

Illuminate Rest Services

User Guide



Contents

1	General API Calls	4
1.1	Login	4
1.2	System Lists	5
1.3	Postcode Details	7
2	Inventory API Calls	9
2.1	Product Search	9
2.2	Get Tree Items	10
2.3	Manage Tree Item	11
2.4	Section Details	12
2.5	Manage Section	13
2.6	Product Details	14
2.7	Manage Product	15
2.8	Purchase Order	17
2.9	Add to Purchase Order	19
2.10	Manage Purchase Order	20
2.11	Create Purchase Order	21
2.12	Goods Inwards	22
2.13	Warehouse Details	23
2.14	Manage Warehouse	24
2.15	Supplier Details	25
2.16	Manage Supplier	26
2.17	Safety and Review	27
3	Customer Management	28
3.1	Customer Types	28
3.2	Customer Search	30
3.3	Customer Details	32
3.4	Manage Customer	35
4	Order Management	37
4.1	Progress Order Info	37
4.2	Progress Order Action	38
4.3	Order Details	39
4.4	Manage Order	40

1 General API Calls

Title	1.1 Login <i>Logs into the system and retrieves a security key ('UserKey').</i>
URL	<code>/login</code>
Method	<code>POST</code>
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>username : your username password : your password</p> <p>Example:</p> <pre>{'username': 'fred', 'status': 'canberra'}</pre>
Success Response	<p>The call returns fields as shown in the example below :</p> <pre>{ "UserKey": "7157052330137238", "Status": "Success", "LastURL": "ReportMain.aspx" }</pre> <p>UserKey is the key which should be passed for all subsequent REST calls for this session.</p> <p>Status indicates success or failure.</p> <p>LastURL indicates the last URL which the user accessed.</p>
Error Response	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "Log in failure" }</code></p>
Notes	

<p>Title</p>	<p>1.2 System Lists <i>Returns the specified system list.</i></p>
<p>URL</p>	<p><code>/systemlist</code></p>
<p>Method</p>	<p><code>POST</code></p>
<p>URL Params</p>	<p>None</p>
<p>Data Params</p>	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>listname: the system list to be retrieved. The system lists which are available can be found in the Notes section.</p> <p>Example:</p> <pre>{'userkey':'7157052330137238', 'listname': 'supplierlist'}</pre>
<p>Success Response</p>	<p>The call returns the list in question, with System IDs, Text and other details, depending on the list. This list is passed in an items array. The example below is for the 'supplierlist' system list.</p> <pre>{ "items":[{ "ItemText":"Supplier 600001", "ItemID":600001 }, { "ItemText":"Supplier 600002", "ItemID":600002 }, { "ItemText":"Supplier 600079", "ItemID":600079 }] }</pre>
<p>Error Response</p>	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "List not available" }</code></p>

Notes	The available system lists are : campaign - list of marketing campaigns shippingType – list of shipping/postage types supplierlist – list of suppliers. user – list of system users
--------------	---

<p>Title</p>	<p>1.3 Postcode Details Returns postcode details for a given addressid. addressid's are returned from customer and supplier search results, and from other areas of the system.</p>
<p>URL</p>	<p>/postcodedetails</p>
<p>Method</p>	<p>POST</p>
<p>URL Params</p>	<p>None</p>
<p>Data Params</p>	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>addressid: the addressed to be retrieved.</p> <p>Example:</p> <pre>{“userkey”: “7157052330137238”, “addressid”: “23821671.00”}</pre>
<p>Success Response</p>	<p>The call returns the address details, in individual fields. These can then be mapped the Customer Fields (via their 'PAF' value) to perform the correct field mappings.</p> <pre>{ "postcodefields": [{ "FieldNo": 1, "AddressData": "" }, { "FieldNo": 2, "AddressData": "" }, { "FieldNo": 3, "AddressData": "Cleanology Ltd" }, { "FieldNo": 4, "AddressData": "7 Lion Yard" }, { "FieldNo": 5, "AddressData": "London" }, { "FieldNo": 6, "AddressData": "" }, { "FieldNo": 7, "AddressData": "SW4 7NQ" }] }</pre>

	<pre>], "responsestring": "Success" }</pre>
Error Response	<p>In the event of an error, an error string is returned. Examples: Content: { responsestring : "Error : Invalid Userkey" }</p>
Notes	

