by country (*1)
   ii. Retrieve a list of locations given the location provider internal id (store id) and optionally, a unique one, given the country code (*2)
   iii. Radius search for active locations, given the radius value and geo-coordinates for the center (*3)
2. Search active locations
   i. Search own stores above radius (*4) (*5)
   ii. Search all locations configured for the retailer (*6) (*7)
      i. Filter by location provider, according to configuration on Location Services (*8) (*9)
      ii. Filter by consolidator, according to configuration on Location Services (*10) (*11)
3. Search by consolidator
   i. Retrieve all locations optionally filtered by country (*12)
   ii. Radius search for active locations, given the radius value and geo-coordinates for the center (*13)

Examples

1. `GET /locations?location_provider_id=<...>[&countrycode=<...>]`
2. `GET /locations?location_provider_id=<...>&store_id=<...>[&countrycode=<...>]`
3. `GET /locations?location_provider_id=<...>&lat=<latitude>&lon=<longitude>[&radius=<radius in meters>]`
4. `GET /locations?ret=<retailer id>&lat=<latitude>&lon=<longitude>&own_stores=true[&own_stores_qty=<number of stores to retrieve>][&multi_country=<...>][&radius=<minimum distance from center>]`
5. `GET /locations?ret=<...>&postcode=<...>&countrycode=<...>&own_stores=true[&own_stores_qty=<number of stores to retrieve>][&multi_country=<...>][&radius=<minimum distance from center>]`
6. `GET /locations?ret=<...>&lat=<latitude>&lon=<longitude>[&countrycode=<...>][&include_own_stores=<...>][&multi_country=<...>][&radius=<...>]`
7. `GET /locations?ret=<...>&postcode=<...>&countrycode=<...>[&include_own_stores=<...>][&multi_country=<...>][&radius=<...>]`
8. `GET /locations?ret=<...>&lat=<latitude>&lon=<longitude>&countrycode=<...>&location_provider_id=<...>[&include_own_stores=<...>][&multi_country=<...>][&radius=<...>]`
9. `GET /locations?ret=<...>&postcode=<...>&countrycode=<...>&location_provider_id=<...>[&include_own_stores=<...>][&multi_country=<...>][&radius=<...>]`
10. `GET /locations?ret=<...>&lat=<latitude>&lon=<longitude>&countrycode=<...>&consolidator_id=<...>[&include_own_stores=<...>][&multi_country=<...>][&radius=<...>]`
11. `GET /locations?ret=<...>&postcode=<...>&countrycode=<...>&consolidator_id=<...>[&include_own_stores=<...>][&multi_country=<...>][&radius=<...>]`
12. `GET /locations?consolidator_id=<...>[&countrycode=<...>]`
13. `GET /locations?consolidator_id=<...>&lat=<latitude>&lon=<longitude>[&radius=<radius in meters>]`

The following diagram lists all of the valid combinations of parameters. Read from left to right, choosing a single path; loops represent optional parameters; forks represent alternatives; a parameter may appear more than once in which case, ignore the subsequent ocurrences if already added:




```
locations
         ::= location_provider_id ( store_id countrycode? language?| lon lat radius  language?| countrycode? )
           | ret ( lon lat | countrycode postcode ) ( own_stores own_stores_qty? | pudo_qty ) ( countrycode | countrycode? loc
ation_provider_id? include_own_stores? ) consolidator_id? multi_country? radius? language?
           | consolidator_id ( lon lat radius? language?| countrycode? )
```

Response

```
HTTP/1.1 200 OK
```


```
[
  {
    "storeId": "0002",
    "storeName": "MetaPack Limited",
    "city": "London",
    "postCode": "WC1X 8XZ",
    "countryCode": "GBR",
    "address": "201 Gray’s Inn Road",
    "latitude": 51.523402,
    "longitude": -0.114333,
    "telephoneNumber": "+442078436720",
    "email": "mkstore@askjeeves.com",
    "logoUrl": null,
    "photoUrls": [],
    "hasDisabledAccess": false,
    "description": null,
    "id": "ec30cb2b-9c64-46ee-a6b4-2db3238901d9",
    "tags": [
       "potato",
       "pharmacy",
       "refrigeration"
    ],
    "locationProvider": {
      "id": "ca0e1f97-dc57-4df5-9430-c93264603197",
      "name": "DoDo Test",
      "ownStores": false
    },
    "links": [
      {
        "rel": "self",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9"
      },
      {
        "rel": "openingTimesRules",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/openingTimesRules"
      },
     {
        "rel": "customData",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/customData"
       }
    ]
  }
]
```

Failure

Invalid parameter combination or data types submitted:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer or location provider does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Search Active Locations API

See Search Active Locations API

Find locations by External Id

Required Permission

`can-search-locations`

See Auth* (scopes).

