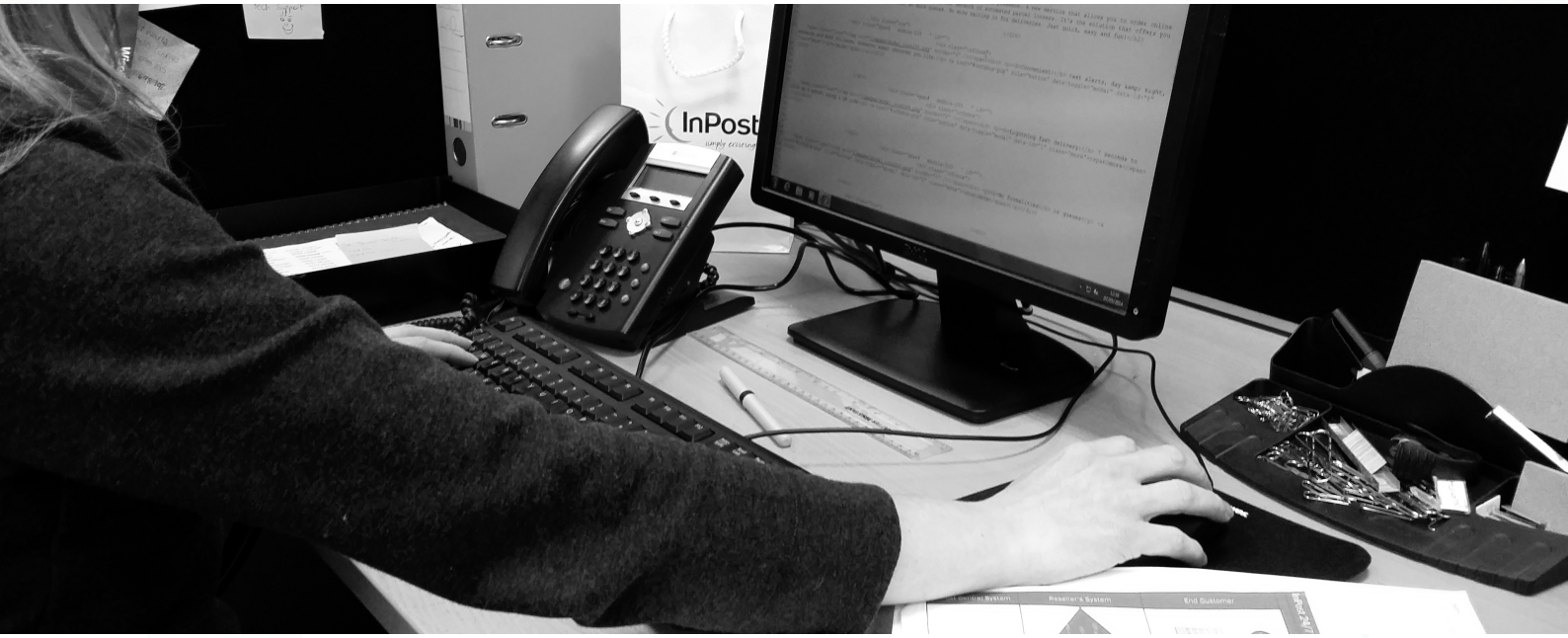
InPost UK Ltd

API Implementation Guide 2014

DOCUMENT ID: IT-EP-13-DZ-1-1.6-UK



simply exciting

Contents

1.0. Before you start

In order to integrate with the InPost system, you will require an API Key. Once an account has been created on the InPost Central System, you can request an API Key or Token which will remain valid throughout the lifetime of your InPost account. This will be required every time a request is submitted to the InPost Central System. The API Key also serves to uniquely identify each customer. It is therefore imperative that you do not share this with anyone.

The InPost API is developed using the REST (Representational State Transfer) software architecture standard. You can learn more about REST by referring to the following online resources.

http://www.ibm.com/developerworks/webservices/library/ws-restful/

http://www.infoq.com/articles/rest-introduction

1.1. Introduction

1.1.1. Requests

Requests can represent one of these methods: GET, POST, PUT and DELETE. The API endpoint is https://api-uk.easypack24.net/. Currently all structures are encoded using JSON. Implementation of other data formats is envisaged for the near future.