2 Inventory API Calls

Title	2.1 Product Search <i>Performs a product search.</i>
URL	<code>/productsearch</code>
Method	<code>POST</code>
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>productcode: A product code to search for.</p> <p>productname: A product name to search for. Wildcards ('*') characters can be included)</p>
Success Response	The call returns an array of product information, with one product in each array element. An empty array is returned if no products can be found.
Error Response	<p>In the event of an error, an error string is returned in the response.</p> <p>Examples:</p> <p>Content: <code>{"error":"Invalid input"}</code></p>
Notes	

Title	<h3>2.2 Get Tree Items</h3> <p><i>Gets tree items (categories and products) for the given parentid. N.b. the top level items can be obtained by passing a parentid of 0.</i></p>
URL	<code>/gettreeitems</code>
Method	<code>POST</code>
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>parentid: the parentid of the items to retrieve.</p>
Success Response	<p>The call returns an array of items, including a 'treeid' which can be used as the parentid in subsequent calls.</p> <p>Note that slightly different output is returned for categories (which have a sectionid) and products (which have a productid).</p>
Error Response	<p>In the event of an error, an error string is returned in the response.</p> <p>Examples:</p> <p>Content: <code>{"error":"Invalid input"}</code></p>
Notes	

2.3 Manage Tree Item

Updates, copies or deletes a hierarchy item.

`/treeitemmanage`

`POST`

This call expects the following parameters :

Mandatory

userkey : your security key

action : this should be set to 'copy', 'update' or 'delete'.

treeid : This is the 'treeid' of the product/section to be copied, updated or deleted.

Optional

parentid : This is the destination parent 'treeid' of the item (not required for deletes).

prevlink : This is the destination 'prevlink' (i.e. position) of the item (not required for deletes).

Examples

```
{"userkey":"Yo", "action":"update", "treeid":"625976", "parentid":"624787", "prevlink":"1"}
```

```
{"userkey":"Yo", "action":"copy", "treeid":"625976", "parentid":"624303", "prevlink":"13"}
```

In the event of successful update a 'success' message is returned. e.g.

Content: `{ "responsestring" : "Success" }`

In the event of an error, an error string is returned.

Examples:

Content: `{ error : "Invalid Input" }`

Title	2.4 Section Details <i>Returns full section details.</i>
URL	<code>/sectiondetails</code>
Method	<code>POST</code>
URL Params	None
Data Params	This call expects the following parameters : Mandatory userkey : your security key sectionid : A sectionid to search for.
Success Response	The call returns section information
Error Response	In the event of an error, an error string is returned in the response. Examples: Content: <code>{"error":"Invalid input"}</code>
Notes	

Title	<p>2.5 Manage Section <i>Updates, inserts or deletes a section.</i></p>
URL	<p><code>/sectionmanage</code></p>
Method	<p><code>POST</code></p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>sectionid : the 'sectionid' for the sectionid. This is not required for new sections (the 'insert' action).</p> <p>action : this should be set to 'insert', 'update' or 'delete'.</p> <p>parentid : This must be included for 'insert' calls only. It is the 'TreeID' of the section to which the item is being added.</p> <p>prevlink : This must be included for 'insert' calls only. This represents the position of the item within the section. The first item is 0, the second item is 1, etc.</p> <p>Optional</p> <p>general object : This contains fields for the section. The field names which should be passed will be expanded and will change over time. However, they are same as those from the 'sectiondetails' API call.</p> <p>Examples :</p> <pre>{"userkey":"m4ES", "parentid":"0", "prevlink":"5", "action":"insert", "general": { "SectionName":"New Section"}}</pre> <pre>{"userkey":"4m4ES", "sectionid":"618581", "action":"delete", "general": {}}</pre>
Success Response	<p>In the event of successful update, the section object is returned and also a 'success' message. e.g.</p> <p>Content: <code>{ "responsestring" : "Success" }</code></p>
Error Response	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "Invalid Input" }</code></p>
Notes	

Title	2.6 Product Details <i>Returns full product details.</i>
URL	<code>/productdetails</code>
Method	<code>POST</code>
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>productcode: A product code to search for.</p> <p>Optional</p> <p>hierarchy : can be 1 (if present). This will return the section and product hierarchy which 'leads' to the product. This can be in conjunction with the productsearch API call.</p>
Success Response	The call returns product information, plus arrays of supplier and warehouse information.
Error Response	In the event of an error, an error string is returned in the response. Examples: Content: <code>{"error":"Invalid input"}</code>
Notes	

