MADE2MANAGE

Web API – Reference Manual

Version: 7.51 SP6

January 2020

Contents

About this Manual

Scope

This manual provides the information you need to use the **Made2Manage Web API** and also sample requests supported by the **API**.

Audience

This manual is intended for individuals who want to use APIs to interact with the Made2Manage application programmatically.

Before you Begin

Two graphics are special:

Tip: This icon indicates a tip, which provides critical information that you must know or points you to related information on other pages.

Note: This icon indicates a note, which explains installation program output or system output you must consider before continuing.

Send questions, comments, or suggestions related to documentation to Info.Aptean@aptean.com.

Overview

The **Made2Manage Web API** is an application program interface intended for individuals who need to access, manipulate and update the **Made2Manage** data outside of the standard user interface and administrative tools.

Note: Ensure to enable the Auto Redirect feature within the development tool you are using to consume the M2M Web API.

Note: You will need a new activation code to work with the M2M Web API.

In this manual, we will cover the CRUD (Create, Read, Update, Delete) operations on the Object permitted in Made2Manage Web API.

Quick Checklist

	With this manual, you will be able to…
☐	Learn about the request headers permitted in the API.
☐	Learn categories of status codes available in the API.
☐	Learn about the CRUD operations on the Object permitted in the API.

Working with the API

Request Headers

Header	Description
CompanyID (Optional)	ID of the company whose information will be accessed/modified via API.
ClientName (Optional)	Client who has been granted access to API.
Authorization	Access token required for authentication.
Content-Type	For POST and PUT requests to specify the type of information passed in the request body.

Note: **Company ID** and **Client Name** are the fields configured on the **APICONFIG** screen.

HTTP Status Codes

HTTP defines forty standard status codes that can be used to convey the results of a client’s request. The status codes are divided into the five categories presented below.

Category	Description
1xx	**Informational** - Communicates transfer protocol-level information.
2xx	**Success** - Indicates that the client’s request was accepted successfully.
3xx	**Redirection** - Indicates that the client must take some additional action in order to complete their request.
4xx	**Client Error** - This category of error status codes points the finger at clients.
5xx	**Server Error** - The server takes responsibility for these error status codes.

Note: Along with status codes, a friendly error message is returned in the response of a Made2Manage Web API which provides more details.

CRUD Operations

In this section, we will cover each of the CRUD (Create, Read, Update, Delete) operations by providing multiple examples ranging from a simple request to a complex request.

API Endpoints

In the Postman tool that is used for interacting with HTTP APIs, the API endpoint (also known as Request) is an URL that specifies the location from which APIs can access the resources to perform the required function.

In an API endpoint URL, the request levels and their associated fields/data in Made2Manage is explained with an example in the following table:

Endpoint: http://<<server>>/<contextpath>/api/<ObjectName>/<Value>/<FriendlyName>/<Value>

Example Endpoint: http://<server>/M2MWebAPI/api/SalesOrder/000584/SalesOrderLineItems/2/

Endpoint Request Level	Request Parameter	Data from Made2Manage
First Level	**Object Name** Ex: Sales Order	**APICONFIG** screen > **Object Name** field
	Value: Ex: 000584	Sales Order Number
Second Level	**Friendly Name** Ex: SalesOrderLineItems	**APICONFIG** screen > **Schema Information** tab > **Friendly Name** column
	Value: Ex: 2	Sales Order Line Item Number

Read Operation

Request Type	**GET**
Purpose	Retrieves the details of the Object based on the input parameters passed along the request.
Endpoint	https://<<server>>/<contextpath>/api/<<Object Name>>/<<input parameters>>

Quick Checklist

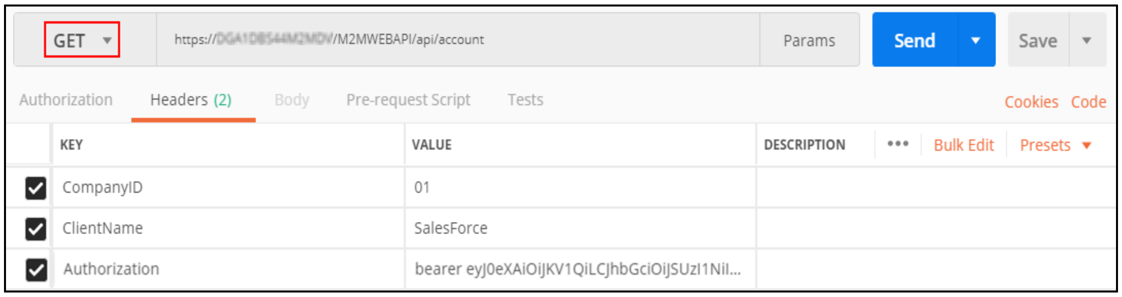
	In this chapter, you will learn about...
☐	GET - *Simple Request:*
☐	GET - *Simple Request with Pagination:*
☐	GET - *Mid Complex Request:*
☐	GET - *Complex Request:*

Simple Request:

The illustration lists the customer collection with brief data.

Endpoint: https://<<server>>/M2MWebAPI/api/Account

Figure 1: GET - Simple Request

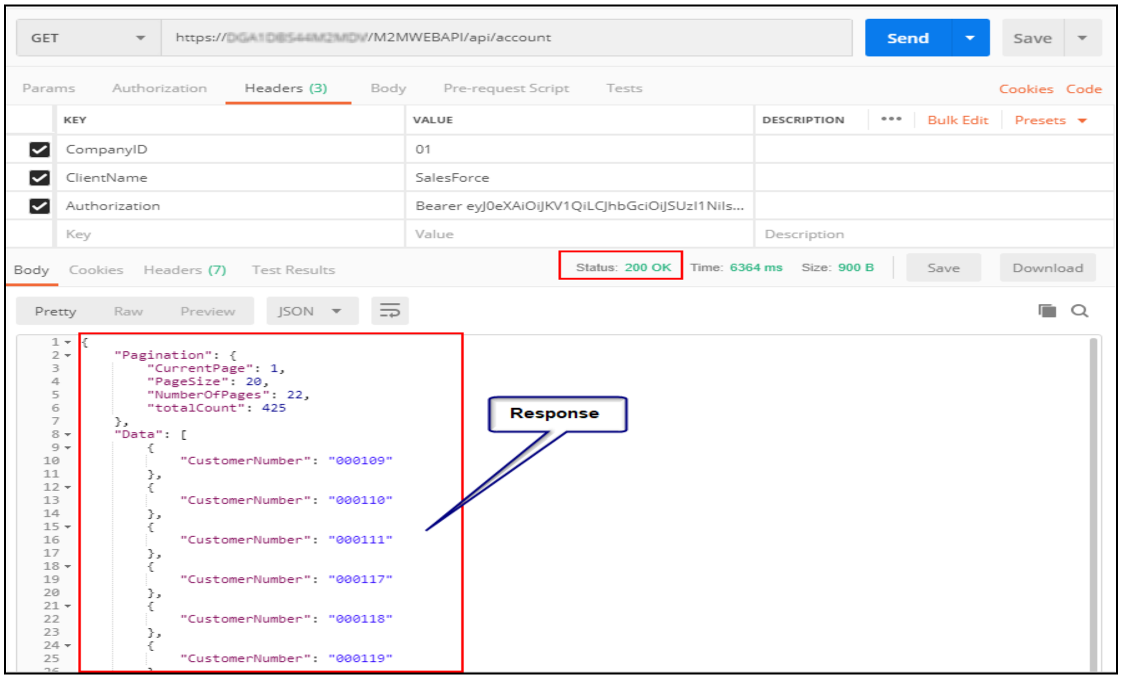


https://[redacted]/M2MWEBAPI/api/account

	KEY	VALUE	DESCRIPTION
✓	CompanyID	01	
✓	ClientName	SalesForce	
✓	Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the requested data in JSON format.

Figure 2: GET - Simple Response



GET https://[redacted]/M2MWEBAPI/api/account

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Nils...	
Key	Value	Description

Body | Cookies | Headers (7) | Test Results — Status: 200 OK Time: 6364 ms Size: 900 B

```
{
    "Pagination": {
        "CurrentPage": 1,
        "PageSize": 20,
        "NumberOfPages": 22,
        "totalCount": 425
    },
    "Data": [
        {
            "CustomerNumber": "000109"
        },
        {
            "CustomerNumber": "000110"
        },
        {
            "CustomerNumber": "000111"
        },
        {
            "CustomerNumber": "000117"
        },
        {
            "CustomerNumber": "000118"
        },
        {
            "CustomerNumber": "000119"
```

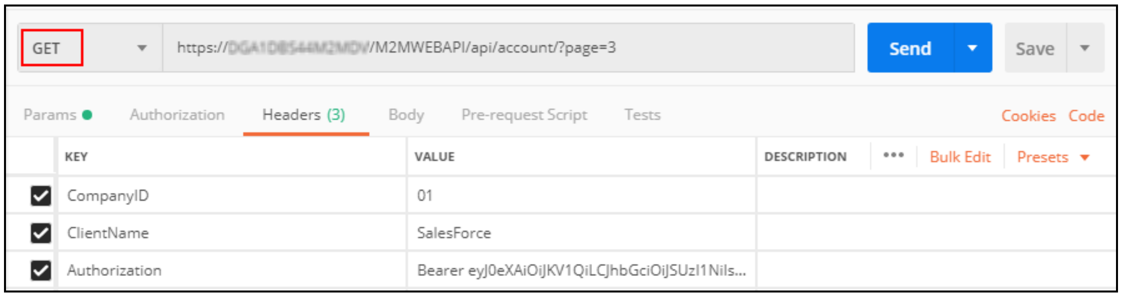
Simple Request with Pagination:

Pagination helps to handle large datasets and responses. You can navigate to a specific page and see the total number of pages and then progress through the total.

The illustration lists the customer collection with pagination.

Endpoint: https://<<server>>/M2MWebAPI/api/Account/pagination

Figure 3: GET - Simple Request with pagination

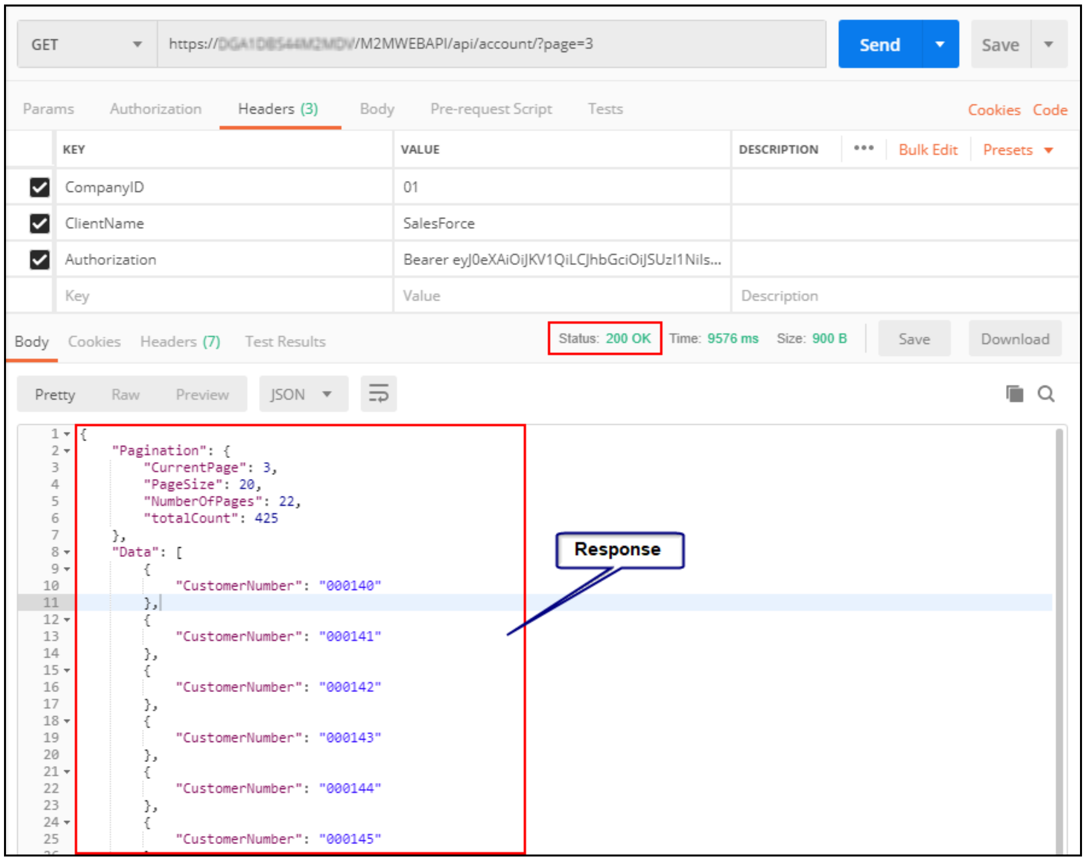


GET https://[redacted]/M2MWEBAPI/api/account/?page=3

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Nils...	