Request

```
GET /locations/findByExternalId?<query string>
```

Parameters

Parameter	Example	Required	Type	Description
`countrycode`	`GBR`	Yes	ISO 3166-1 country code (*1)	country code
`store_id`	`1234`	Yes	`string`	Store (external id)
`postcode`	`WC1X 8XZ`	No	`string`	postcode
`language`	`en`	No	`string`	prefer location data in this language (*2)

Notes

1. Valid 3 digit, 2 characters or 3 characters.
2. Must be a valid lowercase ISO 639-1 two-letter code (see)

Response

```
HTTP/1.1 200 OK
```


```
[
  {
    "storeId": "0002",
    "storeName": "MetaPack Limited",
    "city": "London",
    "postCode": "WC1X 8XZ",
    "countryCode": "GBR",
    "address": "201 Gray's Inn Road",
    "latitude": 51.523402,
    "longitude": -0.114333,
    "telephoneNumber": "+442078436720",
    "email": "mkstore@askjeeves.com",
    "logoUrl": null,
    "photoUrls": [],
    "hasDisabledAccess": false,
    "description": null,
    "id": "ec30cb2b-9c64-46ee-a6b4-2db3238901d9",
    "tags": [
       "potato",
       "pharmacy",
       "refrigeration"
    ],
    "locationProvider": {
```

```
      "id": "ca0e1f97-dc57-4df5-9430-c93264603197",
      "name": "DoDo Test",
      "ownStores": false
    },
    "locationOpeningTimesRules": {
      "openingTimesRules": [
        {
          "rule": "every Sunday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Monday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Tuesday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Wednesday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Thursday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Friday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        },
        {
          "rule": "every Saturday",
          "openingClosingTimes": [
            {
              "openingAt": "00:00",
              "closingAt": "23:59"
            }
          ]
        }
      ]
    },
    "links": [
```

```
      {
        "rel": "self",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9"
      },
      {
        "rel": "openingTimesRules",
        "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/openingTimesRules"
      },
      {
       "rel": "customData",
       "href": "/locations/ec30cb2b-9c64-46ee-a6b4-2db3238901d9/customData"
      }
    ]
  }
]
```


Failure

Invalid parameter combination or data types submitted:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Geocoding

The MetaPack Options geocoding solution has a set of internal APIs that enable geocoding and reverse geocoding postcodes.

Required Permission

`can-geocode`

See Auth* (scopes).

Retrieving Coordinates for an Address or a Postcode

Requests

In order to get coordinates a `HTTP GET` request with one of the following parameter-sets must be made to the geocoding service.

Request path

```
GET /locations/coordinates?<parameters>
```

Request parameters for postcode + country:

Name	Required	Notes
postcode	true	-
countrycode	true	ISO 3166 alpha-2/3 compliant country code

Request parameters for line-based address

Name	Required	Notes
line1	true	-
line2	false	-
line3	false	-
line4	false	-
postcode	true	-
countrycode	true	ISO 3166 alpha-2/3 compliant country code

Request parameters for a structured address

Name	Required	Notes
company_name	false	-
door_code	false	-
house_nr	true	-
house_ext	false	-
floor	false	-
building	false	-
street	true	-
city	false	-
area	false	-
district	false	-
province	false	-
postcode	true	-
countrycode	true	ISO 3166 alpha-2/3 compliant country code

Response

```
HTTP/1.1 200 OK
```

```
{
    "latitude": 51.5663394349283,
    "longitude": -0.129791390131075
}
```

Failure

Invalid parameter combination or data types submitted:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Not found:

```
HTTP/1.1 404 NOT FOUND
```

When service is unavailable. Retrying the request may work:

```
HTTP/1.1 503 SERVICE UNAVAILABLE
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Retrieving the Closest Postcode to Geo-Coordinates

Request

Find postcode nearest to a given latitude and longitude:

```
GET /locations/postcode?&lat=<latitude>&long=<longitude>
```

Response

```
HTTP/1.1 200 OK
```

```
WC1B3QQ
```

Failure

Invalid parameter combination or data types submitted:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Not found:

```
HTTP/1.1 404 NOT FOUND
```

When service is unavailable. Retrying the request may work:

```
HTTP/1.1 503 SERVICE UNAVAILABLE
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

~~Retrieving a List of Closest Postcodes for Geo-Coordinates~~ (deprecated)

Request

Find postcodes within radius of geographic coordinates:

```
GET /locations/postcodes?&lat=<latitude>&long=<longitude>&radius=<radius in meters>
```

Response

```
HTTP/1.1 200 OK
```

```
[
    "WC1E7JW",
    "WC1B5BP"
]
```

Failure

Invalid parameter combination or data types submitted:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Not found:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Managing Retailers

Retailers can be linked to location providers and consolidators in Location Services.

For retailer configuration see corresponding documents.

Creating Retailers

Required Permission

`can-create-retailers`

See Auth* (scopes).

Request

```
POST /retailers HTTP/1.1
Content-Type: application/json
Authorization: Bearer <bearer token>
```

```
{
  "id": "00357362-3219-476e-87cd-4c05b1d1ce02",
  "name": "test",
  "details": "test 2"
}
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
  "rel": "self",
  "href":"/retailers/00357362-3219-476e-87cd-4c05b1d1ce02"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Retrieving Retailers

Required Permission

`can-read-retailers`

See Auth* (scopes).

Request

```
GET /retailers/:id HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `id` : retailer ID

Response

```
HTTP/1.1 200 OK
```

```
{
  "id": "556bd03a-0c4f-11e6-a148-3e1d05defe68",
  "name": "Test",
  "details": null
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Updating Retailers

Required Permission

`can-update-retailers`

See Auth* (scopes).

Request

```
PUT /retailers/:id HTTP/1.1
Content-Type: application/json
Authorization: Bearer <bearer token>
```

```
{
  "name": "test",
  "details": "test 2"
}
```

Path Parameters

- `id` : retailer ID

Response

```
HTTP/1.1 200 OK
```

```
{
  "rel": "self",
  "href":"/retailers/00357362-3219-476e-87cd-4c05b1d1ce02"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Deleting Retailers

Required Permission

`can-delete-retailers`

See Auth* (scopes).

