API Capabilities – v1_9 – 27/09/19

V1_9 Public Functions

Note: Commented VB.NET integration code examples are available on request

---

`create_new_order(mw_api_credentials,mw_api_order)`

Creates a new order on the myWarehouse system. Takes two arguments, the credentials object and the order object, defined below

Returns: (Int32) a unique new order id or 0 if unsuccessful

`get_order_status(mw_api_credentials, order_unique_id)`

Reports back on status of an existing order. Takes two arguments, the credentials object and the order_unique_id to be checked. Note that the order_unique_id must be one that has been supplied back to you via the API's **create_new_order** function above

Returns: (mw_order_status object) as defined below

`get_stock_level(mw_api_credentials, product_sku)`

Shows current free stock of the given SKU. Takes two arguments, the credentials object and the product_sku (unique to you as a client, must correspond to the myWarehouse SKU that shows on the main MMS web forms)

Returns: (Int32) an available stock quantity, 0 for no stock, -1 for error

---

`get_all_skus_with_overnight_stock_levels(mw_api_credentials)`

Gets all active SKUs relating to the given client logs. Takes one argument, the credentials object, and returns an array containing SKUs and Overnight Stock Level for each SKU listed

Returns: (Array of `mw_api_stock_item`) , each with SKU and Overnight Stock Quantity

---

`create_new_product(mw_api_credentials, mw_api_product)`

Creates a new product on MMS. If your API username is in TEST mode, the product is dropped into your TEST product catalogue, if in LIVE then it's dropped into your LIVE product catalogue. The SKU and Barcode must both be unique within your current (TEST or LIVE) catalogue of products on MMS.

Although the `Barcode` field is required, if the barcode is set as "MYWAREHOUSE" then a new MMS barcode will be assigned automatically to the new product. To get this new barcode after creation, call `get_product_by_sku` and examine the barcode field.

In LIVE mode (not TEST), if a product is created with `product_initial_stock_qty` set to a number greater than 0, a new Purchase order **will be created for that new product automatically** so goods can be booked in. If you wish to create multiple new SKUs and assign them to a single PO, please create them first with `product_initial_stock_qty` set to 0, then use the `create_new_purchase_order` function to create the PO.

Returns: (Boolean), Success

`get_product_by_sku(mw_api_credentials, product_sku)`

Retrieves an existing product from MMS. If your API username is in TEST mode, the product is retrieved from your TEST product catalogue, if in LIVE then it's retrieved from the LIVE product catalogue.

Takes two arguments, the credentials object and a product SKU.

Note that an error will be generated if no product is found in your catalogue with that SKU.

Returns: (`mw_api_product`) as defined below

`create_new_purchase_order(mw_api_credentials, mw_api_purchase_order)`

Creates a new PO on the system, against which goods can be booked in. If your API username is in TEST mode, the PO is dropped into your TEST PO table, if in LIVE then it's dropped into your LIVE PO table.

Purchase orders must contain at least one SKU. Every SKU on the PO must already exist on MMS in your LIVE product catalogue (regardless of whether you're creating a TEST or LIVE PO), and every item on the PO must have a quantity greater than 0.

If SKUs need to be created first, use the `create_new_product` function with `product_initial_stock_qty` set to 0, then create a PO using this function

Returns: (Boolean), Success

FUNCTION SUMMARY		
Function	API user in LIVE mode	API user in TEST mode
`create_new_order`	New order created in **LIVE** order table, must contain products from **LIVE** catalogue that are in stock	New order created in **TEST** order table, must contain products from **LIVE** catalogue but no stock required
`get_order_status`	Returns status of **LIVE** order	Returns dummy tracking data
`get_stock_level`	Returns stock level of **LIVE** product	Returns stock level of **LIVE** product
`get_all_skus_with_overnight_stock_levels`	Returns overnight stock levels of all **LIVE** products	Returns overnight stock levels of all **LIVE** products
`create_new_product`	New product created in **LIVE** product catalogue	New product created in **TEST** product catalogue
`get_product_by_sku`	Returns product from **LIVE** catalogue	Returns product from **TEST** catalogue
`create_new_purchase_order`	New PO created in **LIVE** PO table, must contain products from **LIVE** catalogue	New PO created in **TEST** PO table, must contain products from **LIVE** catalogue