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the requested data in JSON format.

Figure 4: GET - Simple Response with Pagination



https://[redacted]/M2MWEBAPI/api/account/?page=3

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Nils...	
Key	Value	Description

Status: 200 OK Time: 9576 ms Size: 900 B

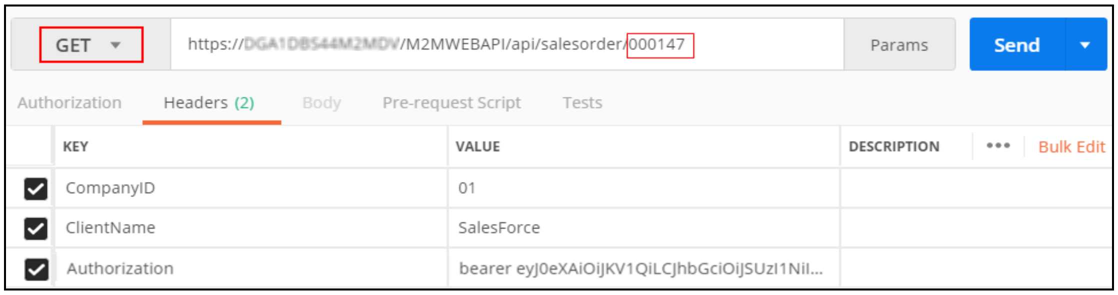
```
{
    "Pagination": {
        "CurrentPage": 3,
        "PageSize": 20,
        "NumberOfPages": 22,
        "totalCount": 425
    },
    "Data": [
        {
            "CustomerNumber": "000140"
        },
        {
            "CustomerNumber": "000141"
        },
        {
            "CustomerNumber": "000142"
        },
        {
            "CustomerNumber": "000143"
        },
        {
            "CustomerNumber": "000144"
        },
        {
            "CustomerNumber": "000145"
```

Mid Complex Request:

The illustration lists details of a specific sales order (`000147`) and all of its related entities.

Endpoint: https://<<server>>/M2MWebAPI/api/SALESORDER/<<sales order number>>

Figure 5: GET - Mid Complex Request

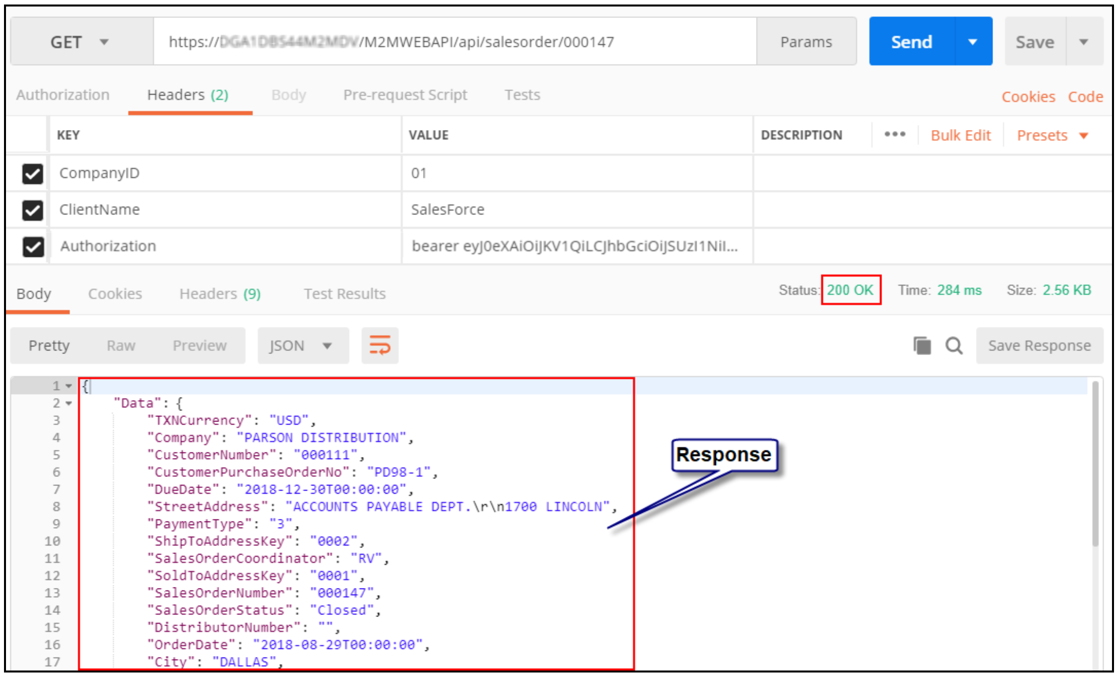


https://[redacted]/M2MWEBAPI/api/salesorder/000147

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Nil...	

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the requested data in JSON format.

Figure 6: GET - Mid Complex Response



https://[redacted]/M2MWEBAPI/api/salesorder/000147

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Nil...	

Status: 200 OK Time: 284 ms Size: 2.56 KB

```
{
    "Data": {
        "TXNCurrency": "USD",
        "Company": "PARSON DISTRIBUTION",
        "CustomerNumber": "000111",
        "CustomerPurchaseOrderNo": "PD98-1",
        "DueDate": "2018-12-30T00:00:00",
        "StreetAddress": "ACCOUNTS PAYABLE DEPT.\r\n1700 LINCOLN",
        "PaymentType": "3",
        "ShipToAddressKey": "0002",
        "SalesOrderCoordinator": "RV",
        "SoldToAddressKey": "0001",
        "SalesOrderNumber": "000147",
        "SalesOrderStatus": "Closed",
        "DistributorNumber": "",
        "OrderDate": "2018-08-29T00:00:00",
        "City": "DALLAS",
```

Complex Request:

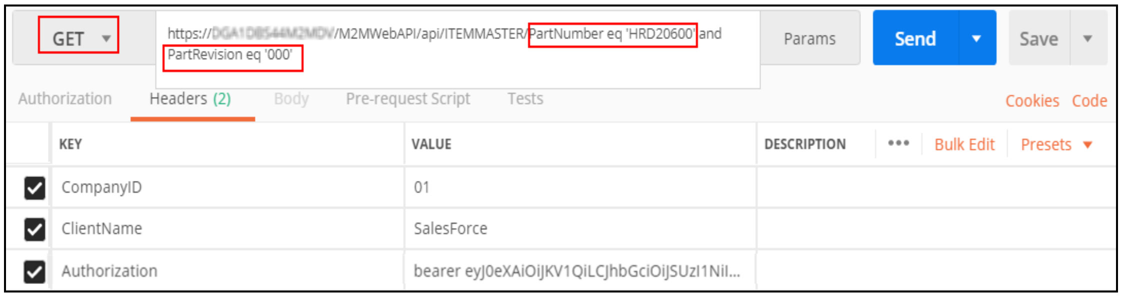
If there is a scenario where the response is obtained by passing multiple composite keys in the request, then passing one primary attribute would not suffice and we need to pass all of the composite keys to fetch specific record.

This can be achieved by passing filter conditions. For more information, see *Read Operation*

The illustration demonstrates the request to fetch a specific part (`HRD20600`) and a specific revision (`000`) from the item master which has parts created with multiple revisions.

Endpoint: https://<<server>>/M2MWebAPI/api/ITEMMASTER/PartNumber eq '<<part number>>' and PartRevision eq '<<revision number>>'

Figure 7: GET - Complex Request

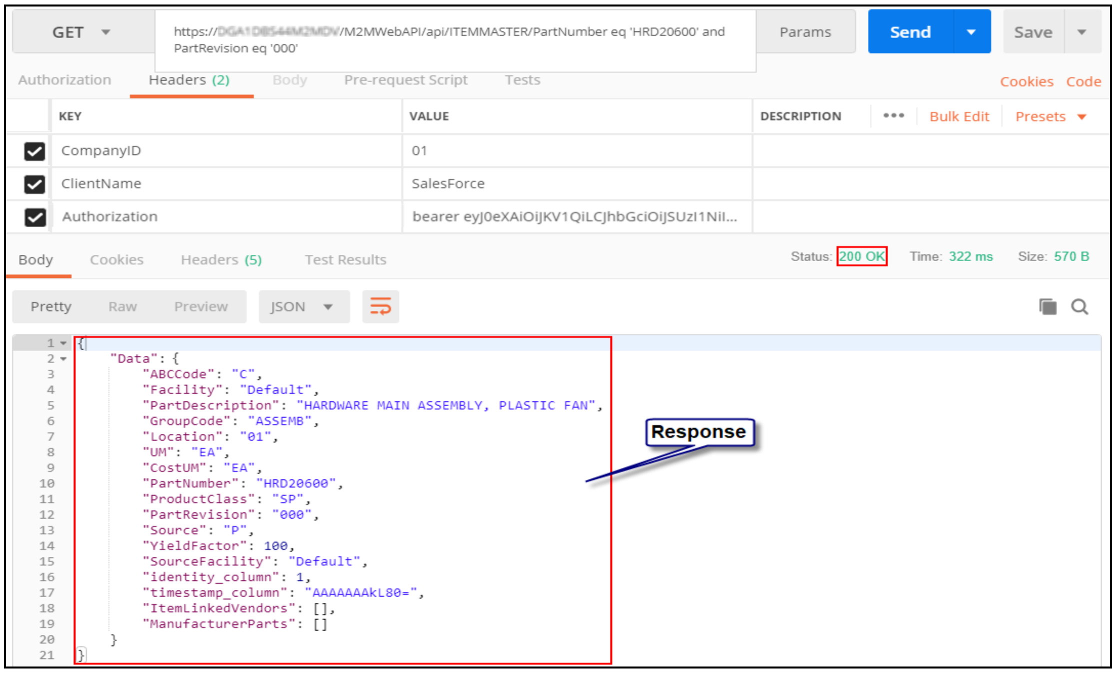


https://[redacted]/M2MWebAPI/api/ITEMMASTER/PartNumber eq 'HRD20600' and PartRevision eq '000'

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Nil...	

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the requested data in JSON format.

Figure 8: GET - Complex Response



```
GET https://[redacted]/M2MWebAPI/api/ITEMMASTER/PartNumber eq 'HRD20600' and PartRevision eq '000'
```

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Status: 200 OK Time: 322 ms Size: 570 B

```
{
    "Data": {
        "ABCCode": "C",
        "Facility": "Default",
        "PartDescription": "HARDWARE MAIN ASSEMBLY, PLASTIC FAN",
        "GroupCode": "ASSEMB",
        "Location": "01",
        "UM": "EA",
        "CostUM": "EA",
        "PartNumber": "HRD20600",
        "ProductClass": "SP",
        "PartRevision": "000",
        "Source": "P",
        "YieldFactor": 100,
        "SourceFacility": "Default",
        "identity_column": 1,
        "timestamp_column": "AAAAAAAkL80=",
        "ItemLinkedVendors": [],
        "ManufacturerParts": []
    }
}
```

Filtering Operation

Note: To retrieve items that contain special characters which are restricted by IIS, use encoding in the Query String. Refer to Filtering with Special Characters

Request Type	**GET**
Purpose	The GET request provides detailed information of the Object based on the object ID passed along the request. We can extract specific information from the response using filter conditions.
Endpoint	https://<<server>>/<contextpath>/api/<<Object Name>>/<<filter conditions>>

The supported operands are:

- IsEqualTo (**eq**)
- NotEqualTo (**ne**)
- IsGreaterThan (**gt**)
- IsLessThan (**lt**)
- GreaterThanOrEqualTo (**ge**)
- LessThanOrEqualTo (**le**)

The supported Odata functions and their corresponding URLs are listed in the following table:

Odata Functions	Corresponding URLs
`substringof`	filter=substringof('PARSON',Company) eq true
`startswith`	filter=startswith(Company,'parson') eq true
`endswith`	filter=endswith(Company,'equipment') eq true
`trim`	filter=trim(Company) eq 'parson distribution'
`day`	filter=day(DueDate) eq '18'
`month`	filter=month(DueDate) eq '12'
`year`	filter=year(DueDate) eq '2018'

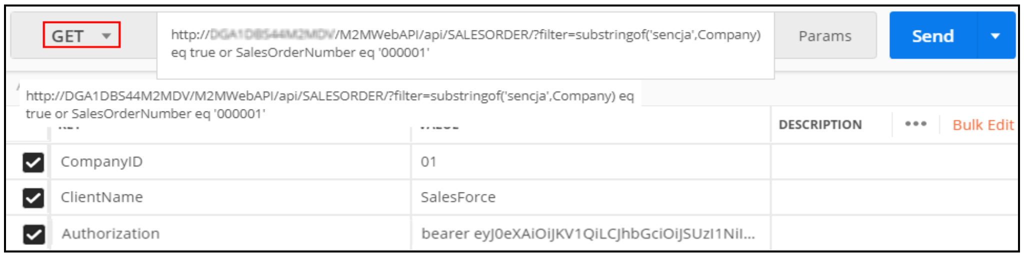
Example: The screen shot illustrates a request to get details regarding the sales order with filtering conditions: *substringof('sencja',Company) eq true or SalesOrderNumber eq '000001'*

The *substringof* function filters records, which match the text that you provide.