Request

```
DELETE /retailers/:id HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `id` : retailer ID

Response

```
HTTP/1.1 200 OK
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer id doesn't exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Listing Retailers

Required Permission

`can-list-retailers`

See Auth* (scopes).

Request

```
GET /retailers HTTP/1.1
Authorization: Bearer <bearer token>
```

Response

```
HTTP/1.1 200 OK
```

```
[
  {
    "id": "b2803ba6-5417-41f1-be1a-b5656a6eb74c",
    "name": "developer",
    "details": null,
    "links": []
  }, {
    "id": "8bb1558f-a7aa-4372-a135-49950fe2f471",
    "name": "default",
    "details": null,
    "links": []
  }
  /** ... **/
]
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location id doesn't match an existing location:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Updating Location Providers Linked with Retailers

Required Permission

`can-update-retailers`

See Auth* (scopes).

Request

```
PUT /retailers/:id/locationProviders HTTP/1.1
Authorization: Bearer <bearer token>
```

```
[{
  "retailerId": "557bd13e-0c4f-11e6-a148-3e1d05defe78",
  "locationProviderId": "54438f2f-349b-4835-b3cb-871766344b45",
  "ownStores": false,
  "excludedLocationTags": ["tag1"]
}, {
  "retailerId": "557bd13e-0c4f-11e6-a148-3e1d05defe78",
  "locationProviderId": "5b1f0987-6636-4bfb-b4e0-0c8351f84790",
  "ownStores": false,
  "countryCode": "GBR"
}]
```

Response

```
HTTP/1.1 200 OK
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Listing Location Providers Linked with Retailers

Required Permission

`can-read-retailers`

See Auth* (scopes).

Request

```
GET /retailers/:id/locationProviders HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `id` : retailer ID

Response

```
HTTP/1.1 200 OK
```

```
[
  {
    "retailerId": "b2893ba5-5417-41f1-be1a-b5656a6eb74c",
    "locationProviderId": "5b1f1987-6636-4bfb-b4e0-0c8351f84780",
    "ownStores": true,
    "excludedLocationTags": [],
    "countryCode": "GBR",
    "links": []
  }, {
    "retailerId": "b2893ba5-5417-41f1-be1a-b5656a6eb74c",
    "locationProviderId": "65b83786-8b63-11e4-aa3e-db85c11f1a50",
    "ownStores": false,
    "excludedLocationTags": [],
    "countryCode": "PRT",
    "links": []
  }
  /* ... */
]
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Linking Consolidators with Retailers

Required Permission

`can-update-retailers`

See Auth* (scopes).

Request

```
POST /retailers/:retailerId/consolidators/:consolidatorId  HTTP/1.1
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
  "rel": "self",
  "href":"/retailers/00357362-3219-476e-87cd-4c05b1d1ce02/consolidators"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Consolidator id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Listing Consolidators Linked with Retailers

Required Permission

`can-read-retailers`

See Auth* (scopes).

Request

```
GET /retailers/:id/consolidators HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `id` : retailer ID

Response

```
HTTP/1.1 200 OK
```

```
[
  {
    "id": "c2798baa-0c4e-11e6-a248-3e1d05defe79",
    "name": "Test consolidator",
    "links": [
      {
        "rel": "self",
        "href": "/consolidators/c2798baa-0c4e-11e6-a248-3e1d05defe79"
      }
    ]
  }
]
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Unlinking Consolidators from Retailers

Required Permission

`can-update-retailers`

See Auth* (scopes).

Request

```
DELETE /retailers/:retailerId/consolidators/:consolidatorId HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "rel": "self",
  "href":"/retailers/00357362-3219-476e-87cd-4c05b1d1ce02/consolidators"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Retailer ID does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Location Providers

Delete Locations

Required Permission

`can-delete-all-locations`

See Auth* (scopes).

Request

Delete all locations for a specific provider:

```
DELETE /locationProviders/<location provider id>/locations
```

Response

If one or more locations are deleted:

```
HTTP/1.1 200 OK
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location provider does not exist or doesn't have locations:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Listing Location Providers

To list location providers we need to do a `GET` request to `/locationProviders`.

Required Permission

`can-list-location-providers`

See Auth* (scopes).

Request

List all location providers:

```
GET /locationProviders
```

Response

If no location provider found:

```
HTTP/1.1 200 OK
```

```
[]
```

If one or more location providers found:

```
HTTP/1.1 200 OK
```

```
[
  {
    "id": "68144875-24e5-4a31-ae74-f0924d2c513e",
    "name": "Test",
    "details": "Test location provider"
  },
  {
    "id": "68144875-24e5-4a31-ae74-f0924d2c513e",
    "name": "Test",
    "details": "Test location provider",
    "locationsSource": {
      "endpoint": "http://remote-locations"
    }
  }
]
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Retrieving Location Providers Data

To retrive a location provider with its id we need to do a `GET` request to `/locationProviders/<Location Provider ID>`

Required Permission

`can-retrieve-location-providers`

See Auth* (scopes).