1.1.2. Responses

Following are the possible responses to requests:

- 200 OK – *request was processed as desired*
- 201 Created – *request processed as desired; redirected to the newly created object*
- 204 No Content – *request completed without errors; no content to be returned*
- 400 Bad Request – *request syntax error*
- 401 Unauthorized – *your credentials prohibit the processing of request*
- 404 Missing – *resource is missing*
- 422 Incorrect Request Parameters – *provided parameters are incorrect*
- 500 Internal Error – *unexpected server error occurred*

1.1.3. Resources

The basic request structure is as follows:

GET https://api-uk.easypack24.net/RESOURCE_NAME?token=[YOUR_TOKEN]

For example, in order to request the complete list of all InPost Locker Terminals in the UK, the request will be in the following format:

GET https://api-uk.easypack24.net/machines?token=[YOUR_TOKEN]

If you intend to retrieve more than one instance of a particular resource, the submitted parameters must be separated by semicolons. It is noteworthy that this option will be available only for certain resources. For example, in order to request multiple parcel labels the request will take the following form:

GET https://api-uk.easypack24.net/stickers/[PID1;PID2;PID3]?token=[YOUR_TOKEN]&format=pdf

Where PID1, PID2 and PID3 denote the unique identifiers[1] for each of the three parcels.

Please note that all resource names are in the plural form, unless otherwise stated in the documentation. Moreover, since it is understood that the token (i.e. the API Key) is an integral part of every request, this will be omitted in all the subsequent examples.

1.2. Machines

The resource *Machines*[2] can only be accessed with the GET method and returns a complete list of locker terminals located throughout the country. This would include both the operational as well as planned terminals.

GET https://api-uk.easypack24.net/machines

In order to retrieve a complete list of only the currently operational locker terminals, you can issue the request with an additional filter parameter 'status', e.g.

GET http://api-uk.easypack24.net/machines?token=[Your_Token]&status=Operating

If, however, you specify a particular machine ID, then the information pertaining only that locker terminal will be returned in the response. The machine ID is a string with a variable length of between 9 to11 characters. Table 1 lists all the fields returned in response to the a request for accessing this resource.

GET https://api-uk.easypack24.net/machines/[MACHINE_ID]

For example,

GET http://api-uk.easypack24.net/machines/UKREA1287

1 *The parcel identified is an 8 character long string e.g. DW017748*
2 *In the context of this document, the term Machine refers to an Automated Parcel Locker*

Name	Description	Field type
address	Address object containing postcode, province[3], street and city	String
id	Unique identifier for a locker terminal	String
latitude	Geographical location coordinate	Float
longitude	Geographical location coordinate	Float
location_description	Brief description of the terminal's location.	String
operating_hours	This will always appear as 24 \ /7, implying 24 Hour operation)	String
status	This can have one of the these values: Operating, NonOperating	String

Table 1: Response Fields when accessing the resource "Machines"

1.3. Forward Logistics

1.3.1. Parcel creation

In order to create a new parcel a JSON encoded POST request should be submitted to https://api-uk.easypack24.net/parcels with the parameters listed in Table 2.

Name	Description	Field type	Usage
description	A brief description of the parcel. This will appear on the final label, however it's not mandatory.	String (Max. length: 50 characters)	This can contain an order reference number.
receiver		JSON object containing email address & mobile number	The mobile number must comprise only the last 9 digits
Size	Parcel size	Char	One of these characters: A,B,C
tmp_id	Temporary parcel identifier	String	Temporary identifier for each parcel.
target_ machine	Unique identifier for the destination parcel locker	String	

Table 2: Request Parameters for Parcel Creation

3 This is a legacy variable set to be deprecated in the upcoming version of the API.

Example of a JSON encoded parcel creation request

```
[
        {
                "description": "order_xyz",
                "receiver" : {
                                "phone": "987654321",
                                "email": "receiver@example.com"
                },

                "size" : "B",
                "tmp_id" : "98543",
                "target_machine" : "UKMAN1068"
        }
 ]
```

If the request is valid, HTTP status code 201 will be returned with a JSON object containing the parcel ID, an 8 character long string. However, if invalid parameters were submitted in the request, the HTTP status code 400 will be returned with the relevant error message(s) in the JSON encoded response.

1.3.2. Existing Parcels

Details of existing parcels can be accessed by using the GET method, e.g.

GET https://api-uk.easypack24.net/parcels/[Parcel_ID]

In order to query information on multiple parcels, you must include the identifiers for each parcel, separated by a semi-colon.