For filtering, OData specification is being used. Supported operand is *IsEqualTo (eq)*.

Endpoint: https://<<server>>/M2MWebAPI/api/SALESORDER/?filter=<<filter conditions>>

Figure 9: Filtered Request



GET

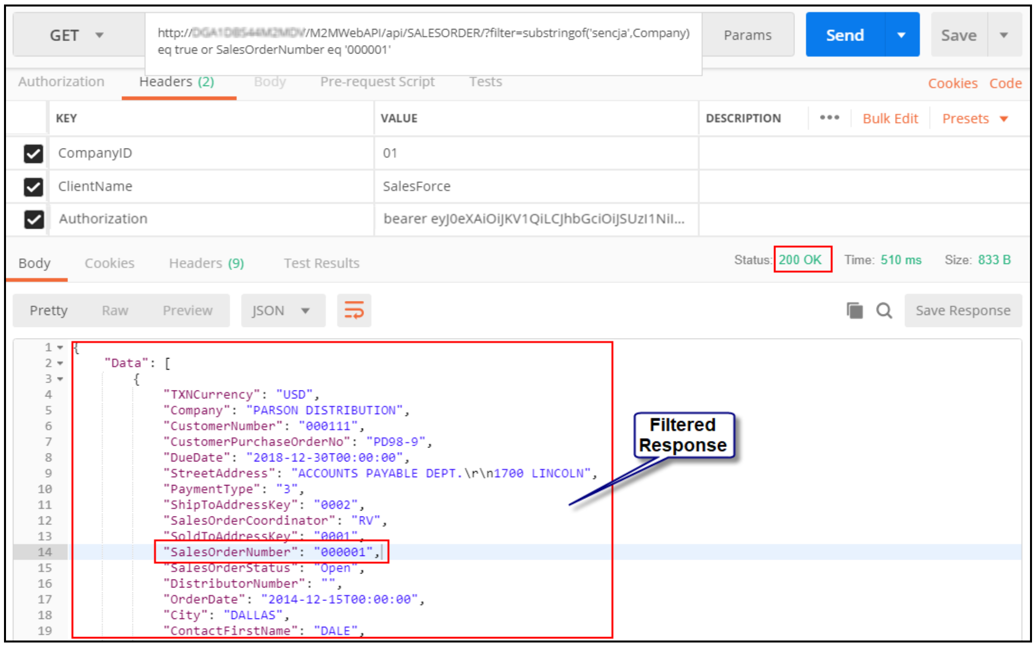
http://[redacted]/M2MWebAPI/api/SALESORDER/?filter=substringof('sencja',Company) eq true or SalesOrderNumber eq '000001'

http://DGA1DBS44M2MDV/M2MWebAPI/api/SALESORDER/?filter=substringof('sencja',Company) eq true or SalesOrderNumber eq '000001'

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

The result will have a HTTP **200** code and the response body contains the filtered information of sales order.

Figure 10: Filtered Response



GET

http://[redacted]/M2MWebAPI/api/SALESORDER/?filter=substringof('sencja',Company) eq true or SalesOrderNumber eq '000001'

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Status: 200 OK Time: 510 ms Size: 833 B

```
{
    "Data": [
        {
            "TXNCurrency": "USD",
            "Company": "PARSON DISTRIBUTION",
            "CustomerNumber": "000111",
            "CustomerPurchaseOrderNo": "PD98-9",
            "DueDate": "2018-12-30T00:00:00",
            "StreetAddress": "ACCOUNTS PAYABLE DEPT.\r\n1700 LINCOLN",
            "PaymentType": "3",
            "ShipToAddressKey": "0002",
            "SalesOrderCoordinator": "RV",
            "SoldToAddressKey": "0001",
            "SalesOrderNumber": "000001",
            "SalesOrderStatus": "Open",
            "DistributorNumber": "",
            "OrderDate": "2014-12-15T00:00:00",
            "City": "DALLAS",
            "ContactFirstName": "DALE",
```

Filtering with Special Characters

Special characters must be encoded in the Query String while they are passed with the URL. Spaces, dashes and other special characters could be common in the data that is passing through the M2M API in our filters.

Consider the following table for commonly used special characters:

Character	From Windows-1252	From UTF-8
Space	%20	%20
!	%21	%21
"	%22	%22
#	%23	%23
$	%24	%24
%	%25	%25
&	%26	%26
'	%27	%27
(	%28	%28
)	%29	%29
*	%2A	%2A
+	%2B	%2B
,	%2C	%2C
-	%2D	%2D
.	%2E	%2E
/	%2F	%2F
:	%3A	%3A
;	%3B	%3B
<	%3C	%3C
>	%3E	%3E
=	%3D	%3D
?	%3F	%3F
@	%40	%40
[	%5B	%5B

Character	From Windows-1252	From UTF-8
\	%5C	%5C
]	%5D	%5D
^	%5E	%5E
`	%60	%60
{	%7B	%7B
}	%7D	%7D
|	%7C	%7C

Note: For more information on URL encoding, refer https://www.urlencoder.org

Example: To filter the customer record for COHEN # within the M2M,

Invalid Url (Produces errors):

http://localhost/M2MWebAPI/api/ACCOUNT/?filter=startswith(Company,'COHEN #') eq true

Valid Url (Produces expected data):

http://localhost/M2MWebAPI/api/ACCOUNT/?filter=startswith(Company,'COHEN%20%23') eq true

To filter items with special characters which also contain Supported parameters in them, use the following key words in the String:

- parentid - To access primary key.
- childEntityName - To access Child by entity name.
- childEntityId - To access child object by the ID.
- grandChildEntityName - To access grandchild by entity name.
- grandChildEntityId -To access grandchild by the ID.

Example:

http://localhost/m2mwebApi/api/STANDARDBOM/?parentID=PartNumber eq 'gun-powder%201%221%27%20%26%202%2F3'&childEntityName=StandardBillOfMaterial

Note: These Supported parameters are separated by an '&' between them.

Note: To include special characters in the JSON Body, the characters must be Escaped before being used in the String. For more information, refer https://www.freeformatter.com/json-escape.html

Create Operation

Request Type	**POST**
Purpose	Inserts a new record under the Object using the details passed along the request body.
Endpoint	https://<<server>>/<contextpath>/api/<<Object Name>>

Quick Checklist

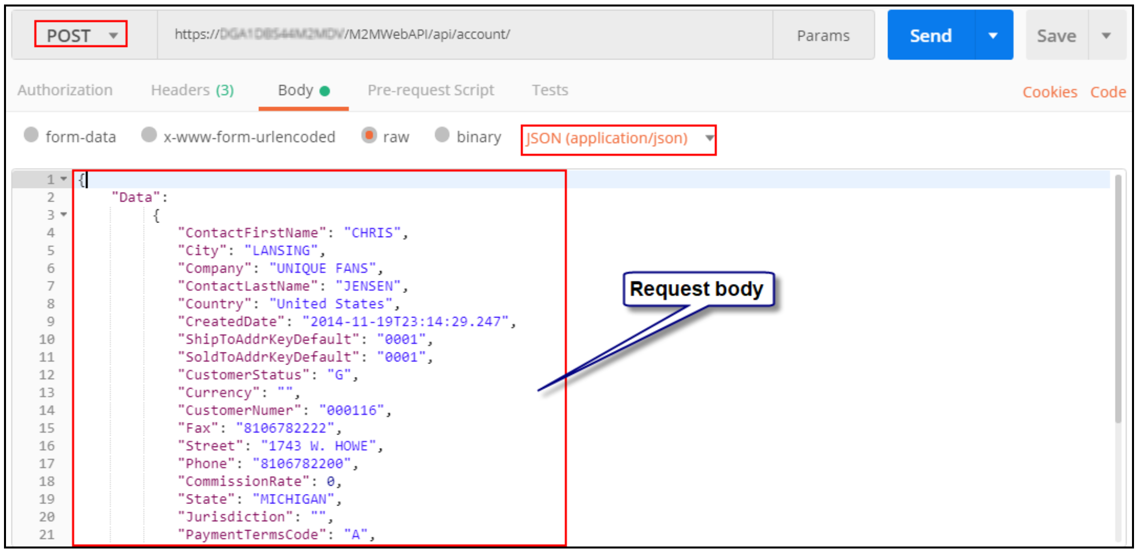
	In this chapter, you will learn about...
☐	POST - *Simple Request with Example*
☐	POST - *Mid Complex Request:*
☐	POST - *Complex Request:*

Simple Request with Example

Creating an account by passing a single entity.

Endpoint: https://<<server>>/M2MWebAPI/api/Account

Figure 11: POST - Simple Request



https://[redacted]/M2MWebAPI/api/account/

JSON (application/json)

```
{
    "Data":
        {
            "ContactFirstName": "CHRIS",
            "City": "LANSING",
            "Company": "UNIQUE FANS",
            "ContactLastName": "JENSEN",
            "Country": "United States",
            "CreatedDate": "2014-11-19T23:14:29.247",
            "ShipToAddrKeyDefault": "0001",
            "SoldToAddrKeyDefault": "0001",
            "CustomerStatus": "G",
            "Currency": "",
            "CustomerNumer": "000116",
            "Fax": "8106782222",
            "Street": "1743 W. HOWE",
            "Phone": "8106782200",
            "CommissionRate": 0,
            "State": "MICHIGAN",
            "Jurisdiction": "",
            "PaymentTermsCode": "A",
```

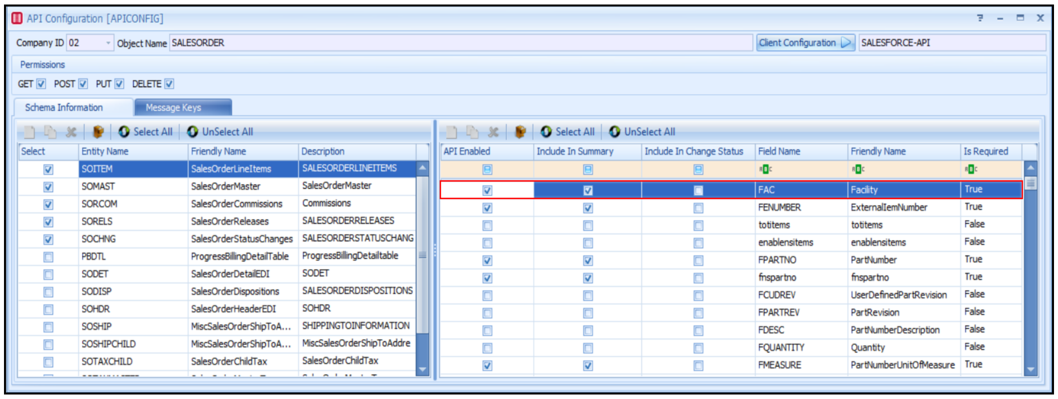
Upon successful processing of the request, the HTTP response code **201** is returned and the response body contains the data of the newly created account in JSON format.

Example:

To create a customer with API

1. Open the **APICLIENT** screen and create a new client. If you want to use an existing client record, then go to step-2.
2. Open the **APICONFIG** screen and click **New**. If you already have configured the **APICONFIG** screen to enable Web API access to remote Web applications (clients or third party applications) for the required Business object (Eg. Sales Order, Account, Accounts Payable) and provided appropriate permissions, then go to step 7.
3. Select the company number and Object Name (Account).
4. Specify the appropriate permissions (GET/POST/PUT/DELETE) based on the Object Name.
5. Click **Client Configuration** and select the **Client Name**.
6. Click **Save**.
7. In the **Schema Information** tab, select the check box to include the entities associated with the Object in the API response. In the right pane, select the check box to include the fields associated with the entity in the API response.

