

PenCarrie API: Integration Guide

- 1 [Overview](#)
- 2 [Scope](#)
- 3 [Who is this guide for?](#)
- 4 [Security](#)
 - 4.1 [IP Addresses](#)
 - 4.2 [Transport Format](#)
 - 4.3 [Request Protocols](#)
- 5 [API Addresses](#)
- 6 [API Functional Limitations](#)
- 7 [Function Overview](#)
 - 7.1 [Rate Limits](#)
 - 7.2 [A note on Parameters](#)
- 8 [Functions](#)
 - 8.1 [pcgetstock](#)
 - 8.2 [pcgetstockprices](#)
 - 8.3 [pcgetstockcsv](#)
 - 8.4 [pcget](#)
 - 8.5 [pclist](#)
 - 8.6 [pcautoorder](#)
 - 8.7 [pcupdate_new](#)
 - 8.8 [pcupdate](#)
 - 8.9 [pcallocate](#)
 - 8.10 [Appendices](#)
 - 8.10.1 [Appendix A: pcautoorder XML](#)
 - 8.10.2 [Appendix B: pcupdate XML](#)
 - 8.10.3 [Appendix C: PCGetStock / PCGetStockPrices field list](#)
 - 8.10.4 [Appendix D: Delivery Codes \(del_tid\)](#)
 - 8.11 [References](#)
 - 8.12 [Version History](#)

Overview [🔗](#)

This document contains integration details and descriptions of the interactions that can be performed against the PenCarrie API. The integration guide is intended for technical readers to develop against the PenCarrie API, in order to achieve direct integration with the PenCarrie order management system. The following sections will outline the features and exposed functions of the PenCarrie API that can be used within the public domain, their expected function and the inputs and outputs of the various calls.

Scope [🔗](#)

The primary function of the PenCarrie API is **order management**. With the API you can integrate into your own systems the ability to create, amend, complete, or cancel orders.

While the API does offer rudimentary product data (styles, colour/size variations, stock levels), if you wish to utilise our full product data we can offer this in various formats, that you can retrieve via the product API (a separately managed API, more details here: <https://www.pencarrie.com/marketing/website-options/data/enhanced-data>), or we can arrange delivery/collection via FTP, contact your sales representative for information on these other options.

Who is this guide for? [🔗](#)

In order to utilise the PenCarrie API, you should have, or have access to, expertise in programming languages and some experience utilising APIs in general. We also have options for placing orders by CSV, and delivering product or stock levels, contact your sales representative for more information.

Table 1: API Service suitability [🔗](#)

Technical Resource → Service ↓	Basic (regular pc user, comfortable with emails)	Adept (comfortable editing multiple file formats)	Expert (coding experience)
Stock/Ordering API	✗	✗	✓
Ordering CSV	✗	✓	✓
Stock/Product (S)FTP	✗	✓	✓
Stock/Product Emails	✓	✓	✓

This document covers "Stock/Ordering API", contact your sales representative for information on other options.

Security [🔗](#)

IP Addresses [🔗](#)

All calls to the PenCarrie API must originate from a static or fixed **IPv4** address that has been whitelisted by the PenCarrie technical team. Whitelisting enhances security by ensuring only authorised IP addresses can make requests, while also allowing PenCarrie to identify from whom requests are being made and allow them to be processed. To have your IP address whitelisted please contact your sales representative.

Transport Format [🔗](#)

The PenCarrie API is reliant on using the platform agnostic transport format: XML. This ensures that you and/or your client(s) are not limited by a provider specific transportation format and can make requests from any platform you deem appropriate. It should be noted that, in addition to IP address checking, PenCarrie will also ensure that the IP address corresponds to the customer code provided in the XML string.

Request Protocols [🔗](#)

All calls to PenCarrie should be made through the HTTPS secure protocol. The requests should each be made via the **POST** web verb, which will allow you to provide the required XML arguments / content in the request body.

API Addresses [🔗](#)

The following addresses should be used for the development, testing and go-live stages of integrations with PenCarrie. Please ensure that all development and testing is completed against the sandbox website prior to switching to live. You must inform PenCarrie of the intention to move to the live web service prior to go-live.

Development URL: <https://sandbox.pencarrie.com/gateway>

Live URL: <https://pencarrie.com/gateway>

API Functional Limitations [↗](#)

The API provided by PenCarrie is intended to allow third parties to perform stock level checks on available garments in addition to providing functions to support order creation and management. An order may only contain one line for each product variant.

Function Overview [↗](#)

Table 2: available functions [↗](#)

Function Name	Details
<code>pcgetstock</code>	Returns the stock level and detail of a particular style of product
<code>pcgetstockprices</code>	Returns the stock details of a particular style of product including your prices
<code>pcgetstockcsv</code>	Returns all stock on the system as a CSV, containing the SKU and free to sell value
<code>pcget</code>	Returns an order from the system based on your customer code and PC Order Code
<code>pclist</code>	Returns the header information of all orders for your customer code
<code>pcautoorder</code>	Generates a new order based on XML representation
<code>pcupdate_new</code>	Creates a new empty, open order using the account default information for new orders on your account
<code>pcupdate</code>	Updates the header details of an order if it is in an editable state
<code>pcallocate</code>	Allocates and deallocates stock to an editable order with the specified PC Order Code

Rate Limits [↗](#)

[placeholder text for rate limits - information pending]

A note on Parameters [↗](#)

Every parameter specified under each **Function** must be it's own discrete parameter defined using form-data or other appropriate method.

Customer Code [↗](#)

All functions require a `code` parameter. Throughout this document this is detailed as "CUSTCODE", but should be replaced with your organisation's customer code, most commonly a 4-8 character alphanumeric string.

Function and parameter names [↗](#)

Function names and parameter names should be lowercase.

XML [↗](#)

Many of the API functions require an XML-format parameter, often describing an object such as an order or a product allocation. These XML strings do **not** require an XML prolog, **must** follow the structure shown, and **all** element/attributes therein should be in named in lowercase as per the examples.

Functions [↗](#)

The following code samples are provided by PenCarrie, and demonstrate how to make simple web requests to the PenCarrie API. Some areas of code have been omitted for reasons of clarity. Samples provided with the command line tool cURL¹ and languages Python², PHP³, and C#⁴. Note Python code samples use the 'requests'⁵ library, C# code samples use the 'RestSharp'⁶ NuGet package.

pcgetstock [↗](#)

Returns the stock details of a particular style of product. This function is designed for getting stock data on demand for individual products and should not be used for bulk updates – the pcgetstockcsv function is provided for bulk updates. Excessive use of this function (see section on [rate limits](#)) may result in your API access being temporarily revoked.

Parameters [↗](#)

Field	Value	Description
function	pcgetstock	function name
code	[your customer code]	e.g. "ABCD"
args[0]	[product style code]	the product style/family code, e.g. "SS11" would include all SS11 colour and size variations

Response [↗](#)

XML containing colours, colour information, sizes and the free-to-sell quantities of each of the size / colour combination. The supplier parameter displays the supplier's free-to-sell stock where available. A full field list is provided in [Appendix C](#).

```
1 <stock leadtime="+6">
2   <colours>
3     <colour colourname="BLK" stat="-">
4       <cwyr>
5         <c longname="Black" head="BLK" hex="0F1315"/>
6       </cwyr>
7       <sizes>
8         <p size="S" fts="703" nextday="0" cap="500" mystat="-" supplier="1404"/>
9         <p size="M" fts="3036" nextday="0" cap="500" mystat="-" supplier="1260"/>
10        <p size="L" fts="4137" nextday="0" cap="500" mystat="-" supplier="11916"/>
11        <p size="XL" fts="2976" nextday="0" cap="500" mystat="-" supplier="2520"/>
12        <p size="XXL" fts="1619" nextday="0" cap="500" mystat="-" supplier="2232"/>
13        <p size="3XL" fts="414" nextday="0" cap="500" mystat="-" supplier="1692"/>
14        <p size="4XL" fts="-8" nextday="0" cap="0" mystat="-" />
15        <p size="5XL" fts="-3" nextday="0" cap="0" mystat="-" />
16      </sizes>
17    </colour>
18    <colour colourname="BOT" stat="-">
19      <cwyr>
20        <c longname="Bottle Green" head="BOT" hex="15451A"/>
21      </cwyr>
22      <sizes>
23        <p size="S" fts="115" nextday="0" cap="500" mystat="-" supplier="1116"/>
24        <p size="M" fts="467" nextday="0" cap="500" mystat="-" supplier="540"/>
25        <p size="L" fts="470" nextday="0" cap="500" mystat="-" supplier="1008"/>
26        <p size="XL" fts="208" nextday="0" cap="500" mystat="-" supplier="1800"/>
27        <p size="XXL" fts="101" nextday="0" cap="500" mystat="-" supplier="2916"/>
28        <p size="3XL" fts="100" nextday="0" cap="0" mystat="-" supplier="1080"/>
29      </sizes>
30    </colour>
31  </colours>
32 </stock>
```

Code samples [🔗](#)

Scenario: *"I want to see current stock for all JH001 hoodies"* [🔗](#)

cURL:

```
1 curl -X POST -F function=pcgetstock -F code=CUSTCODE -F args[0]=JH001 -L https://sandbox.pencarrie.com/gateway
```

Python:

```
1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcgetstock", "code": "CUSTCODE", "args[0]": "JH001"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)
```

PHP:

```
1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array(
4     "function" => "pcgetstock",
5     "code" => "CUSTCODE",
6     "args[0]" => "JH001"
7 );
8 $response = file_get_contents($url, false, stream_context_create(array(
9     'http' => array(
10         'method' => 'POST',
11         'header' => 'Content-type: application/x-www-form-urlencoded',
12         'content' => http_build_query($data)
13     )
14 )));
15 echo $response;
16 ?>
```

C#:

```
1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcgetstock")
5     .AddParameter("code", "CUSTCODE")
6     .AddParameter("args[0]", "JH001");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();
```

pcgetstockprices [🔗](#)

Returns the stock and price details of a particular style of product. Like pcgetstock, this should only be used for on-demand queries. If you require a full pricelist, this can be requested from our Customer Services team in Excel or CSV format. A full field list is provided in [Appendix C](#).