A note about Tracking

Note that from an order monitoring point of view, once in LIVE mode the system returns a real tracking code, but only once the order has reached the "Despatched" status and been manifested to the courier (which happens daily in the early evening). The despatched date/time remains null (or 01/01/0001: 00:00) until we have a confirmed first scan from the courier's depot (usually early morning hours).

So depending on how early you need to expose the tracking info to the client, you might choose to check the status periodically until the date is anything other than that default, then expose the tracking code, or simply check it late evening and just give them the tracking code and not worry about the precise date/time of despatch.

V1_9 API Classes and Properties (subject to change)

mw_api_credentials

username	String
password	String

mw_api_order

order_delivery_name	String (50)
order_delivery_address_1	String (50)
order_delivery_address_2	String (50)
order_delivery_town	String (50)
order_delivery_county	String (50)
order_delivery_postcode	String (50)
order_delivery_country	String (100)
order_delivery_carrier_code	String
order_lines	Array of type: mw_api_order_line
order_reference	String
order_special_instructions	String
order_delivery_carrier_code	String
order_email_address	String (100)
order_mobile_number	String (50)
order_vat_amount	Decimal
order_total	Decimal
order_mywrap	Boolean

mw_api_order_line

line_product_sku	String (50)
line_product_qty	Int32

mw_order_status (returned by the `get_order_status` function)

order_courier_tracking_code	String (50)
order_despatch_date	DateTime
order_despatch_status	String (e.g. "Despatched", "In Pick" etc)

mw_api_stock_item
(array of these is returned by the `get_all_skus_with_overnight_stock_levels` function)

product_sku	String (50)
overnight_stock_level	Integer

mw_api_product
(submitted to the `create_new_product` function)

product_sku	String (50)
product_barcode	String (50)
product_description	String (250)
product_height_in_mm	Decimal
product_width_in_mm	Decimal
product_depth_in_mm	Decimal

product_weight_in_grams	Decimal
product_pack_quantity	Integer
product_initial_stock_qty	Integer
product_initial_stock_arrival_date	Date
product_customs_value	Decimal

mw_api_purchase_order

(submitted to the `create_purchase_order` function)

purchase_order_ref	String (50)
purchase_order_special_instructions	String (250)
purchase_order_expected_arrival	Date
purchase_order_sku_items	Array of type: mw_api_purchase_order_sku_item

mw_api_purchase_order_sku_item

(array of these is required in the **mw_api_purchase_order** object)