Request

Retrive location provider by id:

```
GET /locationProviders/68144875-24e5-4a31-ae74-f0924d2c513e
```

Response

If location provider found:

```
HTTP/1.1 200 OK
```

```
{
  "id": "68144875-24e5-4a31-ae74-f0924d2c513e",
  "name": "Test",
  "details": "{\"tag\":\"test\"}",
  "locationsSource": {
    "endpoint": "http://remote-locations"
  },
  "archivingPeriod": 7
}
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location provider does not exist or doesn't have locations:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Creating Location Providers

To create a new location provider we need to do a `POST` request to `/locationProviders` with the `application/json` content-type request header and a JSON object containing the specified properties in the request body.

Properties

Name	Mandatory	Type	Length	Example
id	`false`	`string` (*1)	36	`"bd4be9b3-913f-4481-b363-88d490b95099"`
name	`true`	`string`	128	`"Test"`
details	`false`	`string` (*2)	-	`"{\"email\": \"test@abc.com\"}"`
locationsSource.endpoint	`false`	`string` (*3)	256	`"http://remote.locations"`
archivingPeriod	`true`	`int` (*4)	-	`7`

Notes

1. If not provided locations service will automatically assign a new id.
2. Must be well formed json, as specified in RFC 4627
3. If specified then indicates that this is a remote location provider. Location services will use provided url to fetch locations.
4. Archiving period must be in the range 1-30

Required Permission

`can-create-location-providers`

See Auth* (scopes).

Request

Create location provider:

```
POST /locationProviders
Content-Type: application/json
```

```
{
  "id": "68144875-24e5-4a31-ae74-f0924d2c513e",
  "name": "Test",
  "details": "{\"tag\":\"test\"}",
  "locationsSource": {
    "endpoint": "http://remote-locations"
  },
  "archivingPeriod": 7
}
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
   "rel":"self",
   "href":"/locations/68144875-24e5-4a31-ae74-f0924d2c513e"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

```
{
   "error":"Invalid property: 'deetails'"
}
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Updating Location Providers Data

To update a location provider we need to do a `PUT` request to `/locationProviders/<Location Provider ID>` with the `application/json` content-type request header and a JSON object containing the specified properties in the request body.

Properties

Name	Mandatory	Type	Length	Example
name	`true`	`string`	128	`"Test"`
details	`false`	`string` (*1)	-	`"{\"email\":\"test@abc.com\"}"`
locationsSource.endpoint	`false`	`string` (*2)	256	`"http://remote.locations"`
archivingPeriod	`true`	`int` (*3)	-	`7`

Notes

1. Must be well formed json, as specified in RFC 4627
2. If specified then indicates that this is a remote location provider. Location services will use provided url to fetch locations.
3. Archiving period must be in the range 1-30

Required Permission

`can-update-location-providers`

See Auth* (scopes).

Request

Update location provider:

```
PUT /locationProviders/68144875-24e5-4a31-ae74-f0924d2c513e
Content-Type: application/json
```

```
{
  "name": "Test",
  "details": "{\"tag\":\"test\"}",
  "locationsSource": {
    "endpoint": "http://remote-locations"
  },
  "archivingPeriod": 7
}
```

Response

Success

```
HTTP/1.1 200 OK
```

```
{
  "rel":"self",
  "href":"/locations/68144875-24e5-4a31-ae74-f0924d2c513e"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

```
{
"error":"Invalid property: 'deetails'"
}
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location provider does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Deleting Location Providers

Required Permission

`can-delete-location-providers`

See Auth* (scopes).

Request

Delete location provider:

```
DELETE /locationProviders/<id>
```

Response

```
HTTP/1.1 200 OK
```

Failure

If location provider has locations or linked with a retailer then this operation will fail with http status unprocessable entity.

```
HTTP/1.1 422 UNPROCESSABLE ENTITY
```

```
{
   "errorMessage":"<Reason for the failure>"
}
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Location provider does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Consolidators

Creating Consolidators

Required Permission

`can-create-consolidators`

See Auth* (scopes).

Request

```
POST /consolidators HTTP/1.1
Content-Type: application/json
Authorization: Bearer <bearer token>
```

```
{
  "id": "00357362-3219-476e-87cd-4c05b1d1ce02",
  "name": "test"
}
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
  "rel": "self",
  "href":"/consolidators/00357362-3219-476e-87cd-4c05b1d1ce02"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Retrieving Consolidators

Required Permission

`can-read-consolidators`

See Auth* (scopes).

Request

```
GET /consolidators/:id HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `id` : consolidator ID

Response

```
HTTP/1.1 200 OK
```

```
{
  "id": "556bd03a-0c4f-11e6-a148-3e1d05defe68",
  "name": "Test"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Consolidator id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Updating Consolidators

Required Permission

`can-update-consolidators`

See Auth* (scopes).

Request

```
PUT /consolidators/:id HTTP/1.1
Content-Type: application/json
Authorization: Bearer <bearer token>
```

```
{
  "name": "test",
  "details": "test 2"
}
```

Path Parameters

- `id` : consolidator ID

Response

```
HTTP/1.1 200 OK
```

```
{
  "rel": "self",
  "href":"/consolidators/00357362-3219-476e-87cd-4c05b1d1ce02"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Consolidator id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Deleting Consolidators

Required Permission

`can-delete-consolidators`

See Auth* (scopes).