Parameters [🔗](#)

Field	Value	Description
function	pcgetstockprices	function name

code	[your customer code]	e.g. "ABCD"
args[0]	[product style code]	the product style/family code, e.g. "SS11" would include all SS11 colour and size variations

Response [🔗](#)

XML containing colours, colour information, sizes, prices, and the free-to-sell quantities of each of the size / colour combination. A full field list is provided in [Appendix C](#).

```

1  <stock leadtime="+6">
2    <colours stat=".">
3      <colour colourname="BLK">
4        <cwyr>
5          <c longname="Black" head="BLK" hex="0F1315" />
6        </cwyr>
7        <sizes>
8          <p size="S" fts="1221" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75"
single="5.15" supplier="2052" />
9          <p size="M" fts="3395" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75"
single="5.15" supplier="2412" />
10         <p size="L" fts="3950" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75"
single="5.15" supplier="4536" />
11         <p size="XL" fts="2708" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75"
single="5.15" supplier="2124" />
12         <p size="XXL" fts="1670" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75"
single="5.15" supplier="3096" />
13         <p size="3XL" fts="369" nextday="0" cap="500" mystat="-" carton="6.40" pack="6.75"
single="7.20" supplier="1260" />
14         <p size="4XL" fts="230" nextday="0" cap="0" mystat="-" carton="6.40" pack="6.75" single="7.20"
supplier="228" />
15         <p size="5XL" fts="65" nextday="0" cap="0" mystat="-" carton="6.40" pack="6.75" single="7.20"
supplier="372" />
16       </sizes>
17     </colour>
18     <colour colourname="BOT">
19       <cwyr>
20         <c longname="Bottle Green" head="BOT" hex="15451A" />
21       </cwyr>
22       <sizes>
23         <p size="S" fts="265" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75" single="5.15"
supplier="144" />
24         <p size="M" fts="534" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75" single="5.15"
supplier="432" />
25         <p size="L" fts="483" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75" single="5.15"
supplier="1548" />
26         <p size="XL" fts="389" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75"
single="5.15" supplier="792" />
27         <p size="XXL" fts="269" nextday="0" cap="500" mystat="-" carton="4.45" pack="4.75"
single="5.15" supplier="5580" />
28         <p size="3XL" fts="100" nextday="0" cap="0" mystat="-" carton="6.40" pack="6.75" single="7.20"
supplier="828" />
29       </sizes>
30     </colour>
31   </colours>
32 </stock>

```

Code samples [↗](#)

Scenario: "I want to see current stock and my custom prices for all JH001 hoodies" [↗](#)

cURL:

```
1 curl -X POST -F function=pcgetstockprices -F code=CUSTCODE -F args[0]=JH001 -L
https://sandbox.pencarrie.com/gateway
```

Python:

```
1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcgetstockprices", "code": "CUSTCODE", "args[0]": "JH001"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)
```

PHP:

```
1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array(
4     "function" => "pcgetstockprices",
5     "code" => "CUSTCODE",
6     "args[0]" => "JH001"
7 );
8 $response = file_get_contents($url, false, stream_context_create(array(
9     'http' => array(
10         'method' => 'POST',
11         'header' => 'Content-type: application/x-www-form-urlencoded',
12         'content' => http_build_query($data)
13     )
14 ));
15 echo $response;
16 ?>
```

C#:

```
1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcgetstockprices")
5     .AddParameter("code", "CUSTCODE")
6     .AddParameter("args[0]", "JH001");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();
```

pcgetstockcsv [↗](#)

Returns all stock on the system as a CSV, containing the SKU and free to sell value. This is designed for performing a periodic bulk update of all our stock levels in a back-end database. Please Note: due to the amount of data returned, calls to pcgetstockcsv are limited to once per fifteen minutes.

Parameters [↗](#)

Field	Value	Description
function	pcgetstockcsv	function name

code	[your customer code]	e.g. "ABCD"
------	----------------------	-------------

Response [↗](#)

CSV style document containing the product code and the free-to-sell (FTS) quantities. There is one product entry per line.

```

1 001M BLK 28/R,32
2 001M BLK 28/L,23
3 001M BLK 30/R,44
4 001M BLK 30/L,11
5 001M BLK 32/R,104
6 001M BLK 32/L,39
7 001M BLK 34/R,106
8 001M BLK 34/L,36
9 001M BLK 36/R,63
10 001M BLK 36/L,14
11 001M BLK 38/R,61
12 001M BLK 38/L,33
13 001M BLK 40/R,32
14 001M BLK 40/L,17
15 001M BLK 42/R,28

```

Code samples [↗](#)

Scenario: *"I need a full up-to-date stock list to load into my own system"* [↗](#)

cURL:

```
1 curl -X POST -F function=pcgetstockcsv -F code=CUSTCODE -L https://sandbox.pencarrie.com/gateway
```

Python:

```

1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcgetstockcsv", "code": "CUSTCODE"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)

```

PHP:

```

1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pcgetstockcsv", "code" => "CUSTCODE");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 )));
11 echo $response;
12 ?>

```

C#:

```

1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcgetstockcsv")

```

```

5 .AddParameter("code", "CUSTCODE");
6 RestResponse response = client.Execute(request);
7 File.WriteAllText(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments), "stock.csv"),
  response.Content);

```

pcget [↗](#)

Returns an order from the system based on your customer code and PC Order Code Parameters. A full field list is provided in [Appendix C](#).

Parameters [↗](#)

Field	Value	Description
function	pcget	function name
code	[your customer code]	e.g. "ABCD"
ordcode	[pc_ordercode]	the system identifier (pc_ordercode) of your order

Response [↗](#)

XML containing the order information recorded on the order header and order lines with the given PenCarrie order code and customer code.

```

1 <order code="CUSTCODE" pc_ordercode="CUSTCODE20240508115533" ordno="6346656" invcountrycurr="GBP"
  ordval="121.5" ordvat="0" totval="121.5" vatval="24.3" duty="0.00" duty_invoiced="0" additional_fees="0.00"
  ordno1="6346656" custorderno="vouchertest" revno="1" pcref="test user" invcompany="Pencarrie Tests Accounts"
  invprincipal="Testing - Stuff" invadd1="Pencarrie House, " invadd2="South View Estate, " invadd3="Willand,"
  invadd4="Cullompton," invadd5="" invpostcode="EX15 2QW" invcountrycode="GB" deladdcode="CUSTCODE20150311130022"
  delcompany="Pencarrie Tests Accounts" deladd1="OWN CARRIER" deladd2="PenCarrie Limited" deladd3="South View
  Estate" deladd4="Willand Town" deladd5="Cullompton: devon" delpostcode="EX15 2QW" delcountrycode="GB"
  cust_recipient="-" cust_add1="-" cust_add2="-" cust_add3="-" cust_add4="-" cust_add5="-" cust_postcode="-"
  cust_countrycode="-" gateway="-" gatewayref="-" cust_del_tid="0" invcarriercode="-" del_service="Own Carrier"
  del_tid="100" delcarrier="ALT" del_otid="100" plaincover="False" plaincoverselection="0" invcardtid="0"
  token="-" cardtype="-" last4digits="-" expirydate="-" invamounttaken="0" invcomment="(A)" dflag="False"
  pflag="False" xflag="False" cflag="False" status="P" sdate="20240508000000" odate="20240712094452" user="test
  user" w_createdate="20240508115533" review="0" ordstat="Pending" canedit="-1" editableordertype="1"
  txtalert="true" txtnumber="-" lastauthorizationamount="-" lastauthorizationdate="-" del_val="0" del_valvat="0"
  despatchnoteavailable="1" orderacknowledgementavailable="1" proformaavailable="1" recalcrequired="0"
  lockedby="-" ordtype="P" trackntrace="Your order has not yet been shipped." line_count="1" cancel="0">
2   <items>
3     <item pc_ordercode="CUSTCODE20240508115533" cref="vouchertest" net="0" vat="0" vatcode="HS1" lrevno="1"
      sku="01170 BLK S" longcolour="Black" description="SOLS Ladies Ride jacket" pc_cartons="0" pc_packs="0"
      pc_singles="0" pc_cartonprice="26.45" pc_packprice="27.45" pc_singleprice="28.45" orderquan="0" qty="0"
      backord="0" longcolour1="Black" picked="0" sortorder="70077814" bonet="0" bo_cartons="0" bo_packs="0"
      bo_singles="0"/>
4     <item pc_ordercode="CUSTCODE20240508115533" cref="test user" net="121.5" vat="24.3" vatcode="HS1"
      lrevno="0" sku="TEST BOT S" longcolour="Bottle" description="Fruit Loom 65/35 Pique Polo" pc_cartons="36"
      pc_packs="0" pc_singles="0" pc_cartonprice="3.75" pc_packprice="3.75" pc_singleprice="3.75" orderquan="36"
      qty="36" backord="0" longcolour1="Bottle" picked="0" sortorder="68841527" bonet="0" bo_cartons="0" bo_packs="0"
      bo_singles="0"/>
5   </items>
6   <emails>
7     <email pc_ordercode="CUSTCODE20240508115533" sn_email="test@pencarrie.com" contact="test user"/>
8   </emails>
9 </order>

```

Code samples [🔗](#)

Scenario: “View all the details of *pc_ordercode CUSTCODE20240508115533*” [🔗](#)

cURL:

```
1 curl -X POST -F function=pcget -F code=CUSTCODE -F ordcode=CUSTCODE20240508115533 -L
https://sandbox.pencarrie.com/gateway
```