product_sku	String (50)
quantity_on_po	Integer

Presumes that you have added the Webservice and that it has been referenced as “mywarehouse”

```
Sub Main()
        'test_create_new_order()
        'test_get_order_status()
        'test_get_stock_level()
        'test_get_all_skus_and_overnight_stock_levels()
        'test_create_new_product()
        'test_get_product_by_sku()
        'test_create_new_purchase_order()
End Sub

    Sub test_create_new_order()
        Try
            'create the mywarehouse webservice proxy
            Dim mywarehouse_proxy As New mywarehouse.mywarehouse_api

            'create a mywarehouse credentials object so we can log in
            Dim mywarehouse_credentials As New mywarehouse.mw_api_credentials

            'populate the credentials object with your own username and password supplied by mywarehouse
admin
            mywarehouse_credentials.username = "your_api_username"
            mywarehouse_credentials.password = "your_api_password"

            'create a new order object
            Dim new_order As New mywarehouse.mw_api_order

            'populate the order details

            new_order.order_reference = "ORDER REF"                       'your reference code for this order
            new_order.order_delivery_name = "Customer Name"               'the name of the customer
            new_order.order_delivery_address_1 = "Address Line 1"         'first line of the address
            new_order.order_delivery_address_2 = "Address Line 2"         'second line of the address
            new_order.order_delivery_town = "Address Town"                'the Town
            new_order.order_delivery_postcode = "Address Postcode"        'the Postcode
            new_order.order_delivery_county = "Address County"            'the County
            new_order.order_delivery_country = "United Kingdom"
                                      'the Country official short name in English as per ISO 3166-1
            new_order.order_special_instructions = "Gift Message"
                                      'Any Gift message for the Customer – will be printed on Despatch note
            new_order.order_delivery_carrier_code = "P"

                                          'the delivery method. Must be one of the following:
                                             '"C24" - for Next Day service
                                             '"CSAT" - for Courier Saturday Morning
                                             '"IC" - for International Courier
                                             '"P" - for RM 1st Class Postage
                                             '"RP" - for RM Recorded Postage
                                             '"TP" - for RM Tracked Postage
                                             '"SP" - for RM Special Delivery Postage
                                             '"IP" - for RM International Postage
                                             '"IPS" - for RM International Postage Signed For
                                             '"IPT" - for RM International Postage
                                             '"COLL" - for Personal Collection from the Warehouse


            new_order.order_email_address = "useareal@email.address"       'the customer’s email address
            new_order.order_mobile_number = "01234 567890"                 'the customer’s mobile number
            new_order.order_mywrap = False                                 'special wrapping on order?


            new_order.order_vat_amount = 0.0                  'the VAT element only of the total order value
            new_order.order_total = 0.0                       'the total order value inc VAT

            'details are done, now add the order line items one by one to the order_lines collection field of
            the above order

            Dim order_line_collection As New ArrayList 'an array that we will fill with order lines
            'these are the orderlines - note that you will almost certainly want to create a for / next loop
            'that steps through your order lines. The important thing is that each line is added to the
            'order_line_collection array

            'xxxxxxx START OF ORDER LINES xxxxxxx

            'For each of your items, do this
            Dim new_order_line As New mywarehouse.mw_api_order_line

            new_order_line.line_product_sku = "A_SKU_OF_YOURS "   'the SKU of the product to add
            new_order_line.line_product_qty = 1               'the quantity of that item to add

            'add the line
            order_line_collection.Add(new_order_line)

            'Next item, loop back until all items are added
            'xxxxxxx END OF ORDER LINES xxxxxxx

            'all done, add the array of items to the order
            new_order.order_lines = order_line_collection.ToArray(GetType(mywarehouse.mw_api_order_line))

            'when the order is submitted, an Integer value is returned that is your unique order reference so