Request

```
DELETE /consolidators/:id HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `id` : consolidator ID

Response

```
HTTP/1.1 200 OK
```

Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Consolidator id doesn't exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Linking Location Providers with Consolidators

Required Permission

`can-update-consolidators`

See Auth* (scopes).

Request

```
POST /consolidators/:consolidatorId/locationProviders/:locationProviderId  HTTP/1.1
```

Response

```
HTTP/1.1 201 CREATED
```

```
{
  "rel": "self",
  "href":"/consolidators/00357362-3219-476e-87cd-4c05b1d1ce02/locationProviders"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Consolidator id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Listing Location Providers Linked with Consolidators

Required Permission

`can-read-consolidators`

See Auth* (scopes).

Request

```
GET /consolidators/:id/locationProviders HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `id` : retailer ID

Response

```
HTTP/1.1 200 OK
```


```
[
  {
    "id": "65b83786-8b63-11e4-aa3e-db85c11f1a50",
    "name": "Provider 65b83786",
    "details": "Metapack hosts location providers locations",
    "locationsSource": {
      "endpoint": null
    },
    "links": [
      {
        "rel": "self",
        "href": "/locationProviders/65b83786-8b63-11e4-aa3e-db85c11f1a50"
      }
    ]
  },
  {
    "id": "b2893ba5-5417-41f1-be1a-b5656a6eb74c",
    "name": "Provider b2893ba5",
    "details": "A location provider requiring API call to get locations",
    "locationsSource": {
      "endpoint": "http://ddo-b2893ba5-locations.metapack.com"
    },
    "links": [
      {
        "rel": "self",
        "href": "/locationProviders/b2893ba5-5417-41f1-be1a-b5656a6eb74c"
      }
    ]
  },
  /* ... */
]
```


Failure

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Consolidator id does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Unlinking Location Providers from Retailers

Required Permission

`can-update-consolidators`

See Auth* (scopes).

Request

```
DELETE /consolidators/:consolidatorId/locationProviders/:locationProviderId HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "rel": "self",
  "href":"/consolidators/00357362-3219-476e-87cd-4c05b1d1ce02/locationProviders"
}
```

Failure

Request error; e.g. invalid JSON object:

```
HTTP/1.1 400 BAD REQUEST
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Consolidator ID does not exist:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

File Import

`Version: 1.1`

This document provides information for uploading location details using a file import. The two supported file import formats are CSV and JSON and the format and structure of each will be explained in the document.

Prerequisite

Users should be able to produce files in CSV and JSON as per the format documented below for a successful file upload. Users should always provide the full list of stores in every file upload, i.e. if 6 stores exist and 4 new stores are added then the file would need to contain 10 stores.

Limitations

Currently files need to be sent to customer support via email. Work is in progress to automate file processing.

CSV File Import

File format

This is a comma delimited format with each field separated with a comma (,) and encapsulated with double quotes (""). Each row in the file denotes a unique store / location and each field (column) in that row carrier specific information about that location / store. Following are the list of fields needed:

Column	Name	Mandatory	Length	Example
1	storeId	`true`	16	`"030HHE"`
2	storeName	`false`	128	`"Haywards Heath"`
3	city	`false`	64	`"Haywards Heath"`
4	postCode	`true`	16	`"RH16 1DR"`
5	countryCode	`true`	3	`"GBR"`
6	address	`true`	256	`"4 Palmerston House; Commercial Square"`
7	latitude	`true`	10	`"51.006035"`
8	longitude	`true`	10	`"-0.102897"`
9	telephoneNumber	`false`	16	`"+44 7894 465 4561"`
10	logoUrl	`false`	128	`"http://logodomain.com/nicelogo.jpg"`
11	photoUrls	`false`	512	`"http://logodomain.com/nicephoto1.jpg,http://logodoma`
12	hasDisabledAccess	`false`	Boolean	`"true"`
13	description	`false`	512	`"Nice Store"`
14	mondayOpeningTimes	`false`	-	`"09:00-15:00,16:00-21:00"`
15	tuesdayOpeningTimes	`false`	-	`"09:00-17:00"`
16	wednesdayOpeningTimes	`false`	-	`"09:00-17:00"`
17	thursdayOpeningTimes	`false`	-	`"09:00-17:00"`
18	fridayOpeningTimes	`false`	-	`"09:00-17:00"`
19	saturdayOpeningTimes	`false`	-	`"09:00-17:00"`
20	sundayOpeningTimes	`false`	-	`"09:00-17:00"`
21	tags	`false`	-	`"pharmacy,refrigerated,potato"`

Example line in the file

```
"030HHE","Haywards Heath","Haywards Heath","RH16 1DR","GBR","4 Palmerston House; Commercial Square","51.006035","-0.102897","+
44 7894 465 4561","http://logodomain.com/nicelogo.jpg","http://logodomain.com/nicephoto1.jpg,http://logodomain.com/nicephoto2.
jpg","true","Nice Store","09:00-15:00,16:00-21:00","09:00-17:00","09:00-17:00","09:00-17:00","09:00-17:00",,"09:00-17:00","pha
rmacy,refrigerated,potato"
```

JSON File Import

All fields used in the JSON format are the same as CSV, so please refer section above.