Title	<h2>2.7 Manage Product</h2> <p><i>Updates, inserts or deletes a product.</i></p>
URL	<code>/productmanage</code>
Method	<code>POST</code>
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>productcode : the 'productcode' for the product. This is not required for new products (the 'insert' action).</p> <p>action : this should be set to 'insert', 'update' or 'delete'.</p> <p>parentid : This must be included for 'insert' calls only. It is the 'TreeID' of the section to which the item is being added.</p> <p>prevlink : This must be included for 'insert' calls only. This represents the position of the item within the section. The first item is 0, the second item is 1, etc.</p> <p>Optional</p> <p>general object : This contains fields for the product. The field names which should be passed will be expanded and will change over time. However, they are same as those from the 'productdetails' API call.</p> <p>Example :</p> <pre>{"userkey":"rjft1^%95I6jEQNGqY ", "productcode":"712141", "action" : "update", "general": {"MagentoID":"567","ProductPrice":"1.99"}}</pre> <p>suppliers array : This contains suppliers for the product, together with an action to 'insert', 'update' or 'delete'.</p> <p>The following fields names can be passed as part of a suppliers array item :</p> <ul style="list-style-type: none">• SupplierName (Required)• SupplierCode• BuyPrice• BarCode <p>Example :</p>

	<pre> {"userkey":" @D83nVW3887qAiZ5!jGSltNb", "productcode":"712141", "action" : "update", "general": {}, "suppliers": [{"action" : "update", "SupplierName":"Clothing Supplier Ltd","Barcode":"11111111"}]} </pre> <p>warehouses array : This contains warehouses for the product, together with an action to 'insert', 'update' or 'delete'.</p> <p>The following fields names can be passed as part of a warehouse array item :</p> <ul style="list-style-type: none"> • WarehouseName (Required) • BinName (Required) • StockLevel <p>Example :</p> <pre> {"userkey":"jGSltNb", "productcode":"712141", "action":"update", "general": {}, "warehouses": [{"action" : "insert", "WarehouseName":"Main Warehouse", "BinName" : "Bin 600008", "StockLevel":"500"}]} </pre>
<p>Success Response</p>	<p>In the event of successful update, a 'success' message is returned. e.g.</p> <p>Content: <code>{ "Response" : "Success" }</code></p>
<p>Error Response</p>	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "Invalid Warehouse Name" }</code></p>
<p>Notes</p>	

Title	<p>2.8 Purchase Order <i>Searches for a purchase order or orders.</i></p>
URL	<p><code>/purchaseorder</code></p>
Method	<p><code>POST</code></p>
URL Params	<p>None</p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>Optional</p> <p>status: can be 'Open' or 'Closed', to return either open or closed purchase orders</p> <p>supplierorderid: A specific purchase order number.</p> <p>Examples:</p> <pre>{'userkey':'7157052330137238', 'status': 'Open'}</pre> <pre>{'userkey':'7157052330137238', 'supplierorderid': '602331'}</pre>
Success Response	<p>The call returns basic supplier order data in a 'response' array. Each order contains an 'items' array of the ordered products.</p> <p>Example :</p> <pre>{ "response": [{ "SubTotal":234.0099994506836, "items": [{ "SupplierOrderStockItemID":733279, "MinQty":0, "Price":0.930000007152557, "DateExpected":"28\11\2014", "Quantity":55, "Description":"Galvanised Steel Mud Guard", "Code":"713005", "SupplierCode":"713005", "StockItemID":713005 }, {</pre>

	<pre> "SupplierOrderStockItemID":733285, "MinQty":0, "Price":3.089999991416931, "DateExpected":"28\11\2014", "Quantity":24, "Description":"Road-550-W", "Code":"712902", "SupplierCode":"712902", "StockItemID":712902 }], "ShipTotal":0, "DeliveryLocation":""," "PONumber":"602244", "SupplierName":"Supplier 600061", "DateExpected1":"28\11\2014", "TotalValue":280.81201171875 }] } </pre>
<p>Error Response</p>	<p>In the event of an error, an error string is returned. Examples: Content: { error : "Log in Error" }</p>
<p>Notes</p>	