            'create an object to populate with this number
            Dim returned_order_number As Int32

            'submit the order to mywarehouse and receive the order confirmation number in return
            Console.WriteLine("Submitting order...")
            returned_order_number = mywarehouse_proxy.create_new_order(mywarehouse_credentials, new_order)

            Console.WriteLine("All done - new order ID is " & CStr(returned_order_number))


            'clean up by releasing the proxy object
            mywarehouse_proxy.Dispose()

        Catch ex As Exception
            'something went wrong - so show the error message
            Console.WriteLine("There was a problem - " & ex.Message)
        End Try

        Console.ReadLine()
    End Sub
    Sub test_get_order_status()

        Try
            'create the mywarehouse webservice proxy
            Dim mywarehouse_proxy As New mywarehouse.mywarehouse_api

            'create a mywarehouse credentials object so we can log in
            Dim mywarehouse_credentials As New mywarehouse.mw_api_credentials
            'populate the credentials object with your own username and password supplied by mywarehouse
admin
            mywarehouse_credentials.username = "your_api_username"
            mywarehouse_credentials.password = "your_api_password"

            'this is the order_id to check - should be a valid order_number received
            'back from MMS when an order was placed using create_new_order
            Dim order_unique_id As Int32
            'as a test, set it to a known order
            order_unique_id = 0000000

            Dim returned_order_status As New mywarehouse.mw_order_status
            'get the order status
            returned_order_status = mywarehouse_proxy.get_order_status(mywarehouse_credentials,
order_unique_id)
            'the order status function returns an Array:
            'despatched status (string) - while you are a TEST user this will show test_despatch_status
            Dim returned_despatched_status As String = returned_order_status.order_despatch_status
            'courier tracking code (string) - while you are a TEST user this will show test_tracking_code
            Dim returned_courier_tracking_code As String = returned_order_status.order_courier_tracking_code
            'date of despatch (datetime) - while you are a TEST user this will show the date and time now
            Dim returned_despatch_date As Date = returned_order_status.order_despatch_date

            Console.WriteLine("order: " & order_id_to_check & ", status: " & returned_despatched_status & _
                                ", courier code: " & returned_courier_tracking_code & ", despatched: " &
returned_despatch_date.ToString)
            'clean up by releasing the proxy object
            mywarehouse_proxy.Dispose()

        Catch ex As Exception
            'something went wrong - so show the error message
            Console.WriteLine("There was a problem - " & ex.Message)
        End Try
        Console.ReadLine()
    End Sub
    Sub test_get_stock_level()

        Try
            'create the mywarehouse webservice proxy
            Dim mywarehouse_proxy As New mywarehouse.mywarehouse_api

            'create a mywarehouse credentials object so we can log in
            Dim mywarehouse_credentials As New mywarehouse.mw_api_credentials

            'populate the credentials object with your own username and password supplied by mywarehouse
admin
            mywarehouse_credentials.username = "your_api_username"
            mywarehouse_credentials.password = "your_api_password"

            Dim sku_to_check
            sku_to_check = "A_SKU_OF_YOURS"    'populate this with the sku you wish to check

            'this will be populated with the returned stock level
            Dim returned_stock_level As Int32
            'check stock
            returned_stock_level = mywarehouse_proxy.get_stock_level(mywarehouse_credentials, sku_to_check)

            Console.WriteLine("Stock level of " & sku_to_check & " is: " & returned_stock_level)

            'clean up by releasing the proxy object
            mywarehouse_proxy.Dispose()

        Catch ex As Exception

            'something went wrong - so show the error message
            Console.WriteLine("There was a problem - " & ex.Message)
        End Try

        Console.ReadLine()
    End Sub


    Sub test_get_all_skus_and_overnight_stock_levels()
        Try
            'create the mywarehouse webservice proxy
            Dim mywarehouse_proxy As New mywarehouse.mywarehouse_api

            'create a mywarehouse credentials object so we can log in
            Dim mywarehouse_credentials As New mywarehouse.mw_api_credentials

            'populate the credentials object with your own username and password supplied by mywarehouse
admin
            mywarehouse_credentials.username = "your_api_username"
            mywarehouse_credentials.password = "your_api_password"