GET https://api-uk.easypack24.net/parcels/[Parcel1_ID; Parcel2_ID;Parcel2_ID;]

The response contains data about the parcel(s) in JSON format, as given below.

```
[
        {

                "id":" DW017748",
                "status":"Prepared",
                "status_updated_at":"2013-09 11T16:13:05.146+01:00",
                "target_machine":"UKABE1126",
                "description":"order_xyz",
                "created_at":"2013-09-11T16:13:05.025+01:00",
                "size":"C",
                "amount":4.5,
                "amount_paid":4.5,
                "cod_amount":0,

                "receiver":{
                                "email":"recipient@xyz.com",
                                "phone":"860617883"
                },

                "sender":{"email":"sender@xyz.com"}
        }
]
```

Additional filter parameters, as listed in Table 5 can be passed to the method.

Name	Description	Field type
amount	The amount required to pay for a parcel.	Float
amount_paid	The amount that has been paid for the parcel	Float
created_at	Parcel Creation Date	DateTime
description	Additional field e.g. for order number, etc. This will appear on the final label	String (Max. Length: 50)
id	Unique Parcel Identifier	String (Fixed Length:8)
receiver	JSON object containing email address & mobile number	JSON Object
cod_amount	Cash-on-delivery amount [4]	Float
sender	JSON object containing email address	JSON Object
size	Parcel Size	Char
status	Parcel statuses	String
status_updated_at	Timestamp of last status update	DateTime
target_machine	Destination locker terminal ID	String

Table 4: Response fields when accessing the “Parcel” resource

Name	Description	Field type
end_date	End date for searching parcels	DateTime
limit	Maximum number of results to be returned. The deault limit is 100	Integer
offset	This is the pagination offset. The default value is 0.	Integer
role	This Parameter specifies the type of the client and takes one of these values: sender, receiver, both.	String
status	Parcel status	String
status_updated_at	Timestamp of the last status update	DateTime
sort_by	This parameter can have one of these values: status, created_at, status_updated_at	String
sort_order	This parameter can have one of these values: asc, desc	String
start_date	Start date for searching parcels	DateTime

Table 5: Filter Parameters for accessing the resource “Parcels”

4 This is not applicable and will be deprecated in the upcoming version

1.3.3. Parcel Payment

Payment for a single parcel can be made by submitting a POST request, e.g.

POST https://api-uk.easypack24.net/**parcels/[Parcel_ID]/pay**

You can pay for multiple parcels simultaneously, by including the relevant parcel identifiers in the request, separated by semicolons, e.g.

POST https://api-uk.easypack24.net/**parcels/[Parcel1_ID;Parcel2_ID]/pay**

Name	Description	Field type
id	Unique parcel identifier	String (Length: 16)

Table 6: Request Parameter a Parcel Payment

If the request is valid HTTP status code 204 will be returned, without any content. Otherwise, HTTP status code 400 will be returned with a list of errors.

1.3.4. Parcel Labels

The resource *Stickers* can only be accessed using the GET method. The response would contain base64 encoded content for the labels. This can be displayed in PDF format after decoding.

GET https://api-uk.easypack24.net/**stickers/[Parcel_ID]**

You can request multiple parcel labels simultaneously, by including the relevant parcel identifiers in the request, separated by semicolons, e.g.

GET https://api-uk.easypack24.net/**stickers/[Parcel1_ID;Parcel2_ID]**

Name	Description (detailed if not self-explanatory)	Field type
format	Format in which the content should be returned. Default: pdf	String
id	Unique parcel identifier	String (Length: 8)
type	Returned content type. Default: normal	String

Table 7: Request Parameters for accessing the “Stickers” resource

1.3.5. Cities

The Resource *Cities* can only be accessed using the GET method. The response contains a list of cities where at least one InPost parcel locker had been installed.
GET https://api-uk.easypack24.net/**cities**

1.3.6. Customer/Pricelist

The resource *Customer/Pricelist* can only be accessed through the GET method. The response contains applicable prices for parcels sized A (small), B (medium), and C (large).

GET https://api-uk.easypack24.net/**customer/pricelist**

Name	Description	Field type
standard	JSON object	service_pricelist_object containing a & b & c

Table 8: Response field when accessing the "Customer/Pricelist" resource

1.3.7. Parcel update

To update a parcel a PUT request must be submitted to https://api-uk.easypack24.net/parcels with the parameters listed in Table 8, included (in JSON format):

Name	Description	Field type
	Updated brief description of the parcel. This will appear on the final label, however it's not mandatory.	String (Max. length: 50 characters)
id	Unique parcel identifier	String (Length: 8)
size	New parcel size	Char
status	New status. The status of any parcels in the "created" state can be updated to "Cancelled". This will prohibit further processing.	String

Table 9: Parameters for Parcel Update Request

If the request is valid HTTP status code 204 will be returned, without content. Otherwise HTTP status code 400 will be returned with a list of errors.