Example File


```
[
   {
      "address":"3 Gray's Inn Road",
      "city":"London",
      "countryCode":"GBR",
      "description":null,
      "hasDisabledAccess":false,
      "latitude":51.006035,
      "logoUrl":null,
      "longitude":-0.102897,
      "photoUrls":[
         "http://logodomain.com/nicephoto2.jpg"
      ],
      "postCode":"N1 1AA",
      "storeId":"STORE002",
      "storeName":"Test Store 2",
      "telephoneNumber":"+447894464567 82",
      "openingTimesRules":[
         {
            "openingClosingTimes":[
               {
                  "closingAt":"19:00",
                  "openingAt":"09:00"
               }
            ],
            "rule":"every Sunday"
         },
         {
            "openingClosingTimes":[
               {
                  "closingAt":"17:00",
                  "openingAt":"09:00"
               }
            ],
            "rule":"every Monday"
         },
         {
            "openingClosingTimes":[
               {
                  "closingAt":"17:00",
                  "openingAt":"09:00"
               }
            ],
            "rule":"every Tuesday"
         },
         {
            "openingClosingTimes":[
               {
                  "closingAt":"17:00",
                  "openingAt":"09:00"
               }
            ],
            "rule":"every Wednesday"
         },
         {
            "openingClosingTimes":[
               {
                  "closingAt":"17:00",
                  "openingAt":"09:00"
               }
            ],
            "rule":"every Thursday"
```

```
        },
        {
           "openingClosingTimes":[
              {
                 "closingAt":"17:00",
                 "openingAt":"09:00"
              }
           ],
           "rule":"every Friday"
        },
        {
           "openingClosingTimes":[
              {
                 "closingAt":"17:00",
                 "openingAt":"09:00"
              }
           ],
           "rule":"every Saturday"
        }
     ],
     "tags":[
        "pharmacy",
        "refrigerated",
        "potato"
     ]
  },
  {
     "address":"5Gray's Inn Road",
     "city":"London",
     "countryCode":"GBR",
     "description":null,
     "hasDisabledAccess":false,
     "latitude":51.006035,
     "logoUrl":null,
     "longitude":-0.102897,
     "photoUrls":[
        "http://logodomain.com/nicephoto2.jpg"
     ],
     "postCode":"N1 1AA",
     "storeId":"STORE003",
     "storeName":"Test Store 2",
     "telephoneNumber":"+44789446456782",
     "openingTimesRules":[
        {
           "openingClosingTimes":[
              {
                 "closingAt":"19:00",
                 "openingAt":"09:00"
              }
           ],
           "rule":"every Sunday"
        },
        {
           "openingClosingTimes":[
              {
                 "closingAt":"17:00",
                 "openingAt":"09:00"
              }
           ],
           "rule":"every Monday"
        },
        {
           "openingClosingTimes":[
              {
                 "closingAt":"17:00",
                 "openingAt":"09:00"
              }
           ],
           "rule":"every Tuesday"
        },
        {
           "openingClosingTimes":[
```

```
            {
               "closingAt":"17:00",
               "openingAt":"09:00"
            }
         ],
         "rule":"every Wednesday"
      },
      {
         "openingClosingTimes":[
            {
               "closingAt":"17:00",
               "openingAt":"09:00"
            }
         ],
         "rule":"every Thursday"
      },
      {
         "openingClosingTimes":[
            {
               "closingAt":"17:00",
               "openingAt":"09:00"
            }
         ],
         "rule":"every Friday"
      },
      {
         "openingClosingTimes":[
            {
               "closingAt":"17:00",
               "openingAt":"09:00"
            }
         ],
         "rule":"every Saturday"
      }
   ],
   "tags":null
}
]
```

File Name Format

For this version of the release the file-name has no significance though is recommended to be as follows:

```
RetailerName-YYYYMMddHHmm.[csv|json]
```

Authentication and Authorisation

Metapack Location services uses the OAuth 2.0 protocol for authentication (identifying a user) and authorization (validating that a given user can access an specific API call).

To be able to access the API calls described in Location Services API examples and Location Services User's Guide, the client must first generate a **bearer token** which identifies the user and his permissions.

Accessing the Location Services API without a bearer token will return :

```
HTTP/1.1 401 UNAUTHORIZED
```

Generating a bearer token

To retrieve a bearer token, the client must first perform a `POST` request to `https://<environment url>/oauth` with body `"grant_type=client_credentials"` and content type `"application/x-www-form-urlencoded"`. This request needs to provide the user credentials, available in a provided `apiKey.properties` file, using basic authentication.

`apiKey.properties`

```
apiKey.id = (api key id value)
apiKey.secret = (api key secret value)
```

The http basic access authentication method is used by providing an `Authorization` request header with the `"Basic "` prefix followed by a `base64` encoded `"<username>:<password>"` string.