Figure 12: APICONFIG screen



Company ID 02 Object Name SALESORDER Client Configuration SALESFORCE-API

Permissions

GET ✓ POST ✓ PUT ✓ DELETE ✓

Select	Entity Name	Friendly Name	Description
✓	SOITEM	SalesOrderLineItems	SALESORDERLINEITEMS
✓	SOMAST	SalesOrderMaster	SalesOrderMaster
✓	SORCOM	SalesOrderCommissions	Commissions
✓	SORELS	SalesOrderReleases	SALESORDERRELEASES
✓	SOCHNG	SalesOrderStatusChanges	SALESORDERSTATUSCHANG
☐	PBDTL	ProgressBillingDetailTable	ProgressBillingDetailtable
☐	SODET	SalesOrderDetailEDI	SODET
☐	SODISP	SalesOrderDispositions	SALESORDERDISPOSITIONS
☐	SOHDR	SalesOrderHeaderEDI	SOHDR
☐	SOSHIP	MiscSalesOrderShipToA...	SHIPPINGTOINFORMATION
☐	SOSHIPCHILD	MiscSalesOrderShipToA...	MiscSalesOrderShipToAddre
☐	SOTAXCHILD	SalesOrderChildTax	SalesOrderChildTax

API Enabled	Include In Summary	Include In Change Status	Field Name	Friendly Name	Is Required
✓	✓	☐	FAC	Facility	True
✓	✓	☐	FENUMBER	ExternalIemNumber	True
☐	☐	☐	totitems	totitems	False
☐	☐	☐	enablensitems	enablensitems	False
✓	✓	☐	FPARTNO	PartNumber	True
✓	✓	☐	fnspartno	fnspartno	True
☐	☐	☐	FCUDREV	UserDefinedPartRevision	False
☐	☐	☐	FPARTREV	PartRevision	False
☐	☐	☐	FDESC	PartNumberDescription	False
☐	☐	☐	FQUANTITY	Quantity	False
✓	✓	☐	FMEASURE	PartNumberUnitOfMeasure	True

The custom entities or fields will display on the M2M business objects in the **APICONFIG** screen as explained below.

Example: When you customize the SO screen and select to add a new Entity (right click on the Sales Order, not one of the tables under it in SCRMNT properties), it is listed under the Sales Order and is not extending a standard table. It is creating a brand new table with your fields. When you look at **APICONFIG** for Sales Order, you would see the new entity listed in the left pane with the tables and its associated fields in the right pane to select and include in your API calls.

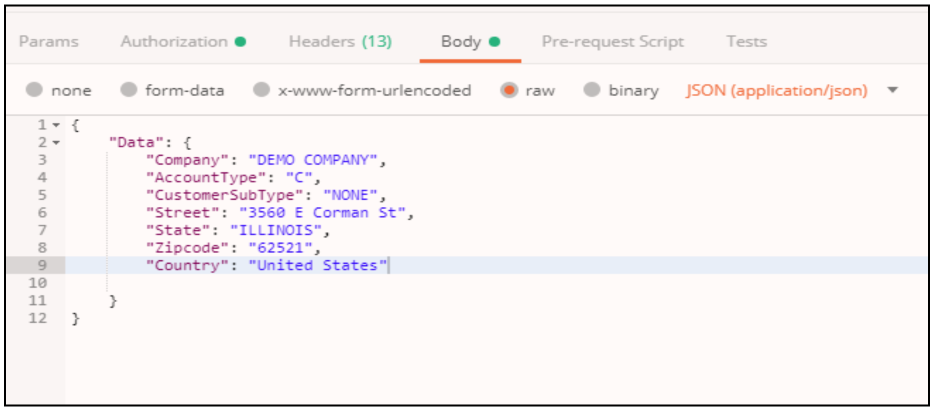
When you customize the SO by adding a new field to the SOMAST/SOITEM/SORELS as a new attribute (right click on SOMAST/SOITEM/SORELS in SCRMNT properties), when you look at **APICONFIG** for the Sales Order, the new field would be listed in the right pane with all the other fields associated to whichever table you added it to. Although when you added the attribute it created the _EXT table, the logic still reads the normal Sales Order business object and grabs the custom XML and confirms the custom table/field but it is all still under the standard table. The custom fields show under the standard table/entity as that is how the business logic reads the customization. If you did CTRL + F3 on your custom field, it would tell you no help exists but it would still refer SOMAST/SOITEM/SORELS, not the _EXT table.

8. Click **Save** to save the record.
9. Switch to POSTMAN tool and create a POST request to get the access token to change the sales order status.
   Endpoint - https://<localservername>/M2MIDSERVER/identity/connect/token
10. In the request header, specify the company ID that you created in the **APICONFIG** screen.
11. Click **Send**.
    You will receive the response from the identity server that contains the access token, expiry time (in seconds) and token type.
12. To send a POST request to create a new customer, you must specify the following details:

- **Endpoint**: http://<<server>>/<contextpath>/api/<ObjectName>/
  Example: http://<servername>/api/account/
- **Header**:
  - **Company ID**: Company ID as specified in the **APICONFIG** screen > **Company ID** field.
  - **ClientName**: Client Name as specified in the **APICONFIG** screen > **Client Configuration** window > **Client Name** field.
  - **Authorization**: access_token received as response from the previous POST operation.

13. In the request body, specify the field values to create a new customer record in the **Accounts (CUST)** screen.

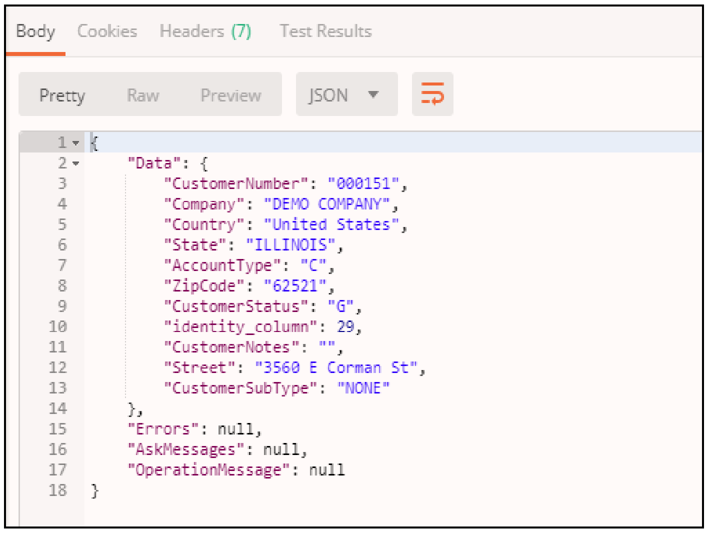
Figure 13: Create Account - Request Body



```
{
    "Data": {
        "Company": "DEMO COMPANY",
        "AccountType": "C",
        "CustomerSubType": "NONE",
        "Street": "3560 E Corman St",
        "State": "ILLINOIS",
        "Zipcode": "62521",
        "Country": "United States"

    }
}
```

14. Click **Send**. You will receive the response data as shown in the following image:

Figure 14: Create Account - Response Body



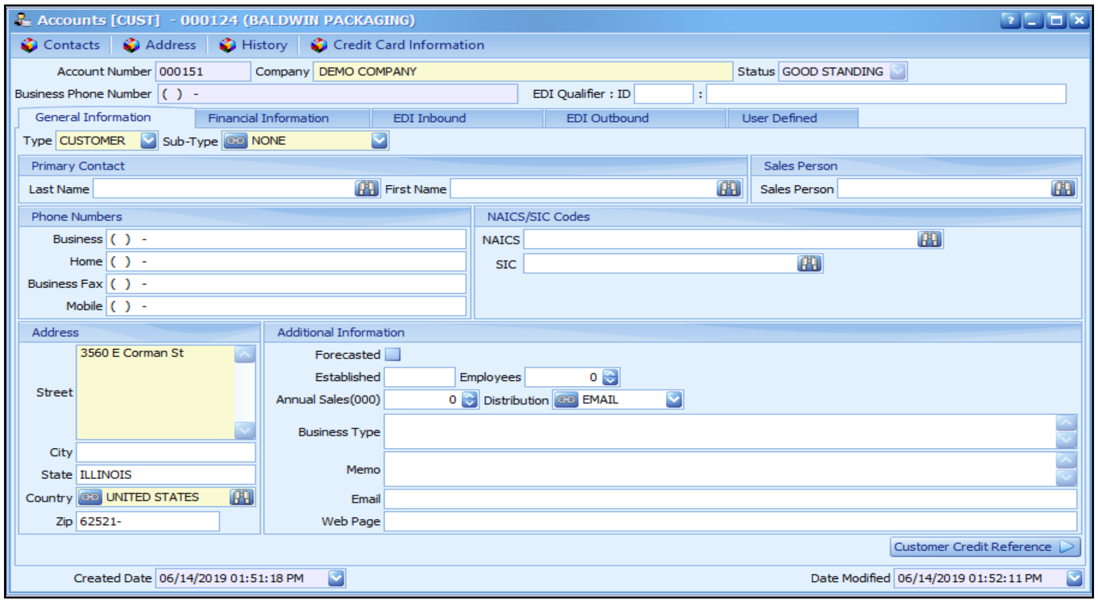
Body | Cookies | Headers (7) | Test Results

Pretty | Raw | Preview | JSON

```
{
    "Data": {
        "CustomerNumber": "000151",
        "Company": "DEMO COMPANY",
        "Country": "United States",
        "State": "ILLINOIS",
        "AccountType": "C",
        "ZipCode": "62521",
        "CustomerStatus": "G",
        "identity_column": 29,
        "CustomerNotes": "",
        "Street": "3560 E Corman St",
        "CustomerSubType": "NONE"
    },
    "Errors": null,
    "AskMessages": null,
    "OperationMessage": null
}
```

15. In **M2M** > **Accounts (CUST)** screen, a new customer record is created with the field values passed from API as shown in the following image:

Figure 15: Accounts (CUST) screen



Accounts [CUST] - 000124 (BALDWIN PACKAGING)

Contacts | Address | History | Credit Card Information

Account Number: 000151
Company: DEMO COMPANY
Status: GOOD STANDING
Business Phone Number: ( ) -
EDI Qualifier : ID

General Information | Financial Information | EDI Inbound | EDI Outbound | User Defined

Type: CUSTOMER
Sub-Type: NONE

Primary Contact
Last Name:
First Name:

Sales Person
Sales Person:

Phone Numbers
Business: ( ) -
Home: ( ) -
Business Fax: ( ) -
Mobile: ( ) -

NAICS/SIC Codes
NAICS:
SIC:

Address
Street: 3560 E Corman St
City:
State: ILLINOIS
Country: UNITED STATES
Zip: 62521-

Additional Information
Forecasted:
Established:
Employees: 0
Annual Sales(000): 0
Distribution: EMAIL
Business Type:
Memo:
Email:
Web Page:

Customer Credit Reference

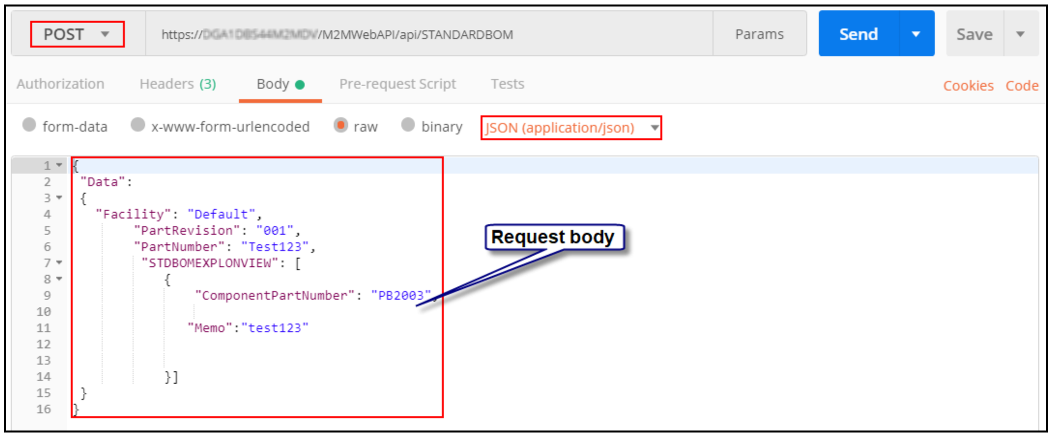
Created Date: 06/14/2019 01:51:18 PM
Date Modified: 06/14/2019 01:52:11 PM

Mid Complex Request:

Creating standard BOM with 2 entities (Standard BOM data and Component data).