Python:

```
1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcget", "code": "CUSTCODE", "ordcode": "CUSTCODE20240508115533"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)
```

PHP:

```
1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array(
4     "function" => "pcget",
5     "code" => "CUSTCODE",
6     "ordcode" => "CUSTCODE20240508115533"
7 );
8 $response = file_get_contents($url, false, stream_context_create(array(
9     'http' => array(
10         'method' => 'POST',
11         'header' => 'Content-type: application/x-www-form-urlencoded',
12         'content' => http_build_query($data)
13     )
14 )));
15 echo $response;
16 ?>
```

C#:

```
1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcget")
5     .AddParameter("code", "CUSTCODE");
6     .AddParameter("ordcode", "CUSTCODE20240508115533");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();
```

Scenario: “I need the delivery address details of *pc_ordercode CUSTCODE20240508115533*” [🔗](#)

C#:

```
1 private static void Main()
2 {
3     var options = new RestClientOptions("https://sandbox.pencarrie.com");
4     using var client = new RestClient(options);
5     var request = new RestRequest("/gateway", Method.Post)
6         .AddParameter("function", "pcget")
7         .AddParameter("code", "CUSTCODE");
8         .AddParameter("ordcode", "CUSTCODE20240508115533");
9     RestResponse response = client.Execute(request);
```

```

10
11     XDocument xDoc = XDocument.Parse(response.Content);
12     XElement orderNode = xDoc.Root;
13     Dictionary<string, string> addressAttributes = ExtractAddressAttributes(orderNode);
14
15     foreach (KeyValuePair<string, string> kvp in addressAttributes)
16     {
17         Console.WriteLine($"{kvp.Key}: {kvp.Value}");
18     }
19 }
20
21 static Dictionary<string, string> ExtractAddressAttributes(XElement orderNode)
22 {
23     Dictionary<string, string> addressAttributes = new Dictionary<string, string>();
24     XElement delAddressNode = orderNode.Element("order");
25     if (delAddressNode != null)
26     {
27         addressAttributes.Add("DelCompany", delAddressNode.Attribute("delcompany").Value);
28         addressAttributes.Add("DelAdd1", delAddressNode.Attribute("deladd1").Value);
29         addressAttributes.Add("DelAdd2", delAddressNode.Attribute("deladd2").Value);
30         addressAttributes.Add("DelAdd3", delAddressNode.Attribute("deladd3").Value);
31         addressAttributes.Add("DelAdd4", delAddressNode.Attribute("deladd4").Value);
32         addressAttributes.Add("DelAdd5", delAddressNode.Attribute("deladd5").Value);
33         addressAttributes.Add("DelPostcode", delAddressNode.Attribute("delpostcode").Value);
34         addressAttributes.Add("DelCountryCode", delAddressNode.Attribute("delcountrycode").Value);
35     }
36     return addressAttributes;
37 }

```

Scenario: "Show the total number of items on pc_ordercode CUSTCODE20240508115533" [🔗](#)

C#:

```

1 private static void Main()
2 {
3     var options = new RestClientOptions("https://sandbox.pencarrie.com");
4     using var client = new RestClient(options);
5     var request = new RestRequest("/gateway", Method.Post)
6         .AddParameter("function", "pcget")
7         .AddParameter("code", "CUSTCODE");
8     .AddParameter("ordcode", "CUSTCODE20240508115533");
9     RestResponse response = client.Execute(request);
10
11     XDocument doc = XDocument.Parse(response.Content);
12     var items = doc.Descendants("items").Elements("item");
13     int totalQty = items.Sum(item => int.Parse(item.Attribute("qty").Value));
14
15     Console.WriteLine("Total quantity: {0}", totalQty);
16 }

```

pclist [🔗](#)

Returns the header information of all orders for your customer code. A full field list is provided in [Appendix C](#).

Parameters [🔗](#)

Field	Value	Description
-------	-------	-------------

function	pclist	function name
code	[your customer code]	e.g. "ABCD"

Response [↗](#)

XML representations of the PenCarrie order header objects stored on the system for your customer code

```

1 <orders>
2   <order currency="GBP" net="121.5" vat="24.3" duty="0.00" duty_invoiced="0" additional_fees="0.00"
  custref="order 1" ordcode="CUSTCODE20240508115533" ordstat="Pending" xx_shipdate="20240508000000"
  orderdate="20240712094452" pflag="False" xflag="False" dflag="False" cflag="False" status="P" user="Test User"
  ordno="6346656" del_tid="100" trackntrace="Your order has not yet been shipped." delcarrier="ALT"
  delcompany="Pencarrie Tests Accounts" deladd1="OWN CARRIER" deladd2="PenCarrie Limited" deladd3="South View
  Estate" deladd4="Willand Town" deladd5="Cullompton: devon" delpostcode="EX15 2QW" delcountrycode="GB"
  deladdcode="CUSTCODE20150311130022" code="CUSTCODE" voucherid="0" voucher="-" vouchervalue="0"
  created="20240508115533" review="0" line_count="1" canedit="1" cancel="0" despatchnoteavailable="0"
  orderacknowledgementavailable="0" proformaavailable="0" recalcrequired="0" lockedby="-" ordtype="P"/>
3   <order currency="GBP" net="436.5" vat="84.55" duty="0.00" duty_invoiced="0" additional_fees="0.00"
  custref="order 2" ordcode="CUSTCODE20240909144930" ordstat="Picking" xx_shipdate="20240909000000"
  orderdate="20240909144936" pflag="True" xflag="False" dflag="False" cflag="True" status="C" user="WEB"
  ordno="6539941" del_tid="91" trackntrace="Your order has not yet been shipped." delcarrier="PL"
  delcompany="Pencarrie Tests Accounts" deladd1="PenCarrie Limited" deladd2="South View Estate" deladd3="Willand
  Town" deladd4="Cullompton" deladd5="devon" delpostcode="EX15 2QW" delcountrycode="GB"
  deladdcode="CUSTCODE20150311130022" code="CUSTCODE" voucherid="0" voucher="-" vouchervalue="0"
  created="20240909144930" review="0" line_count="5" canedit="0" cancel="0" despatchnoteavailable="0"
  orderacknowledgementavailable="0" proformaavailable="0" recalcrequired="0" lockedby="-" ordtype="P"
  minimum_delivery_date="2024-09-10 07:30:00" maximum_delivery_date="2024-09-11 19:30:00"/>
4   <order currency="GBP" net="276.6" vat="55.32" duty="0.00" duty_invoiced="0" additional_fees="0.00"
  custref="order 3" ordcode="CUSTCODE20240909152255" ordstat="Picking" xx_shipdate="20240909000000"
  orderdate="20240909152355" pflag="True" xflag="False" dflag="False" cflag="True" status="C" user="Test User"
  ordno="6539943" del_tid="91" trackntrace="Your order has not yet been shipped." delcarrier="PL"
  delcompany="Pencarrie Tests Accounts" deladd1="PenCarrie Limited" deladd2="South View Estate" deladd3="Willand
  Town" deladd4="Cullompton" deladd5="devon" delpostcode="EX15 2QW" delcountrycode="GB"
  deladdcode="CUSTCODE20150311130022" code="CUSTCODE" voucherid="0" voucher="-" vouchervalue="0"
  created="20240909152254" review="0" line_count="4" canedit="0" cancel="0" despatchnoteavailable="0"
  orderacknowledgementavailable="0" proformaavailable="0" recalcrequired="0" lockedby="-" ordtype="W"
  minimum_delivery_date="2024-09-10 07:30:00" maximum_delivery_date="2024-09-11 19:30:00"/>
5 </orders>

```

Code samples [↗](#)

Scenario: *"I need a full list of my current orders"* [↗](#)

cURL:

```
1 curl -X POST -F function=pclist -F code=CUSTCODE -L https://sandbox.pencarrie.com/gateway
```

Python:

```

1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pclist", "code": "CUSTCODE"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)

```

PHP:

```
1 <?php
```

```

2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pclist", "code" => "CUSTCODE");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 ));
11 echo $response;
12 ?>

```

C#:

```

1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pclist")
5     .AddParameter("code", "CUSTCODE");
6 RestResponse response = client.Execute(request);
7 Console.WriteLine(response.Content);
8 Console.ReadLine();

```

Scenario: “I want to list tracking details for my orders” [🔗](#)

C#:

```

1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pclist")
5     .AddParameter("code", "CUSTCODE");
6 RestResponse response = client.Execute(request);
7 Console.WriteLine(response.Content);
8
9 XDocument xDoc = XDocument.Parse(response.Content);
10
11 var orderList = xDoc.Descendants("order")
12     .Select(order => new
13     {
14         OrdNo = order.Attribute("ordno").Value,
15         TrackNTrace = order.Attribute("trackntrace").Value
16     })
17     .ToList();
18
19 foreach (var order in orderList)
20 {
21     Console.WriteLine($"OrdNo: {order.OrdNo}, TrackNTrace: {order.TrackNTrace}");
22 }

```

pcautoorder [🔗](#)

Generates a new order based on XML representation of an order passed to the function. See [Appendix A](#) for a complete example / definition of a PC Auto Order XML order argument.