Title	2.9 Add to Purchase Order <i>Add an item to a purchase order</i>
URL	<code>/addtopo</code>
Method	<code>POST</code>
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>productcode: a valid product code.</p> <p>suppliername: Supplier Name for the PO.</p> <p>quantity : The quantity to be ordered</p> <p>reason : short description of the reason for the order</p> <p>info : longer description of the reason for the order</p> <p>Example:</p> <pre>{"userkey":"Z20hzDy3##f", "productcode":"712141", "suppliername":"Clothing Supplier Ltd", "quantity":"20","reason":"Required for Orders","info":"Placed by Mike" }</pre>
Success Response	<p>The call returns a response string indicating success, for example :</p> <pre>{"response": "Item added to Purchase Order 507211"}</pre>
Error Response	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "Log in Error" }</code></p>
Notes	

Title	<p>2.10 Manage Purchase Order <i>Amend or delete a purchase order or purchase order items</i></p>
URL	<p><code>/purchaseordermanage</code></p>
Method	<p><code>POST</code></p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>supplierorderid : the 'supplierorderid' of the PO to manage.</p> <p>action : this should be set to 'update' or 'delete'.</p> <p>general object : This contains fields for the order. The field names which should be passed will be expanded and will change over time. However, they are same as those from the 'purchaseorder' API call. The following fields can be updated :</p> <ul style="list-style-type: none"> • deliverymethod • shiptotal • dateexpected <p>Optional</p> <p>items : this is an array of the items within a purchase order. An action should also be specified ('update' or 'delete'). Note that 'supplierorderstockitemid' (the unique key for the item) is required for 'update' and 'delete' actions. The following fields can be updated:</p> <ul style="list-style-type: none"> • quantity • price • dateexpected
Success Response	<p>In the event of successful update, the warehouse object is returned and also a 'success' message. e.g.</p> <p>Content: <code>{ "responsestring" : "Success" }</code></p>
Error Response	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "Invalid Input" }</code></p>
Notes	

Title	2.11 Create Purchase Order <i>Convert a draft purchase order (PO) into a live PO.</i>
URL	<code>/createpo</code>
Method	<code>POST</code>
URL Params	None
Data Params	This call expects the following parameters : Mandatory userkey : your security key supplierorderid: The OrderID of the PO.
Success Response	The call returns a response string indicating success, for example : <code>{"responsestring": " Purchase Order Created 602334"}</code>
Error Response	In the event of an error, an error string is returned. Examples: Content: <code>{ error : "Log in Error" }</code>
Notes	

Title	<p>2.12 Goods Inwards <i>Accepts goods once a delivery has been received.</i></p>
URL	<p><code>/goodsinwards</code></p>
Method	<p><code>POST</code></p>
URL Params	<p>None</p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>supplierorderid: A specific purchase order number.</p> <p>Items array : an array of productcodes and qtyin fields which indicate which items have been received.</p> <p>Examples:</p> <pre>{ 'userkey': '7157052330137238', 'supplierorderid': '602320', 'items': [{ productcode: 713305, qtyin: 30 }] }</pre>
Success Response	<p>The call returns the following response if successful :</p> <pre>{"response": "Goods Inwards Received Successfully"}</pre>
Error Response	<p>In the event of an error, an error string is returned in the response.</p> <p>Examples:</p> <p>Content: <code>{"response": "Incorrect Item : 713305 : 30"}</code></p>
Notes	

Title	2.13 Warehouse Details <i>Returns full warehouse and bin details.</i>
URL	<code>/warehousedetails</code>
Method	<code>POST</code>
URL Params	None
Data Params	This call expects the following parameters : Mandatory userkey : your security key warehouseid : A warehouseid to search for.
Success Response	The call returns warehouse information
Error Response	In the event of an error, an error string is returned in the response. Examples: Content: <code>{"error": "Invalid input"}</code>
Notes	