Endpoint: https://<<server>>/M2MWebAPI/api/STANDARDBOM

Figure 16: POST - Mid Complex Request



https://[redacted]/M2MWebAPI/api/STANDARDBOM

```
{
 "Data":
 {
  "Facility": "Default",
       "PartRevision": "001",
       "PartNumber": "Test123",
        "STDBOMEXPLONVIEW": [
            {
                "ComponentPartNumber": "PB2003",

               "Memo":"test123"


            }]
 }
}
```

Upon successful processing of the request, the HTTP response code **201** is returned and the response body contains the data of the newly created BOM in JSON format.

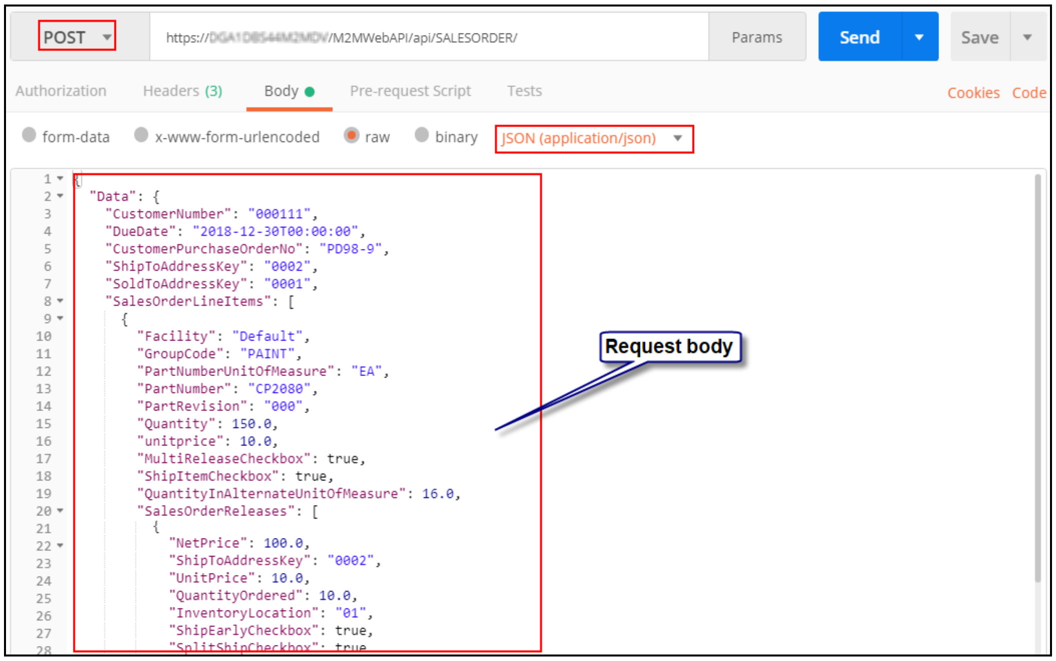
Complex Request:

Creating sales order with 3 entities (Sales Order master data, Sales Order Line Item data and Sales Order Release data).

The screen shot illustrates a request to create a sales order. Since this is a POST request, the request body contains the data being sent to the API. This is in JSON format.

Endpoint: https://<<server>>/M2MWebAPI/api/SALESORDER/

Figure 17: POST - Complex Request



https://[redacted]/M2MWebAPI/api/SALESORDER/

```
{
  "Data": {
    "CustomerNumber": "000111",
    "DueDate": "2018-12-30T00:00:00",
    "CustomerPurchaseOrderNo": "PD98-9",
    "ShipToAddressKey": "0002",
    "SoldToAddressKey": "0001",
    "SalesOrderLineItems": [
      {
        "Facility": "Default",
        "GroupCode": "PAINT",
        "PartNumberUnitOfMeasure": "EA",
        "PartNumber": "CP2080",
        "PartRevision": "000",
        "Quantity": 150.0,
        "unitprice": 10.0,
        "MultiReleaseCheckbox": true,
        "ShipItemCheckbox": true,
        "QuantityInAlternateUnitOfMeasure": 16.0,
        "SalesOrderReleases": [
          {
            "NetPrice": 100.0,
            "ShipToAddressKey": "0002",
            "UnitPrice": 10.0,
            "QuantityOrdered": 10.0,
            "InventoryLocation": "01",
            "ShipEarlyCheckbox": true,
            "SplitShipCheckbox": true
```

Upon successful processing of the request, the HTTP response code **201** is returned and the response body contains the data of the newly created sales order in JSON format.

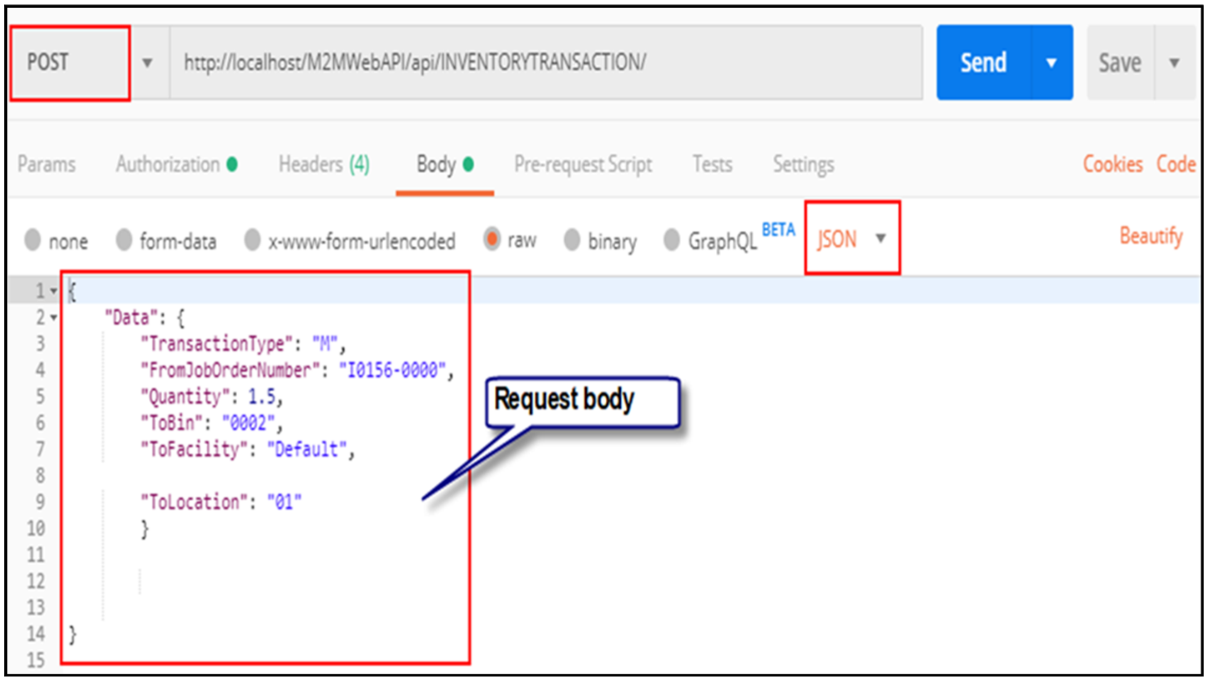
Examples for Inventory Transaction:

The screen shots illustrates a request to create different types of inventory transactions. Since this is a POST request, the request body contains the data being sent to the API. This is in JSON format.

Endpoint: http://<<server>>/M2MWebAPI/api/INVENTORYTRANSACTION/

- **Inventory Transaction Type: Move to Inventory**

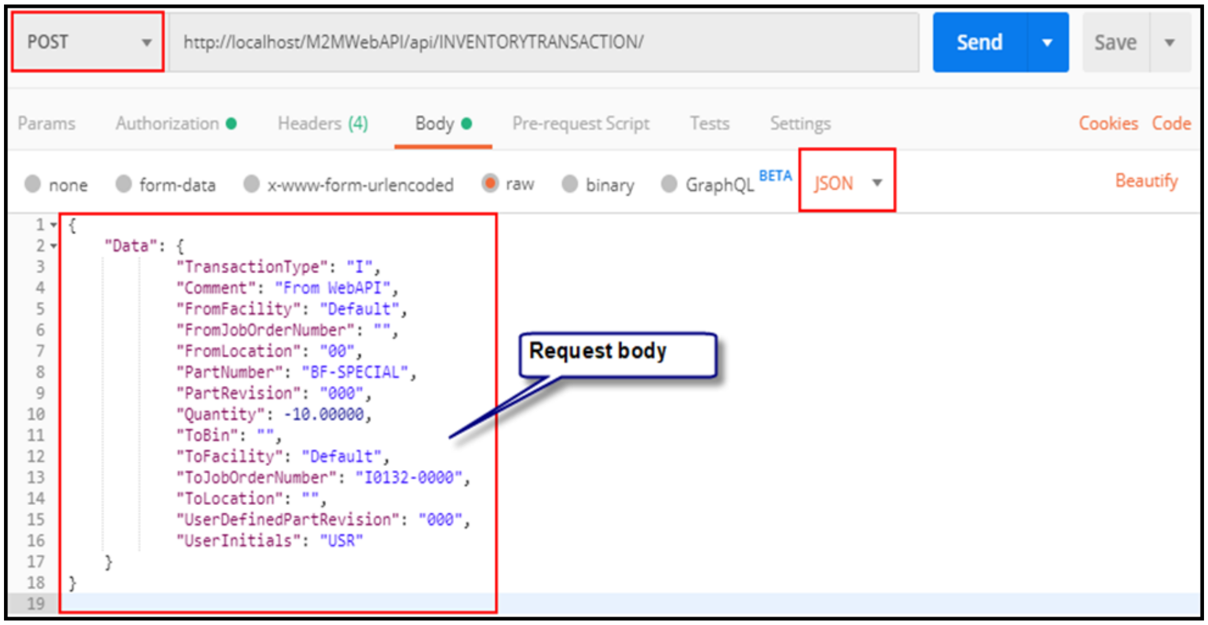
Figure 18: POST - Move to Inventory



```
{
    "Data": {
        "TransactionType": "M",
        "FromJobOrderNumber": "I0156-0000",
        "Quantity": 1.5,
        "ToBin": "0002",
        "ToFacility": "Default",

        "ToLocation": "01"
        }



}
```