1.4. Reverse Logistics

1.4.1. Generating an Active code

In order to create a new parcel for return, a JSON encoded POST request should be submitted to **https://api-uk.easypack24.net/reverselogistics.json** with the parameters listed in Table 10. It is noteworthy that if the address parameter is not included in the request, the parcel will be delivered to the default address of the account to which the API key is registered.

Name	Description	Field type
rma	Return Merchandise Authorisation (RMA) Code *(Optional)*	String
parcel_size	Parcel Size *(Required)*	Char (A, B or C)
expire_at	Expiry Date - This is a required parameter and can be any date within the next 2 years *(Required)*	Date (yyyy-mm-dd)
sender_phone	Last 10 digits of the sender's mobile number *(Required)*	String
sender_email	Sender's email address *(Required)*	String
with_label		Boolean (TRUE or FALSE)
additional_description_1	Optional Field	String
additional_description_2	Optional Field	String
additional_description_3	Optional Field	String
address	Delivery Address (Optional)	Array

Table 10: Request Parameters for Generating an Activation Code

The response contains data about the parcel created for return, in JSON format. If the request is valid, HTTP status code 201 will be returned with a JSON object containing a unique 10 digit code, the status of the parcel, its size and expiry date and time. However, if invalid parameters were submitted in the request, the HTTP status code 400 will be returned with the relevant error message(s) in the JSON encoded response.

```
{
code: 8055401956,
is_active: true,
parcel_size: "A",
expire_at: "2014-02-01T00:00:00+01:00"
}
```

1.4.2. Creating a Return Label

While a return label in PDF format will be automatically emailed to the sender, a GET request can be issued to retrieve this from the InPost Central System with the 10 digit unique code as a request parameter. The response would contain base64 encoded content for the labels. This can be displayed in PDF format after decoding.

GET https://api-uk.easypack24.net/reverselogistics/{code}/label.json

1.4.3. Retrieve a Returns Report

A GET request can be submitted to retrieve consignment reports for all returned parcels.

GET https://api-uk.easypack24.net/reverselogistics.json

Name	Description	Field type
page	Page Number	Integer
per_page	Number of consignment reports per page	Integer
date_before	Start date filter parameter	Date (yyyy-mm-dd)
date_after	End date filter parameter	Date (yyyy-mm-dd)
is_active	Status of Return Code (Always TRUE)	Boolean (true, false)

Table 12: Filter Parameters for accessing consignment report for returned parcels

A typical response in JSON format would be the following.

```
{
_meta: {
total_count: 130,
page: "1",
per_page: "2"
},
data: [{
code: 2169717815,
created_at: "2012-10-17T10:19:41.036+02:00",
sender_phone: "604056593",
sender_email: "dbartkowski@inpost.pl",
sent_at: "2012-10-17T13:09:13.678+02:00",
parcel: {
id: "600000291299999029900001",
href:                                    "http://api30.loc/app_dev.php/
parcels/600000291299999029900001?
locale=pl&token=63a44fb7-19ce-47f8-9de2-ff0f6bda2df4",
status: "CustomerDelivering"
}},
{
code: 1752327483,
created_at: "2012-10-17T10:19:40.265+02:00",
sender_phone: "604056593",
sender_email: "dbartkowski@inpost.pl",
sent_at: "2012-10-17T13:48:05.814+02:00",
parcel: {
id: "600000291299999029900002",
href:                                    "http://api30.loc/app_dev.php/
parcels/600000291299999029900002?
locale=pl&token=63a44fb7-19ce-47f8-9de2-ff0f6bda2df4",
status: "CustomerDelivering"
}}]}
```

1.4.4. Retrieve a Report by Code

A GET request with a 10 digit code parameter can be submitted to retrieve a consignment report for that particular returned parcel.

GET https://api-uk.easypack24.net/reverselogistics/{code}.json

A typical response in JSON format would be the following.

```
{
code: 1306291025,
created_at: "2013-10-09T15:38:56.150+02:00",
sender_phone: "888000000",
sender_email: "test@inpost.pl",
sent_at: "2014-01-09T14:18:12.367+01:00",
parcel: {
id: "602776009499999019900061",
href:                                    "http://api30.loc/app_dev.php/
parcels/602776009499999019900061?
locale=uk&token=63a44fb7-19ce-47f8-9de2-ff0f6bda2df4",
status: "Missing"
}
}
```