Parameters [🔗](#)

Field	Value	Description
-------	-------	-------------

function	pclist	function name
code	[your customer code]	e.g. "ABCD"
xml	[order details XML]	XML-format order information

xml input example [🔗](#)

```
1 <orders dt='2024080509580211' code='CUSTCODE'>
2   <order>
3     <header code='CUSTCODE' ref='Order123' parkorder='1' deladdcode='IG100115115123' del_tid='201'
4     assumebo='0'
5     updateexistingorder='0' placedby='test' contact='test' sn_email='test@test.com' oaemail='0'>
6     </header>
7     <email sn_email='a@b.com' contact='tc' oaemail='0' />
8     <line prodcode='JC001 ACW L' qty='6' ref='36684' />
9     <line prodcode='JC001 ACW M' qty='1' ref='36685' />
10    <line prodcode='JC001 ACW S' qty='7' ref='36686' />
11  </order>
12</orders>
```

Element	Information
orders	root element should contain 2 attributes, - code your customer code - dt an alphanumeric key that ensures idempotent execution of the order request. A datetime string is normally sufficient, typically in the format "YYYYMMDDHHmmSS"
order	encapsulates a single order. multiple order elements can be submitted within orders
header	top-level order information such as delivery address (or deladdcode for pre-registered addresses), delivery option, customer reference, backorder options, contact information
email	additional contacts to receive order confirmation. one element per contact.
line	one for each submitted product, as a minimum must include prodcode and qty

Response [🔗](#)

An XML response from the PenCarrie web service detailing either the order header information, lines and an indication of whether each line was accepted by the PenCarrie service. Alternatively, an error node will be returned which will indicate why the order was rejected.

```
1 <orders dt='2024092415245223' code='CUSTCODE'>
2   <order>
3     <header code='CUSTCODE' ref='Order123' parkorder='2' plaincover='-1' add1='fore street'
4     add3='cullompton' add4='Devon' add5='UK - Mainland' postcode='EX15 2QW' country='GB' del_tid='91'
5     delcompany='test company' assumebo='1' updateexistingorder='0' placedby='test user' mobile='07789123456'
6     sn_email='testuser@test.com' oaemail='1' />
7     <email sn_email='testuser@test.com' contact='test' oaemail='1' />
8     <line prodcode='JC001 ACW L' qty='6' ref='JC001 ACW L' conf='- ' quantity='6' orderquan='6' backord='0'
9     skucodein='JC001 ACW L' matched='JC001 ACW L' status='Submitted' />
10    <line prodcode='JC001 ACW M' qty='1' ref='JC001 ACW M' conf='- ' quantity='6' orderquan='6' backord='0'
11    skucodein='JC001 ACW M' matched='JC001 ACW M' status='Submitted' />
12    <line prodcode='JC001 ACW S' qty='7' ref='JC001 ACW S' conf='- ' quantity='6' orderquan='6' backord='0'
13    skucodein='JC001 ACW S' matched='JC001 ACW S' status='Submitted' />
14    <updateresult>
```

```

9      <failure redbox='PCRef' /> <!-- deprecated -->
10     <ok despcount='0' /> <!-- deprecated -->
11   </updateresult>
12   <linesresult>
13     <allocations pc_ordercode='CUSTCODE_Order123' custorderno='Order123' code='CUSTCODE'>
14       <allocate conf='true' prodcode='JC001 ACW L' quantity='6' orderquan='6' backord='0' fts='303'
picked='0' match='JC001 ACW L' mystat='- ' />
15       <allocate conf='true' prodcode='JC001 ACW M' quantity='1' orderquan='1' backord='0' fts='99'
picked='0' match='JC001 ACW M' mystat='- ' />
16       <allocate conf='true' prodcode='JC001 ACW S' quantity='7' orderquan='7' backord='0' fts='54'
picked='0' match='JC001 ACW S' mystat='- ' />
17     </allocations>
18   </linesresult>
19 </order>
20 </orders>

```

Element	Information
header	top-level order information, e.g. delivery address, contact details
email	additional contacts to receive order confirmation
line	one for each submitted product
updateresult	deprecated
linesresult	a collection of recognised submitted lines, and a true/false processing result for each (attribute: <code>conf</code>)

Code samples [↗](#)

Scenario: “Submit an order for 50 x JH001 Hoodies to be delivered to my main business address using my default delivery option. Automatically create a backorder if they're not all in stock” [↗](#)

XML Argument:

```

1 <orders dt='2024080509580211' code='CUSTCODE'>
2   <order>
3     <header code='CUSTCODE' ref='00081' parkorder='2' deladdcode='ABC123456789' assumebo='1'
placedby='test' contact='test' sn_email='test@test.com' oaemail='0'>
4     </header>
5     <email sn_email='a@b.com' contact='tc' oaemail='0' />
6     <line prodcode='JH001 BLK M' qty='50' ref='hoodiestock' />
7   </order>
8 </orders>

```

note `parkorder=2` to complete the order regardless of backorder status, and `assumebo=1` which automatically creates a backorder for out of stock or partially fulfilled lines

cURL:

```

1 curl -X POST -F function=pcautoorder -F code=CUSTCODE -F xml="<orders dt=\"2024080509580211\"
code=\"CUSTCODE\"><order><header code=\"CUSTCODE\" ref=\"00081\" parkorder=\"2\" deladdcode=\"ABC123456789\"
assumebo=\"1\" placedby=\"test\" contact=\"test\" sn_email=\"test@test.com\" oaemail=\"0\"></header><email
sn_email=\"a@b.com\" contact=\"tc\" oaemail=\"0\" /><line prodcode=\"JH001 BLK M\" qty=\"50\"
ref=\"hoodiestock\" /></order></orders>" -L https://sandbox.pencarrie.com/gateway

```

Python:

```

1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"

```

```

3 data = {"function": "pcautoorder", "code": "CUSTCODE", "xml": "<orders dt='2024080509580211' code='CUSTCODE'>
  <order><header code='CUSTCODE' ref='00081' parkorder='2' deladdcode='ABC123456789' assumebo='1' placedby='test'
  contact='test' sn_email='test@test.com' oaemail='0'></header><email sn_email='a@b.com' contact='tc'
  oaemail='0'></email><line prodcode='JH001 BLK M' qty='50' ref='hoodiestock'></line></order></orders>"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)

```

PHP:

```

1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pcautoorder", "code" => "CUSTCODE", "xml" => "<orders dt='2024080509580211'
  code='CUSTCODE'><order><header code='CUSTCODE' ref='00081' parkorder='2' deladdcode='ABC123456789'
  assumebo='1' placedby='test' contact='test' sn_email='test@test.com' oaemail='0'></header><email
  sn_email='a@b.com' contact='tc' oaemail='0'></email><line prodcode='JH001 BLK M' qty='50' ref='hoodiestock'></line></order>
  </orders>");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 ));
11 echo $response;
12 ?>

```

C#:

```

1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcautoorder")
5     .AddParameter("code", "CUSTCODE")
6     .AddParameter("xml", "<orders dt='2024080509580211' code='CUSTCODE'><order><header code='CUSTCODE'
  ref='00081' parkorder='2' deladdcode='ABC123456789' assumebo='1' placedby='test' contact='test'
  sn_email='test@test.com' oaemail='0'></header><email sn_email='a@b.com' contact='tc' oaemail='0'></email><line
  prodcode='JH001 BLK M' qty='50' ref='hoodiestock'></line></order></orders>");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();

```

Scenario: “Order for 7 x BB10 Beanies delivered to a 3rd party address. Send the order on a 48-hour delivery under under plain cover, and do not automatically create backorders” 

XML Argument:

```

1 <orders dt='2024080509580211' code='CUSTCODE'>
2     <order>
3         <header code='CUSTCODE' ref='00081' parkorder='2' plaincover='-1' del_tid='601' delcompany="Test
  Company" add1="Unit 123" add2="high road" add3="cullompton" add4="Devon" add5="" postcode="EX15 2QW"
  country="GB" assumebo='0' placedby='test' contact='test' sn_email='test@test.com' oaemail='0'>
4         <email sn_email='a@b.com' contact='test' oaemail='0'>
5         <line prodcode='BB10 BLK ONE' qty='7'>
6     </order>
7 </orders>

```

note `parkorder=2` to complete the order regardless of backorder status, and `assumebo=0` which will skip any lines that cannot be fulfilled. `del_tid=601` defines a 48-hour delivery (see [Appendix D](#) for further options). `plaincover='-1'` sends the order under standard plain cover, with PenCarrie information/logos removed.

cURL:

```
1 curl -X POST -F function=pcautoorder -F code=CUSTCODE -F xml="<orders dt='2024080509580211'
code='CUSTCODE'><order><header code='CUSTCODE' ref='00081' parkorder='2' plaincover='-1'
del_tid='601' delcompany='Test Company' add1='Unit 123' add2='high road' add3='cullompton' add4='Devon'
add5='' postcode='EX15 2QW' country='GB' assumebo='0' placedby='test' contact='test'
sn_email='test@test.com' oaemail='0'><email sn_email='a@b.com' contact='test' oaemail='0'><line
prodcode='BB10 BLK ONE' qty='7'></order></orders>" -L https://sandbox.pencarrie.com/gateway
```

Python:

```
1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcautoorder", "code": "CUSTCODE", "xml": "<orders dt='2024080509580211' code='CUSTCODE'>
<order><header code='CUSTCODE' ref='00081' parkorder='2' plaincover='-1' del_tid='601' delcompany='Test
Company' add1='Unit 123' add2='high road' add3='cullompton' add4='Devon' add5='' postcode='EX15 2QW'
country='GB' assumebo='0' placedby='test' contact='test' sn_email='test@test.com' oaemail='0'><email
sn_email='a@b.com' contact='test' oaemail='0'><line prodcode='BB10 BLK ONE' qty='7'></order></orders>"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)
```

PHP:

```
1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pcautoorder", "code" => "CUSTCODE", "xml" => "<orders dt='2024080509580211'
code='CUSTCODE'><order><header code='CUSTCODE' ref='00081' parkorder='2' plaincover='-1' del_tid='601'
delcompany='Test Company' add1='Unit 123' add2='high road' add3='cullompton' add4='Devon' add5=''
postcode='EX15 2QW' country='GB' assumebo='0' placedby='test' contact='test' sn_email='test@test.com'
oaemail='0'><email sn_email='a@b.com' contact='test' oaemail='0'><line prodcode='BB10 BLK ONE' qty='7'>
</order></orders>");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 ));
11 echo $response;
12 ?>
```

C#:

```
1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcautoorder")
5     .AddParameter("code", "CUSTCODE")
6     .AddParameter("xml", "<orders dt='2024080509580211' code='CUSTCODE'><order><header code='CUSTCODE'
ref='00081' parkorder='2' plaincover='-1' del_tid='601' delcompany='Test Company' add1='Unit 123' add2='high
road' add3='cullompton' add4='Devon' add5='' postcode='EX15 2QW' country='GB' assumebo='0' placedby='test'
contact='test' sn_email='test@test.com' oaemail='0'><email sn_email='a@b.com' contact='test' oaemail='0'>
<line prodcode='BB10 BLK ONE' qty='7'></order></orders>");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();
```

pcupdate_new

Creates a new empty, open order using the account default information for new orders on your account.