- **Inventory Transaction Type: Miscellaneous Issues For Job Order**

Figure 19: POST - Miscellaneous Issues For Job Order

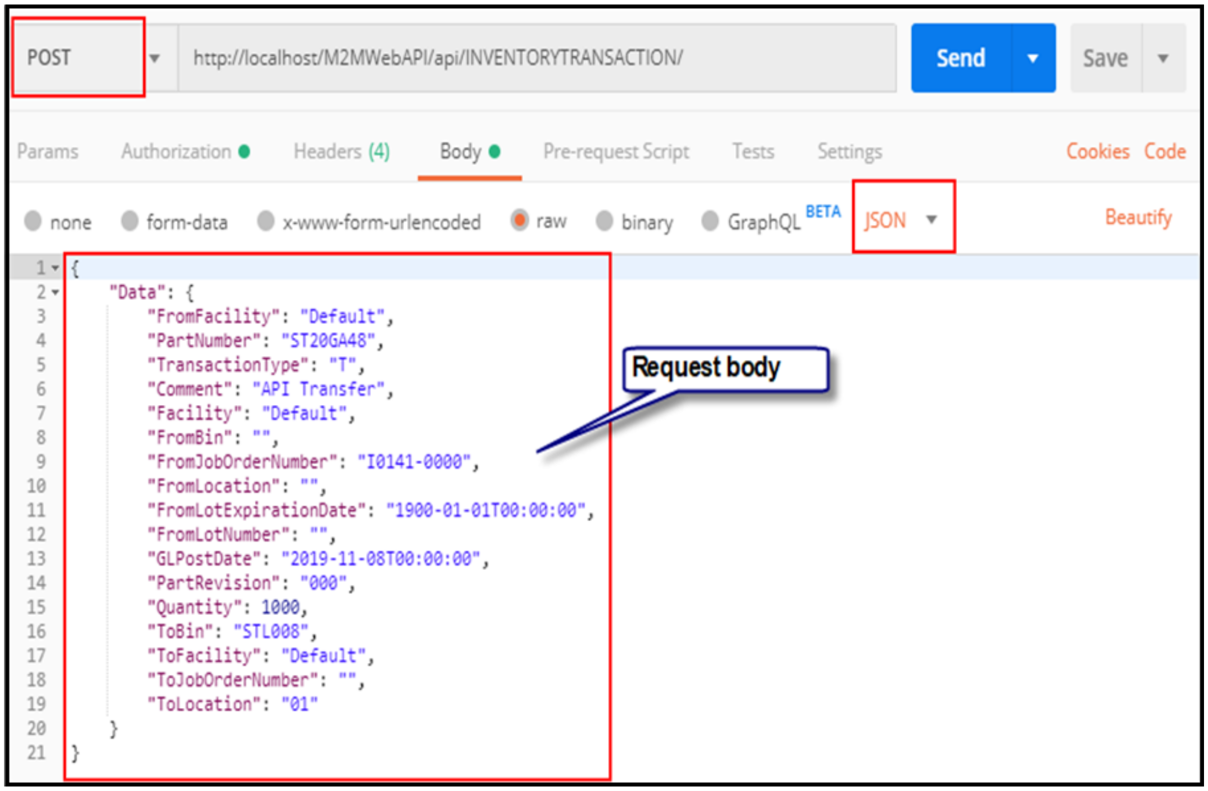


```
http://localhost/M2MWebAPI/api/INVENTORYTRANSACTION/
```

```
{
    "Data": {
            "TransactionType": "I",
            "Comment": "From WebAPI",
            "FromFacility": "Default",
            "FromJobOrderNumber": "",
            "FromLocation": "00",
            "PartNumber": "BF-SPECIAL",
            "PartRevision": "000",
            "Quantity": -10.00000,
            "ToBin": "",
            "ToFacility": "Default",
            "ToJobOrderNumber": "I0132-0000",
            "ToLocation": "",
            "UserDefinedPartRevision": "000",
            "UserInitials": "USR"
    }
}
```

- **Inventory Transaction Type: Transfer**

Figure 20: POST - Transfer

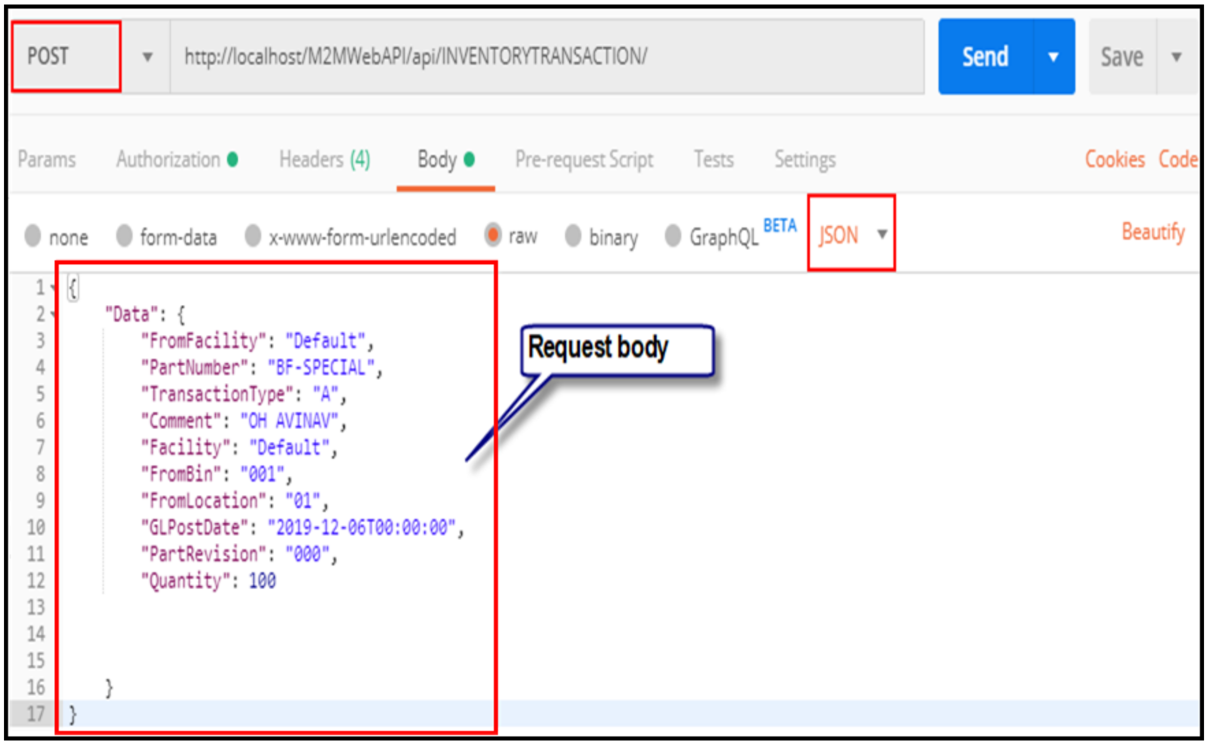


```
http://localhost/M2MWebAPI/api/INVENTORYTRANSACTION/
```

```
{
    "Data": {
        "FromFacility": "Default",
        "PartNumber": "ST20GA48",
        "TransactionType": "T",
        "Comment": "API Transfer",
        "Facility": "Default",
        "FromBin": "",
        "FromJobOrderNumber": "I0141-0000",
        "FromLocation": "",
        "FromLotExpirationDate": "1900-01-01T00:00:00",
        "FromLotNumber": "",
        "GLPostDate": "2019-11-08T00:00:00",
        "PartRevision": "000",
        "Quantity": 1000,
        "ToBin": "STL008",
        "ToFacility": "Default",
        "ToJobOrderNumber": "",
        "ToLocation": "01"
    }
}
```

- **Inventory Transaction Type: On Hold Adjustment**

Figure 21: POST - On Hold Adjustment



http://localhost/M2MWebAPI/api/INVENTORYTRANSACTION/

```
{
    "Data": {
        "FromFacility": "Default",
        "PartNumber": "BF-SPECIAL",
        "TransactionType": "A",
        "Comment": "OH AVINAV",
        "Facility": "Default",
        "FromBin": "001",
        "FromLocation": "01",
        "GLPostDate": "2019-12-06T00:00:00",
        "PartRevision": "000",
        "Quantity": 100



    }
}
```

Update Operation

Request Type	**PUT**
Purpose	Updates the details of the Object based on the input parameters passed along the request with the details passed along the request body.
Endpoint	https://<<server>>/<contextpath>/api/<<Object Name>>/<<input parameters>>

Quick Checklist

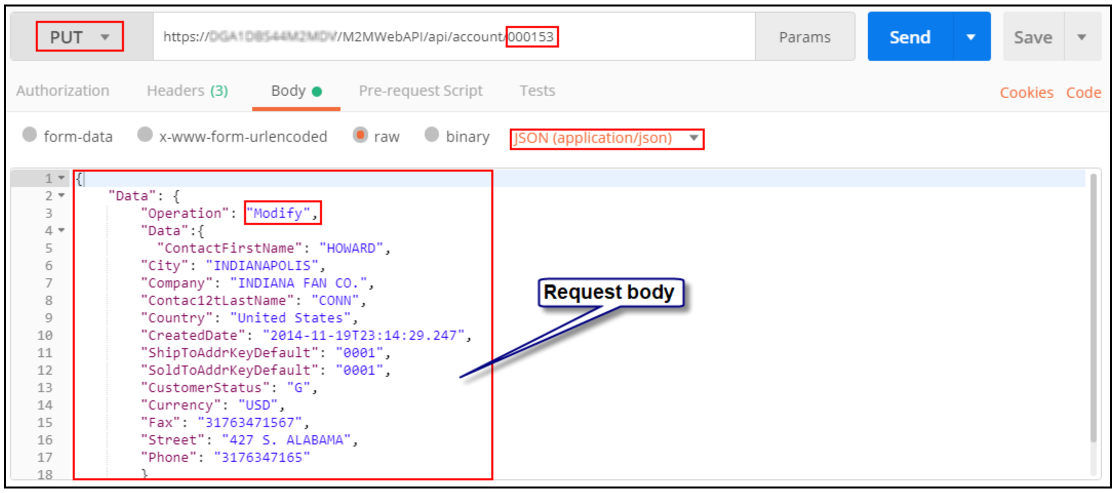
	In this chapter, you will learn about...
☐	PUT - *Simple Request:*
☐	PUT - *Mid Complex Request:*
☐	PUT - *Complex Request:*

Simple Request:

Updating specific customer (`000153`) with single operation (modify).

Endpoint: https://<<server>>/M2MWebAPI/api/Account/<<customer ID>>

Figure 22: PUT - Simple Request



PUT

https://[redacted]/M2MWebAPI/api/account/000153

```
{
    "Data": {
        "Operation": "Modify",
        "Data":{
          "ContactFirstName": "HOWARD",
        "City": "INDIANAPOLIS",
        "Company": "INDIANA FAN CO.",
        "Contac12tLastName": "CONN",
        "Country": "United States",
        "CreatedDate": "2014-11-19T23:14:29.247",
        "ShipToAddrKeyDefault": "0001",
        "SoldToAddrKeyDefault": "0001",
        "CustomerStatus": "G",
        "Currency": "USD",
        "Fax": "31763471567",
        "Street": "427 S. ALABAMA",
        "Phone": "3176347165"
        }
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the updated account in JSON format.

We can also use the PUT request to create as well as delete the nested collection objects. This can be done by passing the command against the **Operation** in the request body.

Command	Description
Modify	To update the nested collection object.
Add	To create a nested collection object.
Delete	To remove a nested collection object.

Note: **Add** and **Delete** commands can be applied only to the nested collection objects.

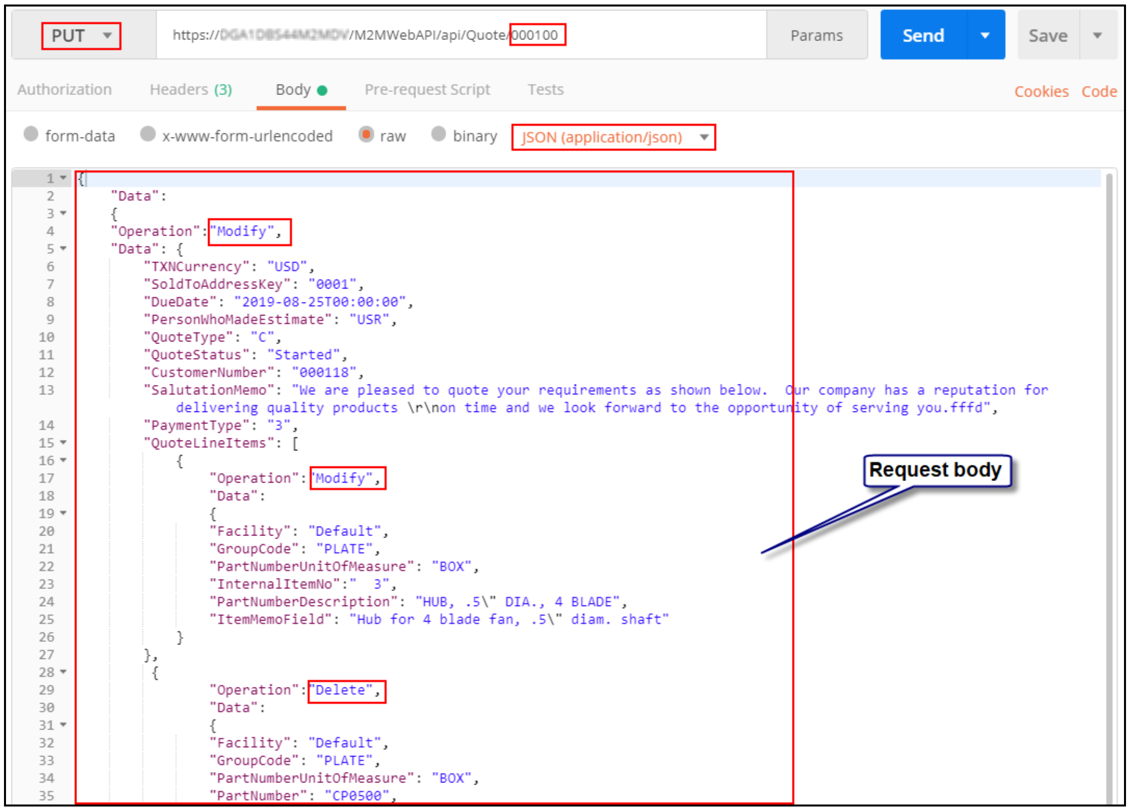
Mid Complex Request:

Updating Quote (`000100`) information with 2 operation (modify and delete). In the illustration, following operations are performed on Quote Line Items:

- Modifying data of one line item.
- Deleting one line item.

Endpoint: https://<<server>>/M2MWebAPI/api/Quote/<<quote number>>

Figure 23: PUT - Complex Request



```
{
    "Data":
    {
    "Operation":"Modify",
    "Data": {
        "TXNCurrency": "USD",
        "SoldToAddressKey": "0001",
        "DueDate": "2019-08-25T00:00:00",
        "PersonWhoMadeEstimate": "USR",
        "QuoteType": "C",
        "QuoteStatus": "Started",
        "CustomerNumber": "000118",
        "SalutationMemo": "We are pleased to quote your requirements as shown below.  Our company has a reputation for
            delivering quality products \r\non time and we look forward to the opportunity of serving you.fffd",
        "PaymentType": "3",
        "QuoteLineItems": [
            {
                "Operation":"Modify",
                "Data":
                {
                "Facility": "Default",
                "GroupCode": "PLATE",
                "PartNumberUnitOfMeasure": "BOX",
                "InternalItemNo":"  3",
                "PartNumberDescription": "HUB, .5\" DIA., 4 BLADE",
                "ItemMemoField": "Hub for 4 blade fan, .5\" diam. shaft"
            }
        },
         {
                "Operation":"Delete",
                "Data":
                {
                "Facility": "Default",
                "GroupCode": "PLATE",
                "PartNumberUnitOfMeasure": "BOX",
                "PartNumber": "CP0500",
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the updated quote in JSON format.