            'prepare an array to receive the items
            Dim sku_array As Array
            sku_array = mywarehouse_proxy.get_all_skus_with_overnight_stock_levels(mywarehouse_credentials)

            'iterate through the array, each item is an instance of mw_api_stock_item
            For Each product_on_mms As mywarehouse.mw_api_stock_item In sku_array
                Console.WriteLine(product_on_mms.product_sku & ", " & product_on_mms.overnight_stock_level)
            Next

            'clean up by releasing the proxy object
            mywarehouse_proxy.Dispose()

        Catch ex As Exception

            'something went wrong - so show the error message
            Console.WriteLine("There was a problem - " & ex.Message)
        End Try
        Console.ReadLine()
    End Sub
```

```
Sub test_create_new_product()
        Try
            'create the mywarehouse webservice proxy
            Dim mywarehouse_proxy As New mywarehouse.mywarehouse_apiSoapClient

            'create a mywarehouse credentials object so we can log in
            Dim mywarehouse_credentials As New mywarehouse.mw_api_credentials

            'populate the credentials object with your own username and password supplied by mywarehouse
admin
            mywarehouse_credentials.username = "your_api_username"
            mywarehouse_credentials.password = "your_api_password"

            'create a new product object
            Dim new_product As New mywarehouse.mw_api_product

            'populate the product details
            new_product.product_sku = "A_NEW_SKU_OF_YOURS"
                                                     'the unique SKU(stock keeping unit) for this product
            new_product.product_barcode = "THE-BARCODE-01234”
                                                     'the unique barcode value for this product
                                                     'set this to “MYWAREHOUSE” for an autogenerated MMS
                      barcode
            new_product.product_description = "Description"
                                                     'text description of the product, max 250 chars
                                                     ‘(plain text only)

            new_product.product_height_in_mms = 10          'the height of the packed product in mm
            new_product.product_width_in_mms = 20           'the width of the packed product in mm
            new_product.product_depth_in_mms = 32           'the depth of the packed product in mm
            new_product.product_weight_in_grams = 200       'the weight of the packed product in grams
            new_product.product_pack_quantity = 1           'how many units are in this SKU? (usually 1)
            new_product.product_customs_value = 0.00        'what value do you want to declare for customs?

            new_product.product_initial_stock_qty = 0
            '** IMPORTANT! if you're intending to use the create_new_po function
            to make a combined PO for this and other SKUs later, leave this at 0

             'if this is a value greater than 0 then a Purchase Order will be
             created to alert the warehouse to the imminent arrival of some units of this one SKU.

            'note that NO ITEMS can be processed by Goods In at the warehouse without a PO


            new_product.product_initial_stock_arrival_date = DateAdd(DateInterval.Day, 3, Date.Today)
            'the estimated date on which the product is expected to arrive at the warehouse.
            'must be either today or in the future (i.e. not in the past!)

            'details are done, now submit this product and check that all went well:
            'a boolean (True/False) value is returned

            Dim returned_success As Boolean

            'submit the order to mywarehouse and receive the confirmation

            Console.WriteLine("Submitting product...")

            returned_success = mywarehouse_proxy.create_new_product(mywarehouse_credentials, new_product)

            If returned_success Then
                Console.WriteLine("All done - product was created successfully")
            Else
                Console.WriteLine("There was a problem creating the product")
            End If

            'clean up by releasing the proxy object
            mywarehouse_proxy = Nothing

        Catch ex As Exception
            'something went wrong - so show the error message
            Console.WriteLine("There was a problem - " & ex.Message)
        End Try

        Console.ReadLine()
    End Sub
    Sub test_get_product_by_sku()

        Try
            'create the mywarehouse webservice proxy
            Dim mywarehouse_proxy As New mywarehouse.mywarehouse_apiSoapClient

            'create a mywarehouse credentials object so we can log in
            Dim mywarehouse_credentials As New mywarehouse.mw_api_credentials

            'populate the credentials object with your own username and password supplied by mywarehouse