Title	<p>2.14 Manage Warehouse <i>Updates, inserts or deletes warehouses and bins.</i></p>
URL	<p><code>/warehousemanage</code></p>
Method	<p><code>POST</code></p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>warehouseid : the 'warehouseid' for the warehouse. This is not required for new warehouses (the 'insert' action).</p> <p>action : this should be set to 'insert', 'update' or 'delete'.</p> <p>general object : This contains fields for the warehouse. The field names which should be passed will be expanded and will change over time. However, they are same as those from the 'warehousedetails' API call.</p> <p>Optional</p> <p>warehousebins : this is an array of the bins within the warehouse. An action should also be specified ('insert', 'update' or 'delete'). Note that 'BinID' is required for 'update' and 'delete' actions. The fields are the same as those returned by the 'warehousedetails' API call.</p> <p>Example :</p> <pre>{ "userkey": "1zWdSpKy", "action": "insert", "general": { "WarehouseName": "New Warehouse", "Address": "8 Lion Yard, London, SW4 7NQ" }, "warehousebins": [{ "action": "insert", "BinName": "New Bin 999" }] }</pre>
Success Response	<p>In the event of successful update, the warehouse object is returned and also a 'success' message. e.g.</p> <p>Content: <code>{ "responsestring" : "Success" }</code></p>
Error Response	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "Invalid Input" }</code></p>
Notes	

Title	2.15 Supplier Details <i>Returns full supplier details.</i>
URL	<code>/supplierdetails</code>
Method	<code>POST</code>
URL Params	None
Data Params	This call expects the following parameters : Mandatory userkey : your security key supplierid : A supplierid to search for.
Success Response	The call returns supplier information in form of an array of 'F' fields (F1, F2, F3 etc., where F1 is the first field, F2 the second, and so on)
Error Response	In the event of an error, an error string is returned in the response. Examples: Content: <code>{"error": "Invalid input"}</code>
Notes	

Title	<p>2.16 Manage Supplier <i>Updates, inserts or deletes suppliers and delivery methods.</i></p>
URL	<p><code>/suppliermanage</code></p>
Method	<p><code>POST</code></p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>supplierid: the 'supplierid' for the supplier. This is not required for new suppliers (the 'insert' action).</p> <p>action : this should be set to 'insert', 'update' or 'delete'.</p> <p>general object : This contains fields for the supplier. The field names which should be passed will be expanded and will change over time. However, they are same numbered 'F' as those from the 'supplierdetails' API call.</p> <p>Optional</p> <p>deliverymethods : this is an array of the delivery methods for the supplier. An action should also be specified ('insert', 'update' or 'delete'). Note that 'deliverymethodid' is required for 'update' and 'delete' actions. The following fields can be specified :</p> <ul style="list-style-type: none"> • cost • deliverytime • isdefault <p>Example :</p> <pre>{ "userkey": "7AEG4", "action": "insert", "general": { "F1": "Illuminate Insight Ltd", "F2": "2 Heswall Rd" }, "deliverymethods": [{ "action": "insert", "deliverymethodid": "2", "cost": "3.99" }] }</pre>
Success Response	<p>In the event of successful update, the supplier object is returned and also a 'success' message. e.g.</p> <p>Content: <code>{ "responsestring" : "Success" }</code></p>
Error Response	<p>In the event of an error, an error string is returned.</p> <p>Examples:</p> <p>Content: <code>{ error : "Invalid Input" }</code></p>

Title	2.17 Safety and Review <i>Get current Safety and Review stock levels for all products</i>
URL	<code>/safetyandreview</code>
Method	<code>POST</code>
URL Params	None
Data Params	This call expects the following parameters : Mandatory userkey : your security key
Success Response	The call returns a response string indicating success, for example : <code>{"responsestring": " Success"}</code> Together with the safety and review stock information.
Error Response	In the event of an error, an error string is returned. Examples: Content: <code>{ error : "Log in Error" }</code>
Notes	

3 Customer Management

Title	<p>3.1 Customer Types <i>Returns the customer types within the system.</i></p>
URL	/customertypelist
Method	POST
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>Example:</p> <pre>{'userkey' : '7157052330137238' }</pre>
Success Response	<p>The call returns the customer types within the system.</p> <pre>{ "response": [{ "customertypeid": 5, "items": [{ "fax": 0, "searchable": 0, "originalfieldno": 1, "searchpos": 0, "autoinc": 1, "fieldno": 1, "customertypeid": 5, "internet": 0, "custpos": 8, "location": 0, "picklistid": 0, "fieldname": "Customer No", "mandatory": false, "pafvalue": 0, "addrpos": 0, "email": 0, "picklist": [],],]] }] }</pre>