Parameters

Field	Value	Description
function	pcupdate_new	function name
code	[your customer code]	e.g. "ABCD"
xml	[order details XML]	XML-format order header information

xml input example

```
1 <new code="CUSTCODE" custorderno="testref" pceref="test user" />
```

Element	Information
new	root element should contain 3 attributes, 1. <code>code</code> your customer code 2. <code>custorderno</code> your own alphanumeric order reference 3. <code>pceref</code> who placed the order, typically a name or department or process, e.g. "Joe Bloggs", "accounts", "autoorder" etc.

Response

XML confirmation of order creation

```
1 <ok pc_ordercode='W_CUSTCODE_2024101011503018' ordno='123456789' />
```

Element	Information
pc_ordercode	PenCarrie unique order reference. This is used for most order-based API functions
ordno	PenCarrie order number

Code samples

Scenario: "Create a new blank order with a reference I'll use for my records"

cURL:

```
1 curl -X POST -F function=pcupdate_new -F code=CUSTCODE -F xml="<new code=\"CUSTCODE\" custorderno=\"testref\" pceref=\"test user\" />" -L https://sandbox.pencarrie.com/gateway
```

Python:

```
1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcupdate_new", "code": "CUSTCODE", "xml": "<new code='CUSTCODE' custorderno='testref' pceref='test user' />"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)
```

PHP:

```
1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pcupdate_new", "code" => "CUSTCODE", "xml" => "<new code='CUSTCODE'
  custorderno='testref' pceref='test user' />");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 ));
11 echo $response;
12 ?>
```

C#:

```
1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcupdate_new")
5     .AddParameter("code", "CUSTCODE")
6     .AddParameter("xml", "<new code='CUSTCODE' custorderno='testref' pceref='test user' />");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();
```

pcupdate [🔗](#)

Adds/changes the header data of editable orders. Also used to complete orders by changing their status to 'F' - including both orders created through `pcupdate_new`, and `pcautoorder` where the order was **not** completed at point of creation.

Parameters [🔗](#)

Field	Value	Description
function	pcupdate	function name
code	[your customer code]	e.g. "ABCD"
xml	[order details XML]	XML-format order header information. attribute "pc_ordercode" is required.

xml input example [🔗](#)

```
1 <order pc_ordercode="W_CUSTCODE_2024101011503018" /*[attributes to update]*/ />
```

Element	Information
order	root element must begin with the <code>pc_ordercode</code> attribute. Following this will be one or more attributes defining header-level order details to change. A full list of attributes that can be changed through <code>pcupdate</code> is available in Appendix B .

Response [↗](#)

XML node confirming successful update

```
1 <ok despcount='0' />
```

Node	Information
ok	presence of the element confirms successful update
despcount	indicates the number of boxes that have already been shipped. If this is greater than zero, the header of the order cannot be edited although its status can still be changed. If the pcupdate call that produced this response included parameters other than pc_ordercode and status, these will have been ignored. In this situation any subsequent call to pallocate will not allow product quantities to be reduced below the levels already shipped. However, further products may be added freely.

Code samples [↗](#)

Samples assume you have an order in an editable state with a pc_ordercode of "W_CUSTCODE_2024101011503018". This order will have been created with pcupdate_new and items added with pallocate. Note the scenarios are purposely separated for reasons of clarity, however pcupdate allows the update of multiple order header data at once and it may be more efficient to request the various updates in a single request.

Scenario: "I've created an order using pcupdate_new, added some items using pallocate, now set to Next Day delivery (see [Appendix E](#) for delivery options)" [↗](#)

cURL:

```
1 curl -X POST -F function=pcupdate -F code=CUSTCODE -F xml="<order op=\"web\"  
pc_ordercode=\"W_CUSTCODE_2024101011503018\" del_tid=\"92\" />" -L https://sandbox.pencarrie.com/gateway
```

Python:

```
1 import requests  
2 url = "https://sandbox.pencarrie.com/gateway"  
3 data = {"function": "pcupdate", "code": "CUSTCODE", "xml": "<order op='web'  
pc_ordercode='W_CUSTCODE_2024101011503018' del_tid='92' />"}  
4 response = requests.post(url, data=data, allow_redirects=True)  
5 print(response.text)
```

PHP:

```
1 <?php  
2 $url = "https://sandbox.pencarrie.com/gateway";  
3 $data = array("function" => "pcupdate", "code" => "CUSTCODE", "xml" => "<order op='web'  
pc_ordercode='W_CUSTCODE_2024101011503018' del_tid='92' />");  
4 $response = file_get_contents($url, false, stream_context_create(array(  
5     'http' => array(  
6         'method' => 'POST',  
7         'header' => 'Content-type: application/x-www-form-urlencoded',  
8         'content' => http_build_query($data)  
9     )  
10 ));  
11 echo $response;  
12 ?>
```

C#:

```

1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcupdate")
5     .AddParameter("code", "CUSTCODE")
6     .AddParameter("xml", "<order op='web' pc_ordercode='W_CUSTCODE_2024101011503018' del_tid='92' />");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();

```

Scenario: *"I've made all the updates I need, now to complete my order"* [🔗](#)

cURL:

```

1 curl -X POST -F function=pcupdate -F code=CUSTCODE -F xml="<order op='web\'
pc_ordercode='W_CUSTCODE_2024101011503018' status='F' />" -L https://sandbox.pencarrie.com/gateway

```

Python:

```

1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcupdate", "code": "CUSTCODE", "xml": "<order op='web\'
pc_ordercode='W_CUSTCODE_2024101011503018' status='F' />"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)

```

PHP:

```

1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pcupdate", "code" => "CUSTCODE", "xml" => "<order op='web\'
pc_ordercode='W_CUSTCODE_2024101011503018' status='F' />");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 ));
11 echo $response;
12 ?>

```

C#:

```

1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcupdate")
5     .AddParameter("code", "CUSTCODE")
6     .AddParameter("xml", "<order op='web' pc_ordercode='W_CUSTCODE_2024101011503018' status='F' />");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();

```

pcallocate [🔗](#)

Allocates and deallocates stock to a valid order. In order to add/remove stock from an order the order must be in an editable state, i.e. a status of either **OPEN** or **PARKED**.

Parameters [↗](#)

Field	Value	Description
function	pcallocate	function name
code	[your customer code]	e.g. "ABCD"
xml	[Allocations XML]	XML-format product allocation information. attribute "pc_ordercode" is required.

xml input example [↗](#)

```
1 <allocations dt="20241025154000" code="CUSTCODE" pc_ordercode="W_CUSTCODE_2024101011503018">
2   <allocate conf="-" prodcode="SS11 BLK M" quantity="10" backord="0" orderquan="10" cref="testref" />
3   <allocate conf="-" prodcode="BB10 BLK ONE" quantity="5" backord="0" orderquan="5" cref="testref" />
4 </allocations>
```

Element	Information
order	root element must include attributes <code>code</code> , <code>dt</code> , and <code>pc_ordercode</code> . nested under the root element will be one or more <code>allocate</code> elements. each of these will detail • (mandatory) <code>conf</code> attribute with a value of "-". In the xml response this attribute value will return a Boolean indicating success or fail allocation of this line. • (mandatory) a product SKU <code>prodcode</code> • (mandatory) a delivery quantity <code>quantity</code> • (mandatory) a backorder quantity <code>backord</code> • (mandatory) a total quantity <code>orderquan</code> this must be equal to the sum of <code>quantity</code> and <code>backord</code> • (optional) a line reference <code>cref</code> . this will appear next to the product line on despatch/invoice document, and can be used for your internal processes. • IMPORTANT: garment values sent using <code>pcallocate</code> are additive, in that they increment the total quantity of an item by the submitted amount. For example, in the above xml input example there is a submitted value of 10 x "SS11 BLK M". This will change the number of "SS11 BLK M" from zero to 10. To then change the value to 13, the next <code>pcallocate</code> submission would have a quantity of 3, changing the total from 10 to 13. Similarly to remove items, or 'delete' a line entirely, you would submit a <code>pcallocate</code> request for a negative amount. To continue the example, to remove all "SS11 BLK M" from the order you would submit -13.