admin
            mywarehouse_credentials.username = "your_api_username"
            mywarehouse_credentials.password = "your_api_password"

            'this is the SKU to check - must be an existing SKU.
            'If you're in TEST mode, this SKU must be in your TEST catalogue, or in your LIVE catalogue if
            you're LIVE

            Dim sku_to_check As String
            sku_to_check = " A_NEW_SKU_OF_YOURS"

            Dim returned_product As New mywarehouse.mw_api_product
            'get the product info
            returned_product = mywarehouse_proxy.get_product_by_sku(mywarehouse_credentials, sku_to_check)

            'the function returns an mw_api_product object, let's display a couple of fields:
            Console.WriteLine("Found - Product SKU: " & returned_product.product_sku & ", Barcode: " &
returned_product.product_barcode)

            'clean up by releasing the proxy object
            mywarehouse_proxy = Nothing

        Catch ex As Exception
            'something went wrong - so show the error message
            Console.WriteLine("There was a problem - " & ex.Message)
        End Try
        Console.ReadLine()
    End Sub
    Sub test_create_new_purchase_order()

        Try
            'create the mywarehouse webservice proxy
            Dim mywarehouse_proxy As New mywarehouse.mywarehouse_apiSoapClient

            'create a mywarehouse credentials object so we can log in
            Dim mywarehouse_credentials As New mywarehouse.mw_api_credentials

            'populate the credentials object with your own username and password supplied by mywarehouse
admin
            mywarehouse_credentials.username = "your_api_username"
            mywarehouse_credentials.password = "your_api_password"

            'create a new purchase order object
            Dim new_purchase_order As New mywarehouse.mw_api_purchase_order

            'populate the purchase order details
            new_purchase_order.purchase_order_ref = "NEW_PO_REFERENCE" 'the PO Reference (your ref) for this
purchase order
            new_purchase_order.purchase_order_expected_arrival = Today
            'the estimated date on which the product is expected to arrive at the warehouse.
            'must be either today or in the future (i.e. not in the past!)

            Dim sku_to_add As New mywarehouse.mw_api_purchase_order_sku_item
            sku_to_add.product_sku = "A_LIVE_SKU_OF_YOURS"
            sku_to_add.quantity_on_po = 4

            Dim sku_to_add_2 As New mywarehouse.mw_api_purchase_order_sku_item
            sku_to_add_2.product_sku = "ANOTHER_LIVE_SKU_OF_YOURS"
            sku_to_add_2.quantity_on_po = 2

            Dim sku_array_to_add As New mywarehouse.ArrayOfMw_api_purchase_order_sku_item
            sku_array_to_add.Add(sku_to_add)
            sku_array_to_add.Add(sku_to_add_2)

            new_purchase_order.purchase_order_sku_items = sku_array_to_add

            'details are done, now submit this purchase order and check that all went well:
            'a boolean (True/False) value is returned

            Dim returned_success As Boolean

            'submit the purchase order to mywarehouse and receive the confirmation

            Console.WriteLine("Submitting purchase order...")

            returned_success = mywarehouse_proxy.create_new_purchase_order(mywarehouse_credentials,
new_purchase_order)

            If returned_success Then
                Console.WriteLine("All done - purchase order was created successfully")
            Else
                Console.WriteLine("There was a problem creating the purchase order")
            End If

            'clean up by releasing the proxy object
            mywarehouse_proxy = Nothing

        Catch ex As Exception
            'something went wrong - so show the error message
            Console.WriteLine("There was a problem - " & ex.Message)
        End Try

        Console.ReadLine()
    End Sub
```

Version Log

V1 – 01.07.09

Original API Doc

V1_2 - 02.12.09

Added `get_all_skus_with_overnight_stock_levels`

Added **mw_api_stock_item**

Added `order_vat_amount` to **mw_api_order**

Added `order_total` to **mw_api_order**

V1_3 - 10.12.09

No changes, just version update to match code examples doc

V1_4 - 02.10.10

Added `order_email_address` to **mw_api_order**

Added `order_mobile_number` to **mw_api_order**

(to support Premium courier options)

V1_5 -21.01.11

Added `order_mywrap` to **mw_api_order**

(to support wrapping service options)

V1_6 -23.03.11

Amended description of `get_order_status` call to show name of input parameter as "order_unique_id" (was "order_id")

V1_7-12.05.11

Added `create_new_product`

Added `get_product_by_sku`

Added **mw_api_product**

Added `create_new_purchase_order`

Added **mw_api_purchase_order**

Added **mw_api_purchase_order_sku_item**

V1_8-12.10.12

Added `product_customs_value` to **mw_api_product**

V1_9-21.08.15

Updated Code Examples

V1_9-21.08.15

Revised branding