	<pre> "addrline":0, "webpos":0, "phone":0 }, { "fax":0, "searchable":1, "originalfieldno":2, "searchpos":1, "autoinc":0, "fieldno":2, "customertypeid":5, "internet":1, "custpos":10, "location":0, "picklistid":0, "fieldname":"Postcode", "mandatory":true, "pafvalue":7, "addrpos":1, "email":0, "picklist":[], "addrline":7, "webpos":9, "phone":0 }] }</pre>
Error Response	In the event of an error, an error string is returned. Examples: Content: { error : "List not available" }
Notes	

<p>Title</p>	<p>3.2 Customer Search <i>Searches for customers within the system.</i></p>
<p>URL</p>	<p><code>/customersearch</code></p>
<p>Method</p>	<p><code>POST</code></p>
<p>URL Params</p>	<p>None</p>
<p>Data Params</p>	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>customertype : the customer type being searched.</p> <p>searchtype : indicates whether Billing ('billing') or Delivery ('delivery') addresses are being searched.</p> <p>soundex : indicates whether a 'sounds like' search is being performed (true or false)</p> <p>searchfields : indicates the search fields and criteria, which consists of an array of 'fieldname' and 'fieldvalue' pairs.</p> <p>Example:</p> <pre> } {"userkey":"7157052330137238", "customertype": "Consumer", "searchtype":"billing", "soundex":false, "searchfields":[{"fieldname":"Forename", "fieldvalue":"T*"}, {"fieldname":"Surname", "fieldvalue":"Stockard"}]}; </pre>
<p>Success Response</p>	<pre> { "postcodelist": [], "response": [{ "Address": "1 Sarah St", "RFM": 0, "Expired": 0, "TotalSpend": "0.00", "OriginalSource": "ebay", "NoEMail": "", "Forename": "Tessie", "LastOrder": "08\10\2014", "NoRent": "", </pre>

	<pre>"Town/City": "Verwood", "EntryDate": "08/10/2014", "E-Mail": "tstockard@hotmail.com", "CreditStatus": "0.00", "Telephone": "01341-875380", "Country": "United Kingdom", "Mobile": "01274-655509", "Surname": "Stockard", "E-Bay": "", "Title": "", "Ship": "", "NoDM": "2", "Ebay": "", "County/State": "Dorset", "CustomerID": 675671, "Customer": "", "Postcode": "BH21 6QX" }] }</pre>
Error Response	<p>In the event of an error, an error string is returned. Examples: Content: { error : "Invalid Criteria" }</p>
Notes	

<p>Title</p>	<p>3.3 Customer Details <i>Returns full details for a customer within the system, together with orders and CRM contacts for the customer.</i></p>
<p>URL</p>	<p><code>/customersearch</code></p>
<p>Method</p>	<p><code>POST</code></p>
<p>URL Params</p>	<p>None</p>
<p>Data Params</p>	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>customerid : the customerid for the customer.</p> <pre>{“userkey”:"7157052330137238", "customerid": "5819875"}</pre>
<p>Success Response</p>	<pre>{ "response": [{ "Address": "28 Oakleigh", "RFM": 0, "Expired": 0, "TotalSpend": "0.00", "OriginalSource": "ebay", "NoEMail": "0", "deliveryaddresses": [{ "Address": "4 Montague Avenue", "Forename": "Ti", "Town/City": "London", "E-Mail": "tim.gibbons@igdev.co.uk", "Telephone": "0207622 3631", "Country": "Poland", "DeliveryID": 601388, "Mobile": "", "Surname": "Gibbons", "E-Bay": "", "Title": "Mr", "Ship": "", "Ebay": "", "County/State": "", "Customer": "(new address)", "Postcode": "SE4 1YP" }] }] }</pre>


```
    }
  ],
  "Forename": "Tyisha",
  "LastOrder": "05\12\2014",
  "NoRent": "0",
  "Town\City": "Ryton-on-Dunsmore",
  "EntryDate": "05\12\2014",
  "E-Mail": "tworster@worster.co.uk",
  "CreditStatus": "-9.30",
  "Telephone": "01769-652551",
  "Country": "United Kingdom",
  "Mobile": "01767-348686",
  "contacts": [
    {
      "DateLogged": "15\01\2015",
      "Category2": "",
      "Category1": "New Category",
      "Status": "Open",
      "Category": "Cats",
      "ContactID": 8504
    }
  ],
  "orders": [
    {
      "DespatchMethod": "First Class",
      "DateLogged": "05\12\2014",
      "ProductStatus": "SH",
      "Quantity": 1,
      "OrderID": 46117,
      "Description": "LL Fork",
      "ShipDate": "05\12\2014",
      "PricePaid": "6.85"
    }
  ],
  "Surname": "Worster",
  "E-Bay": "",
  "Title": "",
  "Ship": "",
  "NoDM": "2",
  "Ebay": "",
  "County\State": "Warwickshire",
  "CustomerID": 682025,
  "Customer": "",
  "Postcode": "CV8 3EA"
}
]
```