Response [↗](#)

XML node confirming allocations processed

```
1 <allocations code='CUSTCODE' dt='20241025154000' pc_ordercode='W_CUSTCODE_2024101011503018'>
2   <allocate conf='true' cref='testref' prodcode='SS11 BLK M' orderquan='10' quantity='10' backord='0' fts='105'
picked='0' match='SS11 BLK M' mystat='- ' />
3   <allocate conf='false' cref='testref' prodcode='BB10 BLK ONE' orderquan='5' quantity='4' backord='1' fts='4'
picked='0' match='BB10 BLK 0' mystat='- ' />
4 </allocations>
```

Element	Information
allocations	header element echoing submitted order data

allocate (one or more)	This function will only post allocation lines that are a 'perfect fit': where the <code>quantity</code> value can actually be allocated from available stock. If an allocation request is made with a <code>quantity</code> value exceeding available stock, the system will return a modified allocation XML document showing how much can be allocated from stock as a <code>quantity</code> value with any remainder shown as a <code>backord</code> value. In this example, 10 x "SS11 BLK M" were requested and, since there was stock, successfully allocated, as denoted by the attribute <code>conf='true'</code>. However, there was not enough stock to satisfy the request of 5 x "BB10 BLK ONE", thus the line was not processed, as denoted by the attribute <code>conf='false'</code>. The response further indicates you <i>could</i> submit a subsequent pallocate request for the 5 desired items, placing 4 on the order immediately and registering the remaining 1 for backorder. Note the item response lines include a Free To Sell <code>fts</code> attribute indicating current stock.
------------------------	--

Code samples [↗](#)

Scenario: "Add 20 x SS11 BLK M to my order" [↗](#)

cURL:

```
1 curl -X POST -F function=pcallocate -F code=CUSTCODE -F xml="<allocations dt=\"20241025154000\"
  code=\"CUSTCODE\" pc_ordercode=\"W_CUSTCODE_20241011503018\"><allocate conf=\"-\" prodcode=\"SS11 BLK M\"
  quantity=\"20\" backord=\"0\" orderquan=\"20\" cref=\"testref\" /></allocations>" -L
  https://sandbox.pencarrie.com/gateway
```

Python:

```
1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcallocate", "code": "CUSTCODE", "xml": "<allocations dt='20241025154000' code='CUSTCODE'
  pc_ordercode='W_CUSTCODE_20241011503018'><allocate conf='-' prodcode='SS11 BLK M' quantity='20' backord='0'
  orderquan='20' cref='testref' /></allocations>"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)
```

PHP:

```
1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pcallocate", "code" => "CUSTCODE", "xml" => "<allocations dt='20241025154000'
  code='CUSTCODE' pc_ordercode='W_CUSTCODE_20241011503018'><allocate conf='-' prodcode='SS11 BLK M'
  quantity='20' backord='0' orderquan='20' cref='testref' /></allocations>");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 ));
11 echo $response;
12 ?>
```

C#:

```
1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4     .AddParameter("function", "pcallocate")
```

```

5 .AddParameter("code", "CUSTCODE")
6 .AddParameter("xml", "<allocations dt='20241025154000' code='CUSTCODE'
pc_ordercode='W_CUSTCODE_2024101011503018'><allocate conf='- ' prodcode='SS11 BLK M' quantity='20' backord='0'
orderquan='20' cref='testref' /></allocations>");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();

```

Scenario: *"I have 20 x SS11 BLK M on my order, but now only need 14. Amend the amount"* [🔗](#)

cURL:

```

1 curl -X POST -F function=pcallocate -F code=CUSTCODE -F xml="<allocations dt=\"20241025154000\"
code=\"CUSTCODE\" pc_ordercode=\"W_CUSTCODE_2024101011503018\"><allocate conf=\"- \" prodcode=\"SS11 BLK M\"
quantity=\"-6\" backord=\"0\" orderquan=\"-6\" /></allocations>" -L https://sandbox.pencarrie.com/gateway

```

Python:

```

1 import requests
2 url = "https://sandbox.pencarrie.com/gateway"
3 data = {"function": "pcallocate", "code": "CUSTCODE", "xml": "<allocations dt='20241025154000' code='CUSTCODE'
pc_ordercode='W_CUSTCODE_2024101011503018'><allocate conf='- ' prodcode='SS11 BLK M' quantity='-6' backord='0'
orderquan='-6' /></allocations>"}
4 response = requests.post(url, data=data, allow_redirects=True)
5 print(response.text)

```

PHP:

```

1 <?php
2 $url = "https://sandbox.pencarrie.com/gateway";
3 $data = array("function" => "pcallocate", "code" => "CUSTCODE", "xml" => "<allocations dt='20241025154000'
code='CUSTCODE' pc_ordercode='W_CUSTCODE_2024101011503018'><allocate conf='- ' prodcode='SS11 BLK M'
quantity='-6' backord='0' orderquan='-6' /></allocations>");
4 $response = file_get_contents($url, false, stream_context_create(array(
5     'http' => array(
6         'method' => 'POST',
7         'header' => 'Content-type: application/x-www-form-urlencoded',
8         'content' => http_build_query($data)
9     )
10 ));
11 echo $response;
12 ?>

```

C#:

```

1 var options = new RestClientOptions("https://sandbox.pencarrie.com");
2 using var client = new RestClient(options);
3 var request = new RestRequest("/gateway", Method.Post)
4 .AddParameter("function", "pcallocate")
5 .AddParameter("code", "CUSTCODE")
6 .AddParameter("xml", "<allocations dt='20241025154000' code='CUSTCODE'
pc_ordercode='W_CUSTCODE_2024101011503018'><allocate conf='- ' prodcode='SS11 BLK M' quantity='-6' backord='0'
orderquan='-6' /></allocations>");
7 RestResponse response = client.Execute(request);
8 Console.WriteLine(response.Content);
9 Console.ReadLine();

```

Appendices [🔗](#)

Appendix A: pcautoorder XML [🔗](#)

NODE	Attribute	Value Type	Data Type / Size	Description
<orders>				
	dt	TEXT	16	Unique Submission ID Used to determine if a given XML submission has been received before.
	code	TEXT	5	Concatenated with Dt to ensure each xml submission can be identified uniquely across all accounts.
<order>				
<header>				
	ref	TEXT	14	an alphanumeric PO identifier (underscore “_” characters are permitted) MUST BE UNIQUE
	code	TEXT	5	Your account code
	deladdcode	TEXT	20	CONDITIONAL – replaces address attributes Please contact us if you require a list of your stored address codes
	delcompany	TEXT	50	(Optional) Name of the company or person taking delivery of the order. This will default to your company name if not specified.
	add1	TEXT	50	Address Line Data
	add2	TEXT	50	Address Line Data
	add3	TEXT	50	Address Line Data
	add4	TEXT	50	Address Line Data
	add5	TEXT	50	Address Line Data
	postcode	TEXT	15	Full postcode
	country	TEXT	3	Country code
	placedby	TEXT	40	Person placing the order
	dte	Date	dd/mm/yyyy	
	due	Date	dd/mm/yyyy	Obsolete – Please use <i>shipdate</i> instead
	shipdate	Date	dd/mm/yyyy	The date on which you would like us to despatch your order. If a 48-hour delivery service is selected we reserve the right to

				despatch your order a day later on a next-day service at no extra charge. Please note that if you specify a date that PenCarrie is not open, your order will be shipped on the next available date.
	plaincover	Number	Signed Integer	Plain Carriage Option Identifier
	invpicktext	TEXT	255	Instructions for the Picking Team
	Invcomment	TEXT	255	Free text; can be used to provide instructions to the customer services team and will be printed on the despatch note unless otherwise instructed.
	del_tid	Number	Unsigned Integer	Carriage Option; See Appendix D for available values
	parkorder	Number	0-2	1 = After submission, leave this order in a state that allows additions 0 = Process this order through Credit Control for picking (will not process if the order contains any items that have been allocated as backorders) 2 = Process this order through Credit Control for picking (will process irrespective of item backorder status) (If omitted the system will assume 1)
	assumebo	number	1-0	1 = Create back orders for any stock shortages 0 = Skip any lines that cannot be completely satisfied from available stock
	mobile	TEXT	30	Mobile telephone number for DPD to send parcel tracking alert texts to. Only compatible with DPD delivery services, if other services are selected no alerts will be sent.
	tracking_email_address	Text	255	Deliveries outside of the UK and Republic of Ireland Only. Specifies an alternative email address to provide to DPD for delivery tracking emails. Your account's default email address will be used if this is not specified, is not a valid email address, or your order ships within the UK or Republic of Ireland.
	atrisk	number	1-0	(Windsor Framework - required for NI only)

				1=true 0=false. Determines if goods moving to NI are at risk of moving onwards further to the EU, more info here: EU - GOV.UK .
	b2c	number	1-0	(Windsor Framework - required for NI only) 1=true 0=false. If the goods are considered as delivered to a consumer address, rather than a business address.
	thirdparty	number	1-0	(Windsor Framework - required for NI only) 1=true 0=false. False if delivery is to your registered business address, true otherwise.
				Additional notes on NI - Windsor Framework • If the order is shipping to a third party and that is a consumer and not a business, it must declare <code>thirdparty =1</code>, <code>b2c =1</code>, <code>atrisk =0</code> • If shipping to a third party that is a business, it must declare <code>thirdparty =1</code> and <code>b2c =0</code>, and <code>atrisk</code> as appropriate • If not shipping to a third party, must declare <code>thirdparty =0</code> and <code>b2c =0</code>, and <code>atrisk</code> as appropriate • If your UKIMS number is not registered with PenCarrie, must declare <code>atrisk =1</code> if <code>b2c =0</code>
</header>				
<email>				
	sn_email	TEXT	100	Email address for notification purposes
	contact	TEXT	50	Contact name
	oaemail	Number	Byte	Flag to indicate that an order acknowledgement should be sent to this address.
</email>				
<line>				
	prodcode	TEXT	32	PENCARRIE product code (MUST BE PENCARRIE'S SKU CODE)
	qty	Number	Integer	TOTAL requirement
	ref	TEXT	100	Your reference for this line
</line>				
<exportdetails> >				Non-GB orders or shipping to Northern Ireland

				Child node of <i>order</i> specifying additional contact information required for orders leaving the United Kingdom*
	receivername	Text		Required - The name of a person at the delivery address who can be contacted by DPD or Customs if there are any problems with the order
	receiverphone	Text		Conditional** - A telephone number to contact the above
	receiveremailaddresses	Text		Conditional** - An email address to contact the above
</exportdetails>				
<order />				
<orders />				

* If you are shipping to a third-party address, these need to be contact details for someone at that address. If we are shipping directly to your customer, please ensure that you provide their details so the courier or customs authorities can contact them if there are any problems with their delivery. Please see [Appendix E](#) for further information.