In our example:

```
base64Encode("1DR3434KDFJLR32L53L4235J6:wqe34tyyh7679jokkDa$7df46bbbbjui+b7-sdfsdfY")
    = "MURSMzQzNEtERkpMUjMyTDUzTDQyMzVKNjp3cWUzNHR5eWg3Njc5am9ra0RhJDdkZjQ2YmJiYmp1aStiNy1zZGZzZGZZ"
```

The resulting successful response body will consist of a JSON object with the following fields:

Field Name	Description
`access_token`	The actual token value to use calls that require OAuth2.0 authentication
`expires_in`	TTL (seconds)
`scope`	Space separated list of permissions
`token_type`	OAuth2.0 token type

Example

Request

```
POST /oauth HTTP/1.1
Host: dmo.metapack.com
Authorization: Basic MURSMzQzNEtERkpMUjMyTDUzTDQyMzVKNjp3cWUzNHR5eWg3Njc5am9ra0RhJDdkZjQ2YmJiYmp1aStiNy1zZGZzZGZZ
Content-Type: application/x-www-form-urlencoded
grant_type=client_credentials
```

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Headers: X-Requested-With,Content-Type
Access-Control-Allow-Methods: GET,PUT,POST,DELETE
Access-Control-Allow-Origin: *
Content-Length: 591
Content-Type: application/json; charset=utf-8
Date: Wed, 14 Jan 2015 14:42:51 GMT
X-Powered-By: Express
```

```
{
    "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1R1oyS0NWQzBXTDJMM1E1UVdITjA4SUw1IiwiaXNzIjoiaHR0cHM6Ly9hcGkuc3Rvcm1wYXRoLmNvbS92MS9hcHBsaWNhdGlvbnMvN01ibXRzWVRhSjBQOEY0bjBseHVJOSIsImlhdCI6MTQyMTI0NjU3MSwiZXhwIjoxNDIxMjUwMTcxLCJzY29wZSI6ImNhbi1jcmVhdGUtbG9jYXRpb25zIGNhbi1kZWxldGUtb3duLWxvY2F0aW9ucyBjYW4tdXBkYXRlLW93bi1sb2NhdGlvbnMgY2FuLWNyZWF0ZS1vd24tbG9jYXRpb24tcnVsZXMgIn0.HMoDne2AHWcAQCghCWbGRVbZeXITMfVA31t8HPUP4Rw",
    "expires_in": 3600,
    "scope": "can-create-locations can-delete-own-locations can-update-own-locations can-create-own-location-rules ",
    "token_type": "bearer"
}
```

Permissions

A user can have bespoke permissions if needed, but by default, they are attached to the role he belongs to. The OAuth2.0 token scope contains the list of permissions and it's also available in the `/oauth` response.

This is the list of permissions per role, as configured in out user repository.

`location-provider`

```
{
  "permissions": [
    "can-create-locations",
    "can-retrieve-locations",
    "can-update-locations",
    "can-delete-own-locations",
    "can-manage-opening-times-rules",
    "can-search-locations"
  ]
}
```

`retailer`

```
{
  "permissions": [
    "can-retrieve-locations",
    "can-search-locations",
    "can-manage-opening-times-rules"
  ]
}
```

`metapack`

```
{
  "permissions": [
    "can-geocode",
    "can-create-locations",
    "can-retrieve-locations",
    "can-update-locations",
    "can-delete-own-locations",
    "can-manage-opening-times-rules",
    "can-search-locations",
    "can-delete-all-locations",
    "can-list-retailers",
    "can-edit-retailer",
    "can-create-retailers",
    "can-read-retailers",
    "can-update-retailers",
    "can-delete-retailers",
    "can-create-consolidators",
    "can-read-consolidators",
    "can-update-consolidators",
    "can-delete-consolidators",
    "can-list-location-providers",
    "can-create-location-providers",
    "can-update-location-providers",
    "can-retrieve-location-providers",
    "can-manage-accounts",
    "can-create-configurations",
    "can-list-configurations",
    "can-read-configurations"
  ]
}
```


`metapack-mgmt`


```
{
  "permissions": [
    "can-search-locations",
    "can-list-retailers",
    "can-edit-retailer",
    "can-create-retailers",
    "can-read-retailers",
    "can-update-retailers",
    "can-delete-retailers",
    "can-create-consolidators",
    "can-read-consolidators",
    "can-update-consolidators",
    "can-delete-consolidators",
    "can-list-location-providers",
    "can-create-location-providers",
    "can-update-location-providers",
    "can-retrieve-location-providers",
    "can-manage-accounts",
    "can-create-configurations",
    "can-list-configurations",
    "can-read-configurations",
    "can-delete-all-locations"
  ]
}
```

Accessing the Location Services API

The bearer token duration (seconds) is available in the `"expires_in"` field. In our example, we have a TTL of 3600s (1h). This means that we can use the **same token** for all Location Services API calls in that period of time. After this, it expires and we need to generate a new one.

The Location Services API calls use the OAuth 2.0 *bearer token* authorization schema. This includes all operations such as creating, retrieving, changing and deleting locations, each of which, represented by a permission that needs to be present on the `scope` JSON field value.

In practical terms, all requests to Location Services must have an `"Authorization"` request header with the `"Bearer "` prefix, followed by the token (copied from `access_token` JSON field value).

Example

Request