Error Response	In the event of an error, an error string is returned. Examples: Content: <code>{ error : "Invalid Criteria" }</code>
Notes	

Title	<p style="text-align: center;">3.4 Manage Customer <i>Updates, inserts or deletes a customer record.</i></p>
URL	<p><code>/customermanage</code></p>
Method	<p><code>POST</code></p>
URL Params	<p>None</p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>customerid : the 'customerid' for the customer. This should be set to -1 for a new customer.</p> <p>action : this should be set to 'insert', 'update' or 'delete'.</p> <p>customertype : the customertype (e.g. 'Consumer' of the customer). This is required for inserting new customers only</p> <p>A list of customer can also be included. Field Names must be valid and match those received from the customerdetails api call.</p> <p>An array of deliveryaddresses must also be passed, even if empty. If filled, this should include a unique 'deliveryid' to indicate which delivery address to update, and also an "action" field to indicate to 'create', 'update' or 'delete'. Deliveryid should be set to -1 for a new delivery address.</p> <p>Example (Customer Update) :</p> <pre> {"userkey": "7157052330137238", "customerid": "5819875", "action": "update", "E-Mail" : "tim.gibbons@igdev.co.uk", "deliveryaddresses" : [{"action" : "update", "deliveryid" : "61716", "Forename": "Tim"}, {"action" : "insert", "deliveryid" : "-1", "Forename": "Mike"}] }</pre>
Success Response	<p>In the event of successful update, a 'success' message is returned. If a new customer has been inserted, the customerid is returned.</p> <p>e.g.</p> <p>Content: <code>{ info : "Customer Inserted", "customerid" : 12345 }</code></p>

Error Response	In the event of an error, an error string is returned. Examples: Content: <code>{ error : "Invalid Criteria" }</code>
Notes	

4 Order Management

Title	<h3>4.1 Progress Order Info</h3> <p><i>Retrieves information required to fill the 'Progress Orders' form.</i></p>
URL	<code>/progressorderinfo</code>
Method	<code>POST</code>
URL Params	None
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>orders array : an array of order numbers</p> <p>Examples:</p> <pre>{"userkey": "bchcvbhcvb", "orders": ["178101", "178102", "178103", "178104"]}</pre>
Success Response	<p>The call returns 3 arrays which can be used to fill the 'Progress Orders' form.</p> <p>orders – can be used to fill the main grid</p> <p>actions – can be used to generate the 'Action to Perform' tick boxes and their values.</p> <p>documents - can be used to generate the 'Documents to Print' tick boxes and their values</p>
Error Response	<p>In the event of an error, an error string is returned in the response.</p> <p>Examples:</p> <p>Content: <code>{"error": "Invalid Order Number"}</code></p>
Notes	

Title	<p style="text-align: center;">4.2 Progress Order Action <i>Submits information required for printing an order or orders.</i></p>
URL	<p><code>/progressorderaction</code></p>
Method	<p><code>POST</code></p>
URL Params	<p>None</p>
Data Params	<p>This call expects the following parameters :</p> <p>Mandatory</p> <p>userkey : your security key</p> <p>The call accepts 3 arrays, similar to those which are generated by the 'progressorderinfo' API call.</p> <p>orders – this is an array which should contain one item for each order. Order fields to be included should be as follows :</p> <ul style="list-style-type: none"> • OrderID • DespatchType (e.g. 'First Class') • NumLabels (Numeric) • Service (10, 12, 24 or 48) • RequireInvoice (true or false) <p>actions – This array are the same format as that generated by the 'progressorderinfo' API call.</p> <p>documents - This array are the same format as that generated by the 'progressorderinfo' API call.</p>
Success Response	<p>In the event of success, a success string is returned in the response. Examples: Content: <code>{"success": "Progress Order Completed"}</code></p>
Error Response	<p>In the event of an error, an error string is returned in the response. Examples: Content: <code>{"error": "Invalid Order Number"}</code></p>
Notes	

Title	4.3 Order Details <i>Returns full order details.</i>
URL	<code>/orderdetail</code>
Method	<code>POST</code>
URL Params	None
Data Params	This call expects the following parameters : Mandatory userkey : your security key orderid : An order to search for.
Success Response	The call returns order information and various arrays such as orderproducts, orderpayments, orderlog, BillingAddress and DeliveryAddress which contain further detailed information about the order.
Error Response	In the event of an error, an error string is returned in the response. Examples: Content: <code>{"error":"Invalid input"}</code>
Notes	

4.4 Manage Order

Updates, inserts or deletes an order.