** At least one of receiverphone or receiveremailaddress must be provided if export details are required on the order.

Appendix B: pcupdate XML [🔗](#)

NODE	Attribute	Value Type	Description
<order>			
	pc_ordercode	Text	REQUIRED The unique machine identifier for your order: this can be obtained: - Through a call to <i>PCList</i> - In the response to <i>PCUpdate_New</i> - If the order was created through <i>PCAutoOrder</i>, by concatenating the customer code, underscore and the order reference (i.e. [code]_[ref])
	status	Text	Will attempt to change the order to the specified status, if that change is valid. See above for a list of valid status changes. This should only be included if the intention of the PCUpdate call is to change the status of the order.

	<code>review</code>	Integer	Set review='0' before completing an order if it was originally raised via pcautoorder and has a review flag against it important: <code>review</code> and <code>status</code> should not be specified in the same call. An attempt to set <code>status</code> to F will fail if the order's <code>review</code> field is nonzero and has not been cleared by setting review to 0 through this call (i.e. this must be done first)
	<code>custorderno</code>	Text	Sets the customer reference/PO number for the order
	<code>pcref</code>	Text	The name of the person responsible for the order from the customer's point of view (i.e. who Customer Services will ask for if there is a problem with the order)
	<code>pcrefemail</code>	Text	Contact email address for the person named in pcref
	<code>deladdcode</code>	Text	Unique identifier for your stored address. A full list of available address codes can be provided upon request. Alternatively a custom address can be specified using the parameters below. If a <code>deladdcode</code> and an address are specified, the <code>deladdcode</code> will overwrite the address. If neither is specified, the address will not be changed.
	<code>delcompany</code>	Text	Specifies a custom delivery address. <i>Delcompany</i> sets the receiver name. Note that some address labels may not display <i>deladd5</i> .
	<code>deladd1</code>		If shipping to the UK, <i>delpostcode</i> must contain a space between the first and second parts of the postcode.
	<code>deladd2</code>		
	<code>deladd3</code>		For deliveries to Great Britain and Northern Ireland, please use <i>GB</i> as the country code. For the Republic of Ireland, please use <i>EI</i> .
	<code>deladd4</code>		
	<code>deladd5</code>		
	<code>delpostcode</code>		
	<code>delcountrycode</code>		
	<code>del_tid</code>	Number	Delivery code for required service; if not specified the default service for your stored address will be used if a stored address is specified; otherwise the default service for the selected postcode will be used (usually next day). Please see Appendix D for a list of valid codes.
	<code>shipdate</code>	Date	Date, in the format YYYYMMDD000000 (e.g. 20241127000000 for 27 November 2024), upon which you would like the order to leave PenCarrie. If a 48-hour delivery service is selected we reserve the right to despatch your order a day later on a next-day service at no extra charge. Please note that if you specify a date that PenCarrie is not open, your order will be shipped on the next available date.
	<code>txtalert</code>	Boolean	True or False; indicates whether you would like to receive delivery text alerts from DPD (DPD delivery services only)
	<code>txtnumber</code>	Text	Mobile phone number to receive text alerts if txtalert is set to True. Required if txtalert is true but can be set in separate calls

	tracking_email_address	Text	Deliveries outside of the UK and Republic of Ireland Only. Specifies an alternative email address to provide to DPD for delivery tracking emails. Your account's default email address will be used if this is not specified, is not a valid email address, or your order ships within the UK or Republic of Ireland. Once a tracking email address has been set on an order it will be remembered for that order unless a subsequent <i>pcupdate</i> call provides a different email address. To remove the email address and return to the default behaviour, pass an empty string into this field.
	atrisk	number 1-0	(Windsor Framework - required for NI only) 1=true 0=false. Determines if goods moving to NI are at risk of moving onwards further to the EU, more info here: EU - GOV.UK.
	b2c	number 1-0	(Windsor Framework - required for NI only) 1=true 0=false. If the goods are considered as delivered to a consumer address, rather than a business address.
	thirdparty	number 1-0	(Windsor Framework - required for NI only) 1=true 0=false. False if delivery is to your registered business address, true otherwise.
			Additional notes on NI - Windsor Framework • If the order is shipping to a third party and that is a consumer and not a business, it must declare <code>thirdparty =1</code>, <code>b2c =1</code>, <code>atrisk =0</code> • If shipping to a third party that is a business, it must declare <code>thirdparty =1</code> and <code>b2c =0</code>, and <code>atrisk</code> as appropriate • If not shipping to a third party, must declare <code>thirdparty =0</code> and <code>b2c =0</code>, and <code>atrisk</code> as appropriate • If your UKIMS number is not registered with PenCarrie, must declare <code>atrisk =1</code> if <code>b2c =0</code>
	plaincover	Number	Selects the branding to be used on the documents that ship with your order. 0 = normal PenCarrie branding -1 = unbranded If we apply custom branding to your orders, a list of other applicable values is available upon request.
	invpicktext	Text	Instructions for the picking team
	invcomment	Text	Free text; can be used to provide instructions to the customer services team and will be printed on the despatch note unless otherwise instructed.
</order>			

<email>			Optional child node of <i>order</i> ; specifies additional contact addresses to send order acknowledgements, despatch notes to
	contact	Text	Name of the contact
	sn_email	Text	Email address to send order documentation to
	pc_ordercode	Text	REQUIRED IN EACH <email> NODE The same value as included in the pc_ordercode attribute of the <order> node
</email>			End of email node
<exportdetails>			REQUIRED if 1) delcountrycode is not GB or 2) or if delcountrycode is GB but the address is in Northern Ireland Child node of <i>order</i> specifying additional contact information required for orders leaving the United Kingdom*
	receivername	Text	Required - The name of a person at the delivery address who can be contacted by DPD or Customs if there are any problems with the order
	receiverphone	Text	Conditional** - A telephone number to contact the above
	receiveremailaddress	Text	Conditional** - An email address to contact the above
</exportdetails>			
>			
</order>			

* If you are shipping to a third-party address, these need to be contact details for someone at that address. If we are shipping directly to your customer, please ensure that you provide *their* details so the courier or customs authorities can contact them if there are any problems with their delivery.

** At least one of receiverphone or receiveremailaddress must be provided if export details are required on the order.

Appendix C: PCGetStock / PCGetStockPrices field list [↗](#)

The pcgetstock function returns information about PenCarrie's current stock for a product. The pcgetstockprices function also includes pricing information, making it slightly slower to run but providing the user with more data. The responses to these functions contain the following fields:

Element	Attribute	Description	pcgetstock	pcgetstockprices
stock	leadtime	Number of days' lead time for supplier stock	Yes	Yes
stock/colours	stat	Deprecated	No	Yes
stock/colours/colour	colourname	Code to identify this colour	Yes	Yes
stock/colours/colour/cwy/c	longname	Full name of the colour	Yes	Yes

stock/colours/colour/cwy/c	head	Corresponds to the identifying code for the colour	No	Yes
stock/colours/colour/cwy/c	hex	Approximate hexadecimal representation of the colour	Yes	Yes
stock/colours/colour/sizes/p	size	Code to identify this garment size	Yes	Yes
stock/colours/colour/sizes/p	fts	PenCarrie's current free-to-sell stock	Yes	Yes
stock/colours/colour/sizes/p	nextday	Deprecated	Yes	Yes
stock/colours/colour/sizes/p	cap	Deprecated	Yes	Yes
stock/colours/colour/sizes/p	mystat	Current status of this product (see below)	Yes	Yes
stock/colours/colour/sizes/p	carton	Price per unit at carton rate	No	Yes
stock/colours/colour/sizes/p	pack	Price per unit at pack rate	No	Yes
stock/colours/colour/sizes/p	single	Price per unit at single rate	No	Yes
stock/colours/colour/sizes/p	supplier	Our supplier's stock level	Yes	Yes

The following product status codes may be returned in the `mystat` field:

Code	Description
-	No special flags or status (this is the normal status for most products)
A	Special-order item purchased upon request, non-returnable
C	Discontinued product, non-returnable (FTS stock is available to order, backorders are not permitted)
D	Product will be discontinued shortly, non-returnable (backorders permitted)
N	New product that has recently been added
S	Code for internal use – please treat this status as if it was “-”

Appendix D: Delivery Codes (del_tid) [🔗](#)

These codes are optional and can be passed into API functions that take a `del_tid` attribute. To ensure your orders are despatched on the correct service, please use an appropriate delivery code for the postcode of the delivery address. If you do not specify a delivery code the default delivery option for your chosen address will be selected automatically. These are indicated in red below.

The list below gives standard options for delivery services.

- Collections
- England and Wales
- Scotland (5 regions)
- Northern Ireland
- Republic of Ireland (Eire)
- Channel Islands
- Isle of Wight
- Isle of Man
- Isles of Scilly
- International Destinations

Collections [🔗](#)

If you are submitting an order for collection from PenCarrie, please specify a custom address, with all address elements and the postcode set to "COLLECT" and a delivery code of 201.

Note that this only applies if you or a representative of your company intends to collect your order in person. If you intend to use a courier to collect your parcel, you will need to provide a full delivery address and select the "Own Carrier" delivery option in the relevant region.

England and Wales [🔗](#)

We use the same delivery codes for all addresses in mainland England and Wales.