Complex Request:

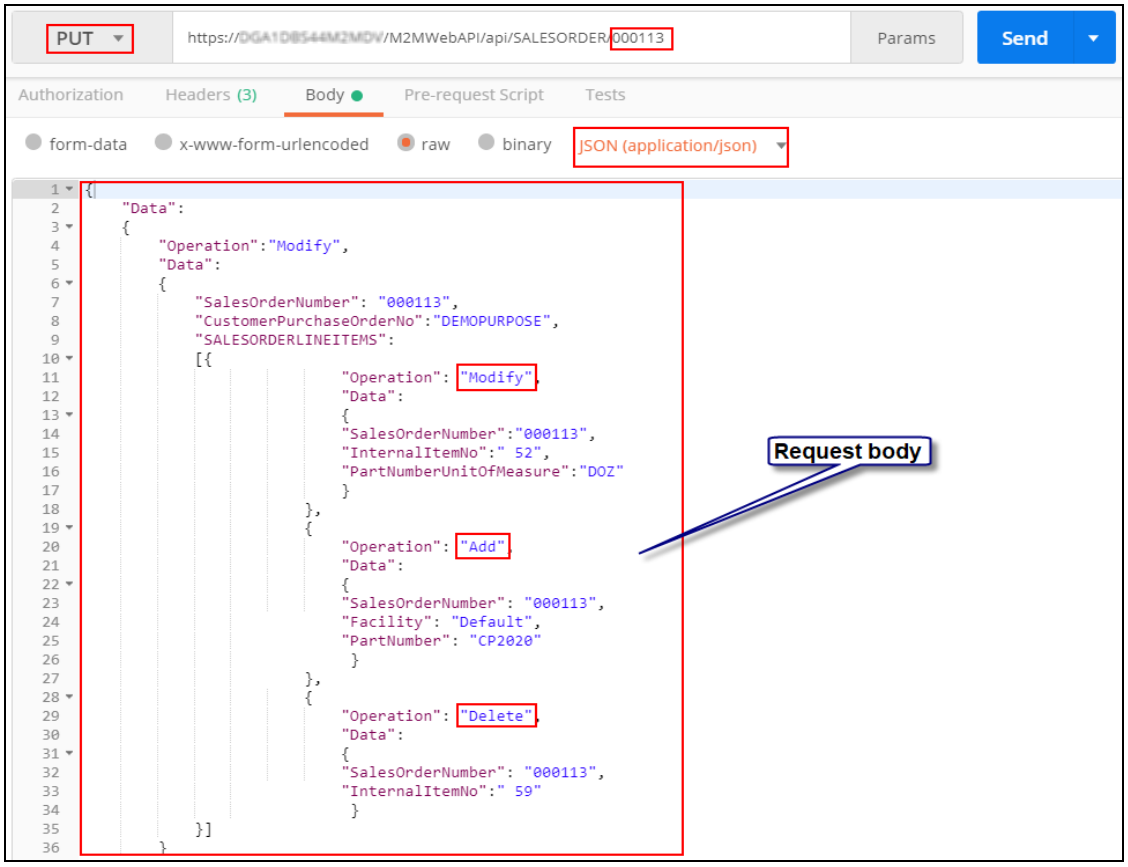
Updating sales order information with 3 operation (modify, add and delete). In the illustration, following operations are performed on Sales Order Line Items:

- Modifying data of one line item.
- Adding a new line item.
- Deleting one line item.

The screen shot illustrates a request to update sales order number: `000113`. The request body contains the data being sent to the API. This is in JSON format.

Endpoint: https://<<server>>/M2MWebAPI/api/SALESORDER/<<sales order number>>

Figure 24: PUT - Complex Request



https://[redacted]/M2MWebAPI/api/SALESORDER/000113

```
{
    "Data":
    {
        "Operation":"Modify",
        "Data":
        {
            "SalesOrderNumber": "000113",
            "CustomerPurchaseOrderNo":"DEMOPURPOSE",
            "SALESORDERLINEITEMS":
            [{
                            "Operation": "Modify",
                            "Data":
                            {
                            "SalesOrderNumber":"000113",
                            "InternalItemNo":" 52",
                            "PartNumberUnitOfMeasure":"DOZ"
                            }
                        },
                        {
                            "Operation": "Add",
                            "Data":
                            {
                            "SalesOrderNumber": "000113",
                            "Facility": "Default",
                            "PartNumber": "CP2020"
                             }
                        },
                        {
                            "Operation": "Delete",
                            "Data":
                            {
                            "SalesOrderNumber": "000113",
                            "InternalItemNo":" 59"
                             }
            }]
        }
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the updated sales order in JSON format.

Delete Operation

Request Type	**DELETE**
Purpose	Removes an entry under the Object based on the input parameters passed along the request.
Endpoint	https://<<server>>/<contextpath>/api/<<Object Name>>/<<input parameters>>

Quick Checklist

	In this chapter, you will learn about...
☐	DELETE - *Simple Request:*
☐	DELETE - *Mid Complex Request:*
☐	DELETE - *Complex Request:*

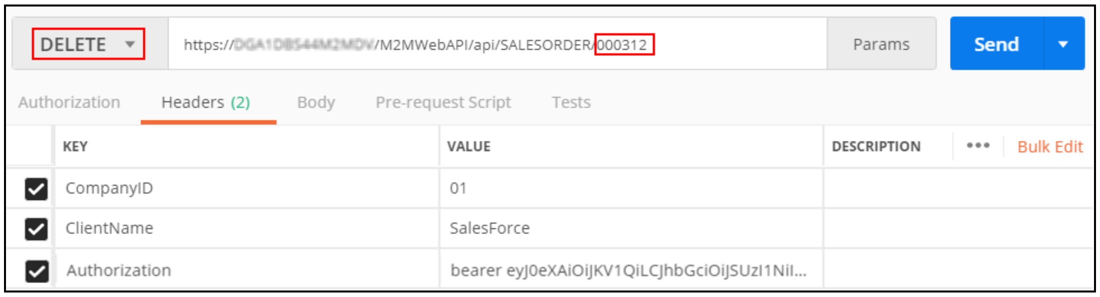
Simple Request:

Deleting a specific sales order.

The screen shot illustrates a request for deleting a sales order number: `000312`.

Endpoint: https://<<server>>/M2MWebAPI/api/SALESORDER/<<sales order number>>

Figure 25: DELETE - Simple Request

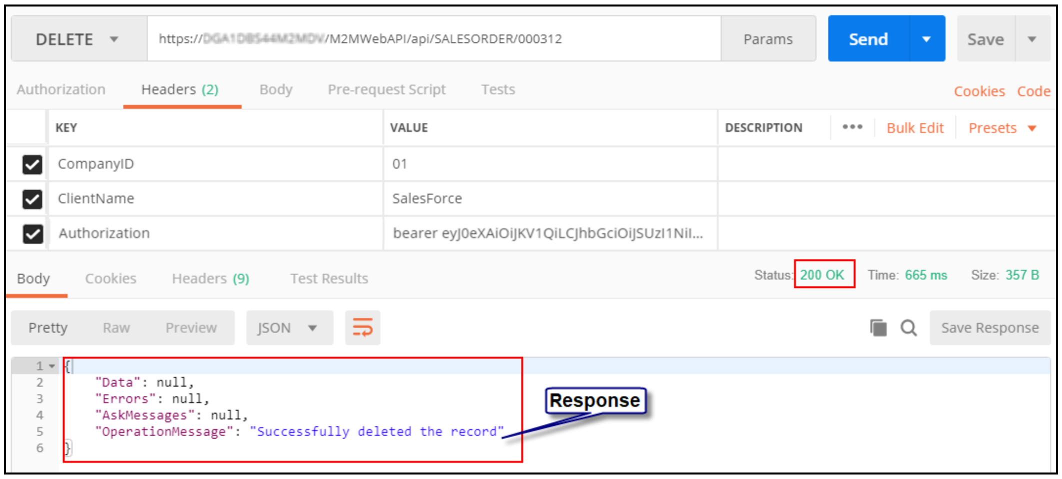


https://[redacted]/M2MWebAPI/api/SALESORDER/000312

	KEY	VALUE	DESCRIPTION
✓	CompanyID	01	
✓	ClientName	SalesForce	
✓	Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Upon successful processing of the request, the HTTP response code **200** is returned.

Figure 26: DELETE - Simple Response



DELETE https://[redacted]/M2MWebAPI/api/SALESORDER/000312

Headers (2)

KEY	VALUE	DESCRIPTION
CompanyID	01	
ClientName	SalesForce	
Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Status: 200 OK Time: 665 ms Size: 357 B

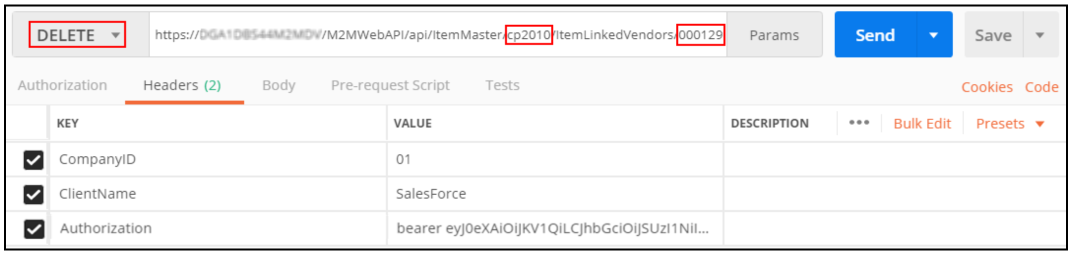
```
{
    "Data": null,
    "Errors": null,
    "AskMessages": null,
    "OperationMessage": "Successfully deleted the record"
}
```

Mid Complex Request:

Deleting a specific vendor (`000129`) that belong to a specific part (`cp2010`).

Endpoint: https://<<server>>/M2MWebAPI/api/ItemMaster/<<part number>>/ItemLinkedVendors/<<vendor ID>>

Figure 27: DELETE - Mid Complex Request



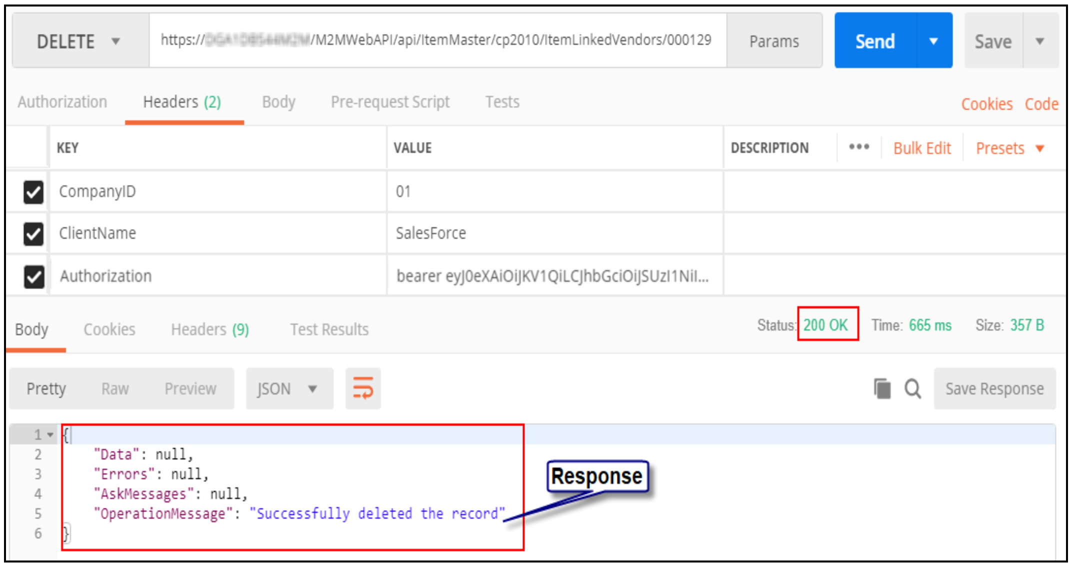
DELETE https://[redacted]/M2MWebAPI/api/ItemMaster/cp2010/ItemLinkedVendors/000129

Headers (2)

KEY	VALUE	DESCRIPTION
CompanyID	01	
ClientName	SalesForce	
Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Upon successful processing of the request, the HTTP response code **200** is returned.

Figure 28: DELETE - Mid Complex Response



https://[redacted]/M2MWebAPI/api/ItemMaster/cp2010/ItemLinkedVendors/000129

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Status: 200 OK Time: 665 ms Size: 357 B

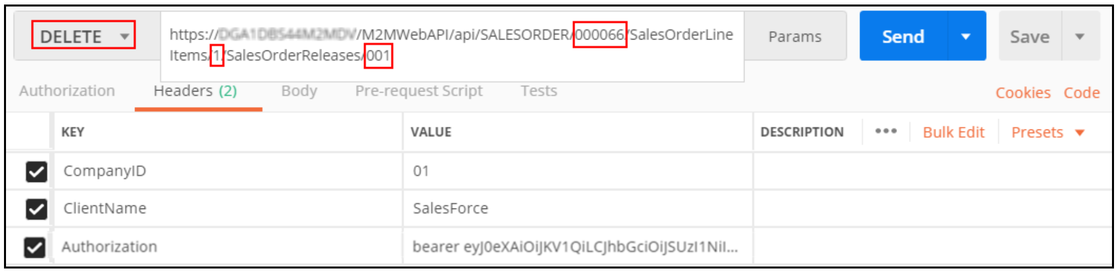
```
{
    "Data": null,
    "Errors": null,
    "AskMessages": null,
    "OperationMessage": "Successfully deleted the record"
}
```

Complex Request:

Deleting a specific sales order release (`001`) of a specific sales order line item (`1`) that belong to a specific sales order (`000066`).

Endpoint: https://<<server>>/M2MWebAPI/api/SALESORDER/<<sales order number>>/SalesOrderLineItems/<<sales order line item>>/SalesOrderReleases/<<sales order release number>>

Figure 29: DELETE - Complex Request

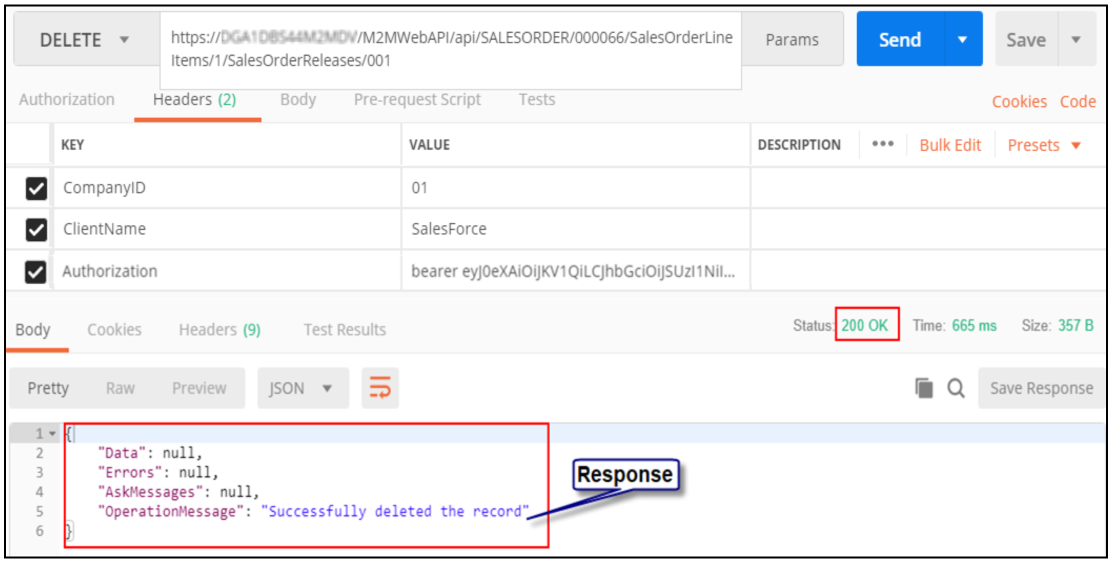


https://[redacted]/M2MWebAPI/api/SALESORDER/000066/SalesOrderLineItems/1/SalesOrderReleases/001

KEY	VALUE	DESCRIPTION
✓ CompanyID	01	
✓ ClientName	SalesForce	
✓ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI...	

Upon successful processing of the request, the HTTP response code **200** is returned.

Figure 30: DELETE - Complex Response



https://[redacted]/M2MWebAPI/api/SALESORDER/000066/SalesOrderLineItems/1/SalesOrderReleases/001

KEY	VALUE	DESCRIPTION
☑ CompanyID	01	
☑ ClientName	SalesForce	
☑ Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Nil...	

Status: 200 OK Time: 665 ms Size: 357 B