```
GET /dmoptions/locations/e8ca062c-6921-43d5-a857-aea5ca562690 HTTP/1.1
Host: dmo.metapack.com
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1R1oyS0NWQzBXTDJMM1E1UVdITjA4SUw1IiwiaXNzIjoiaHR0cHM6Ly9
hcGkuc3Rvcm1wYXRoLmNvbS92MS9hcHBsaWNhdGlvbnMvN01ibXRzWVRhSjBQOEY0bjBseHVJOSIsImlhdCI6MTQyMTI0NjU3MSwiZXhwIjoxNDIxMjUwMTcxLCJzY
29wZSI6ImNhbi1jckVhdGUtbG9jYXRpb25zIGNhbi1kZWxldGUtb3duLWxvY2F0aW9ucyBjYW4tdXBkYXRlLW93bi1sb2NhdGlvbnMgY2FuLWNyZWF0ZS1vd24tbG9
jYXRpb24tcnVsZXMgIn0.HMoDne2AHWcAQCghCWbGRVbZeXITMfVA31t8HPUP4Rw
```

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Headers: X-Requested-With,Content-Type
Access-Control-Allow-Methods: GET,PUT,POST,DELETE
Access-Control-Allow-Origin: *
Content-Type: application/json
Date: Thu, 15 Jan 2015 16:35:47 GMT
Server: Jetty(9.2.z-SNAPSHOT)
Transfer-Encoding: chunked
```

```
{
    "address": "Doddle Waterloo; Waterloo Station; Waterloo Road",
    "city": "London",
    "countryCode": "GBR",
    "description": null,
    "hasDisabledAccess": false,
    "id": "e8ca062c-6921-43d5-a857-aea5ca562690",
    "latitude": 51.502903,
    "links": [
        {
            "href": "/locations/e8ca062c-6921-43d5-a857-aea5ca562690",
            "rel": "self"
        },
        {
            "href": "/locations/e8ca062c-6921-43d5-a857-aea5ca562690/details",
            "rel": "details"
        }
    ],
    "locationProvider": {
        "id": "5b1f0987-6636-4bfb-b4e0-0c8351f84780",
        "name": "Doddle"
    },
    "logoUrl": null,
    "longitude": -0.1129436,
    "photoUrls": [],
    "postCode": "SE1 7LY",
    "storeId": "4",
    "storeName": "Waterloo",
    "telephoneNumber": "+442076203777"
}
```

Creating User Accounts

Required Permission

`can-manage-accounts`

See Auth* (scopes).

Creating User Accounts

Request

```
POST /accounts/:role HTTP/1.1
Authorization: Bearer <bearer token>
```

```
{
  "id": "7e510d3d-a2e4-4a61-bad4-ca3be00dcf9a",
  "username": "test-username",
  "email": "test-email@gmail.com",
  "name": "Stuff"
}
```

Path Parameters

- `role` : `locationProviders` / `retailers`

Response

```
HTTP/1.1 201 CREATED
```

Failure

Missing, invalid or expired token:

```
HTTP/1.1 401 UNAUTHORIZED
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Invalid role:

```
HTTP/1.1 404 NOT FOUND
```

Inconsistent data (Stormpath groups / Users):

```
HTTP/1.1 409 CONFLICT
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Retrieving User Accounts

Request

```
GET /accounts/:role/:id HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `role`: `locationProviders` / `retailers`
- `id`: location provider ID / retailer ID

Response

```
HTTP/1.1 200 OK
```

```
{
  "apiKeys": [{
    "id": "719ZUFAK878ZYY83DCQLW1COX"
  }],
  "username": "test-username",
  "email": "test-email@gmail.com",
  "name": "Stuff",
  "createdAt": "2016-07-29T13:08:47.021Z",
  "modifiedAt": "2016-07-29T13:08:47.021Z"
}
```

Failure

Missing, invalid or expired token:

```
HTTP/1.1 401 UNAUTHORIZED
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Invalid role or resource id:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Deleting User Accounts

Request

```
DELETE /accounts/:role/:id HTTP/1.1
Authorization: Bearer <bearer token>
```

Path Parameters

- `role` : `locationProviders` / `retailers`
- `id` : location provider ID / retailer ID

Response

```
HTTP/1.1 200 OK
```

Failure

Missing, invalid or expired token:

```
HTTP/1.1 401 UNAUTHORIZED
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Invalid role or resource id:

```
HTTP/1.1 404 NOT FOUND
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

Creating API Keys

Request

```
HTTP/1.1 POST /accounts/:role/:id/apiKeys
Authorization: Bearer <bearer token>
```

Path Parameters

- `role` : `locationProviders` / `retailers`
- `id` : location provider ID / retailer ID

Response

```
HTTP/1.1 201 CREATED
```

```
{
  "id": "4GWOS9FMWAW886O2AP9JVKXBH",
  "secret": "zikp6xAwZhrWNac4JpdpVEoCi6XDxNyXRMIKcpd0pPs"
}
```

Failure

Missing, invalid or expired token:

```
HTTP/1.1 401 UNAUTHORIZED
```

User doesn't have the required permission:

```
HTTP/1.1 403 FORBIDDEN
```

Invalid role:

```
HTTP/1.1 404 NOT FOUND
```

Inconsistent data (Stormpath groups / Users):

```
HTTP/1.1 409 CONFLICT
```

Other errors:

```
HTTP/1.1 500 INTERNAL SERVER ERROR
```

References

- RFC 6749 - The OAuth 2.0 Authorization Framework
- RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage - Authenticated Requests