Next Day	91
48 Hours	601
Pre-10:30*	93
Pre-12:00*	92
Saturday Pre-12:00	94
Saturday anytime	2242
Sunday Pre-12:00	2244
Sunday anytime	2243
Own Carrier	100

* Note that DPD market the Pre-10:30 and Pre-12:00 services as Pre-11:00 and Pre-13:00 respectively to some addresses in Cornwall. A new region may be created for these addresses in the future which will require different codes to be used. We will advise you if/when this happens.

Scotland [↗](#)

Scotland is divided into five regions by available delivery services. Region 1, which has courier services up to and including pre-11 delivery, consists of addresses with postcodes that begin:

DG, EH, FK1-16, G1-82, G90, KA1-26, KA29-30, KY1-5, KY11-13, KY99, ML, PA1-19 and TD

Region 2 has next day and weekend courier services available, but no timed delivery services. This consists of postcodes beginning with:

AB10-16, AB21-25, AB32, DD, G83-84, KY6-10, KY14-16 and PH1-14

Region 3 has next day courier services available, but no weekend deliveries are possible. This region consists only of postcodes beginning **AB30**.

Region 4 covers areas of the highlands where our couriers cannot guarantee a next day service. Instead, only a two day delivery service is available. This covers postcodes beginning:

AB31, AB33-56, AB99, FK17-21, IV, KA27-28, PA20-80, PH15-39 and KW1-14

Region 5 covers the Orkneys, Shetlands and Outer Hebrides, where only a 2-5 day courier service is available. This is for postcodes beginning:

HS, KW15-17 and ZE

	Region 1	Region 2	Region 3	Region 4	Region 5
2-5 Day	•	•	•	•	203
Two Day	•	•	•	101	•
Next Day	103	231	162	•	•
48 Hours	602	603	604	•	•
Pre-11:00	224	•	•	•	•
Pre-13:00	104	•	•	•	•
Saturday Pre-13:00	107	•	•	•	•
Saturday anytime	106	236	•	•	•
Sunday Pre-13:00	108	•	•	•	•
Sunday anytime	105	235	•	•	•
Own Carrier	175	2248	2251	205	2153

Northern Ireland [↗](#)

Two Day

211

Next Day	210
Own Carrier	209

Republic of Ireland / Eire [🔗](#)

Please note: If you are still using the legacy ZZ71 or ZZ75 postcodes, please give a postcode of ZZ75 and a delivery code of 171 for a two day service outside Dublin, a postcode of ZZ71 and a delivery code of 151 for a two day service to Dublin, or a postcode of ZZ75 and a delivery code of 190 if you are using your own carrier.

If your stored address includes your Eircode postcode, or you are using a custom address and have specified your Eircode in the postcode attribute of the API call, please use the below delivery codes:

Service	Dublin 1-10 / ZZ71	Outside Dublin* / ZZ75
Two Day	151	171
Own Carrier	187	190

* The DPD Dublin region contains the Dublin 1 to Dublin 10 areas, including 6W. For addresses in Dublin 11-24, please use the “Outside Dublin” service codes.

Channel Islands [🔗](#)

The “Other” region includes Sark and Alderney (GY9-10) and special postcodes beginning JE5.

Service	Jersey/Guernsey	Other
2-5 Days	•	161
Two Day	131	•
Own Carrier	193	2245

Isle of Wight [🔗](#)

Next Day	143
48 Hours	605
Pre-12:00	144
Saturday Pre-12:00	145
Saturday anytime	146
Own Carrier	181

Isle of Man [🔗](#)

Two Day	136
Own Carrier	184

Isles of Scilly [🔗](#)

2-5 Days	139
Own Carrier	178

International Destinations [🔗](#)

Outside of the United Kingdom and Republic of Ireland, most countries have a single service code for delivery to all destinations, and a separate service code if you would like to arrange for your own carrier to collect the parcel from PenCarrie and deliver it for you. Country codes highlighted in yellow differ from the standard ISO3166 codes. To ensure your order is delivered correctly, please use the codes specified in this document when shipping to those countries.

Country	Code	DPD Service	Own Carrier
Multiple Services (see above)			
United Kingdom	GB	See Above	
Republic of Ireland	EI		
DPD Classic Two Day service			
Belgium	BE	2372	2373
Luxembourg	LU	2380	2381
Netherlands	NL	218	222
DPD Classic 2-3 Day service			
France	FR	220	221
Germany	WG	2366	2367
DPD Classic Three Day service			
Austria	AT	2255	2256
Denmark	DK	2378	2379
Switzerland	CH	2404	2405
DPD Classic 3-4 Day service			
Czech Republic	CZ	2374	2375
Italy	IT	2368	2369
Slovakia	SK	2400	2401
Spain	ES	2370	2371

DPD Classic 3-5 Day service			
Sweden	SE	219	223
DPD Classic 4-5 Day service			
Croatia	CT	2388	2389
Finland	FI	2364	2365
Hungary	HU	2392	2393
Poland	PL	2382	2383
Portugal	PT	2384	2385
Slovenia	SI	2402	2403
DPD Classic 4-6 Day service			
Bosnia and Herzegovina	BA	2410	2411
Bulgaria	BG	2386	2387
Estonia	EE	2390	2391
Iceland	IC	2406	2407
Latvia	LA	2394	2395
Lithuania	LT	2396	2397
Norway	NO	2408	2409
Romania	RO	2398	2399
Serbia	RS	2412	2413
DPD Classic 4-8 Day service			
Greece	GR	2376	2377
Any Other Destination	Please Contact Us		

Appendix E: Additional Data Requirements for International Shipping [🔗](#)

EDI customers will need to provide this information through the `pcupdate` or `pcautoorder` API calls when they complete their orders, when shipping to any country other than the United Kingdom (i.e. country code is anything other than GB). The additional data is the same for both API calls, and is outlined below in addition to in the descriptions of the relevant calls.

In addition, this data is also be required for shipments within the UK to Northern Ireland. To ensure future compliance we ask that this be provided for all shipments to Northern Ireland, regardless of our carriers' current requirements.

The information provided needs to be for the recipient of the order. **If you are shipping to a third party, it needs to be their contact details, not your own.** These details will be passed to the courier for use in processing the order – we will not use them to contact your customers without your permission and delete them once no longer required.

Field Name	Description	Data Type
------------	-------------	-----------

<code>receivername</code>	Required - The name of a person at the delivery address who can be contacted by DPD or Customs if there are any problems with the order	String (max. 100 characters)
<code>receiverphone</code>	Conditional* - A telephone number to contact the above	String (max. 25 characters)
<code>receiveremailaddress</code>	Conditional* - An email address to contact the above	String (max. 255 characters)

* At least one of `receiverphone` and `receiveremailaddress` is required per order. If the same field will be used for all orders (i.e. you will always provide a phone number, or you will always provide an email address) your CSV file can omit the unused column. Alternatively we recommend including all three columns and populating `receiverphone` and `receiveremailaddress` as required for each individual order.

You are responsible for validating this data before submission. If it is not completed correctly it may cause delivery of your orders to be delayed.

These fields will be required on all orders completed through the API. This may be subject to change going forwards as we add the ability to store and recall saved details. It is only required if you complete your order through the API (i.e. by using `pcupdate` to set the status to F, or `pcautoorder` with a `parkorder` value of 0 or 2) – if you use the API to create the order, but then confirm it manually through our website, you will be prompted to enter these details when you complete the order online.

Once the additional receiver details have been stored against an order, they will be visible in the response to the `pcget` API call when invoked for that order. They can be changed at any time prior to the order shipping, via the `pcupdate` call or by editing your order on our website.

In addition to these changes, some delivery services may take longer to reach their destinations than usual.

References [↗](#)

1. cURL: <https://curl.se/> command line tool and library for transferring data with URLs
2. Python: <https://www.python.org/> Python is a high-level, general-purpose programming language that is used for a variety of purposes
3. PHP: <https://www.php.net/> A popular general-purpose scripting language that is especially suited to web development.
4. C#: <https://dotnet.microsoft.com/en-us/languages/csharp> a general-purpose programming language that's used to develop a range of applications
5. Request Library: <https://pypi.org/project/requests/> allows you to send HTTP/1.1 requests in Python
6. RestSharp package: <https://restsharp.dev/> Simple REST and HTTP API Client for .NET

Version History [↗](#)

Version	Date	Comment
Current Version (v. 16)	14 Apr, 2025 14:16	Trevor Coleman : addition of Windsor Framework attributes and explanatory notes
v. 15	29 Nov, 2024 16:54	Trevor Coleman ip specification
v. 14	29 Nov, 2024 13:36	Trevor Coleman grammar and typo correction

v. 13	27 Nov, 2024 12:48	Trevor Coleman reduced content depth. table reformatting.
v. 12	27 Nov, 2024 12:09	Trevor Coleman GDPR adherence. added redirect for other sample languages. Scope section edits for clarity. pcupdate attribute changes.
v. 11	27 Nov, 2024 11:22	Trevor Coleman added cURL URL redirection switch
v. 10	27 Nov, 2024 11:04	Trevor Coleman pcupdate code samples, php format, delivery services updated, appendix corrections, spelling and grammar corrections
v. 9	31 Oct, 2024 13:41	Trevor Coleman updated appendices
v. 8	31 Oct, 2024 09:26	Trevor Coleman code descriptions
v. 7	29 Oct, 2024 09:50	Trevor Coleman added rate limit placeholder
v. 6	29 Oct, 2024 09:37	Trevor Coleman added supplementary information notes for parameters
v. 5	28 Oct, 2024 11:48	Trevor Coleman proofreading and updated dev environment references
v. 4	28 Oct, 2024 09:27	Trevor Coleman added PHP code samples, appendices, and clarified language
v. 3	10 Oct, 2024 09:36	Trevor Coleman Improved clarity in overview paragraph, added contact advice on other options, function format
v. 2	27 Sept, 2024 12:26	Trevor Coleman added pcautoorder samples
v. 1	20 Sept, 2024 10:01	Trevor Coleman