`/ordermanage`

`POST`

None

This call expects the following parameters :

Mandatory

userkey : your security key

action : this should be set to 'insert', 'update' or 'delete'.

Other parameters depend on whether the 'action' value:

action	Parameters required
insert	customerid : the id of the customer
update	orderid : the id of the order
delete	orderid : the id of the order

Optional

general object : This contains fields for the order. The field names which should be passed will be expanded and will change over time. However, they are same as those from the **orderdetail** API call. The object can be empty if there are no fields to be included.

Example of a simple 'insert' to create a new order.

```
{"userkey":"3KASIW4cQ%E$w8fMeph"}, "action":"insert", "customerid":"675001", "general": {}, "orderproducts": []}
```

orderproducts array : This can be included for the 'insert' and 'update' actions and contains order product values, together with an 'action' of 'insert', 'update' or 'delete' depending on the action required.

The product object parameters required depends on the 'action' value:

action	Parameters required
insert	productcode : the code of the product quantity : the quantity required pricepaid : unit price for the item
update	orderproductid : the unique orderproductid of the order item quantity : the quantity required pricepaid : unit price for the item

cancel	orderproductid : the unique orderproductid of the order item quantity : the quantity being cancelled reason : the reason for the cancellation
return	orderproductid : the unique orderproductid of the order item quantity : the quantity being returned reason : the reason for the return nextaction : the next action to be performed
delete	orderproductid : the unique orderproductid of the order item

Example of inserting a product :

```
{"userkey":"3KASIW4cQ%E$w8fMeph)", "action":"update", "orderid":"178128", "general":
{"orderproducts": [{"action":"insert", "productcode":"712141", "quantity":"20", "pricepaid":"6.50"}]}
```

Example of updating a product :

```
{"userkey":"3KASIW4cQ%E$w8fMeph)", "action":"update", "orderid":"178128", "general":
{"orderproducts": [{"action":"update", "orderproductid":"700792", "quantity":"30",
"pricepaid":"6.50"}]}
```

Example of cancelling a product:

```
{"userkey":"3KASIW4cQ%E$w8fMeph)", "action":"update", "orderid":"178128", "general":
{"orderproducts": [{"action":"cancel", "orderproductid":"700792", "quantity":"30", "reason":"Change
of Mind"}]}
```

Example of returning a product:

```
{"userkey":"3KASIW4cQ%E$w8fMeph)", "action":"update", "orderid":"178128", "general":
{"orderproducts": [{"action":"return", "orderproductid":"700792", "quantity":"30", "reason":"Change
of Mind", "nextaction": "Return to Stock"}]}
```

orderpayments array : This can be included for the 'insert' and 'update' actions and contains order payments values, together with an 'action' of 'insert', 'update' or 'delete' depending on the action required.

The payment object parameters required depends on the 'action' value:

action	Parameters required
insert	none
update	paymentid : the paymentid being updated
delete	paymentid : the paymentid being deleted

Example of inserting a payment :

```
{"userkey":"3KASIW4cQ%E$w8fMeph)", "action":"update", "orderid":"178128", "general":  
{}, "orderpayments": [{"action":"insert"}]}
```

Example of updating a payment :

```
{"userkey":"3KASIW4cQ%E$w8fMeph)", "action":"update", "orderid":"178128", "general": {},  
orderpayments ": [{"action":"update", "paymentid":"700792", "cardnumber":"4751300078651234",  
"amount":"6.50", "cardtype":"VISA"}]}
```

The payment fields passed can be any previously returned from the **orderdetail** API call.

In the event of successful update, a 'success' message is returned. The complete order object is also returned.

Content: { "responsestring" : "Success" }

In the event of an error, an error string is returned.

Examples:

Content: { error : "Invalid Input" }