```
{
    "Data": null,
    "Errors": null,
    "AskMessages": null,
    "OperationMessage": "Successfully deleted the record"
}
```

Copy and Change Status

Quick Checklist

	In this chapter, you will learn about...
☐	*CopyPath in the JSON body*
☐	POST & PUT - *Coping data except Line Items*
☐	POST -*Copying from Job Order*
☐	POST - *Copying data from same record*
☐	POST - *Copying data from other record*
☐	POST - *Copying specific item from item level of other record*
☐	POST - *Copying line items from other record*
☐	PUT - *Copying specific line item from item level of same record*
☐	PUT - *Change Status*

Request Type	**POST** and **PUT**
Purpose	Create new records from existing records.
Endpoint	https://<<server>>/<contextpath>/api/<<Object Name>>

Copy Operation

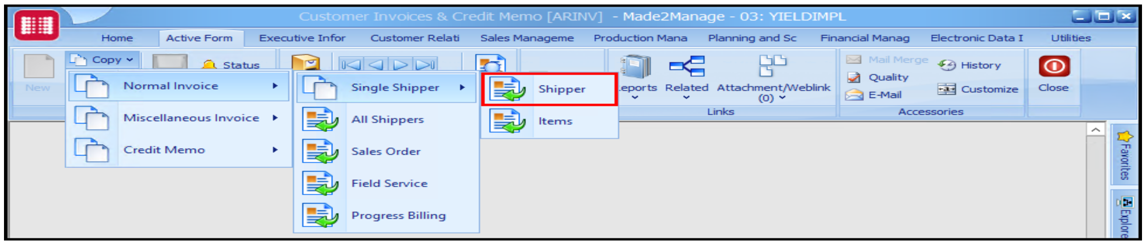
CopyPath in the JSON body

The CopyPath line in the JSON body is determined according to the screen from which the user wants to copy and POST.

Example:

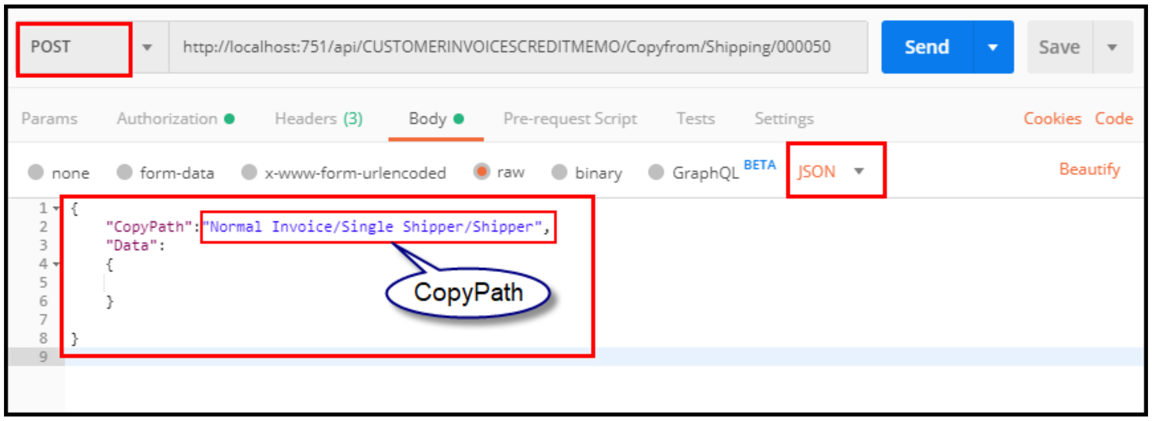
The screen shot illustrates the CopyPath for Shipper in the Customer Invoices & Credit Memo screen.

Figure 31: Customer Invoices & Credit Memo - CopyPath of Shipper



As the path in the screen for Shipper is **Normal Invoice** > **Single Shipper** > **Shipper**, the CopyPath line in the JSON body will be `Normal Invoice/Single Shipper/Shipper`.

Figure 32: POST Request - CopyPath in JSON



http://localhost:751/api/CUSTOMERINVOICESCREDITMEMO/Copyfrom/Shipping/000050

```
{
    "CopyPath":"Normal Invoice/Single Shipper/Shipper",
    "Data":
    {

    }
}
```

Coping data except Line Items

Screens which do not contain Line Items in them require two separate requests passed from the Postman to create new data and modify the data in it, which is POST and then PUT.

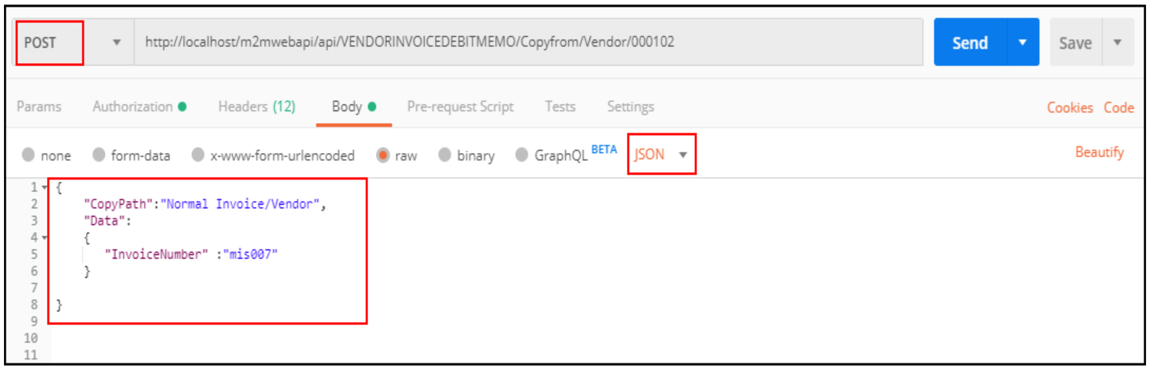
The copy options which do not contain Line Item information are:

- Customer in ARINV under Miscellaneous Invoice.
- Vendor in APINV under Miscellaneous Invoice.

Endpoint:
https://<<server>>/M2MWebAPI/api/VENDORINVOICEDEBITMEMO/Copyfrom/Vendor/<< vendor number>>

Screen shot illustrates a request to create a new Vendor Invoice from an existing Vendor Invoice. This request creates a new Miscellaneous Vendor Invoice without any Line Items information.

Figure 33: POST Request - Coping data from Vendor Invoice



POST http://localhost/m2mwebapi/api/VENDORINVOICEDEBITMEMO/Copyfrom/Vendor/000102

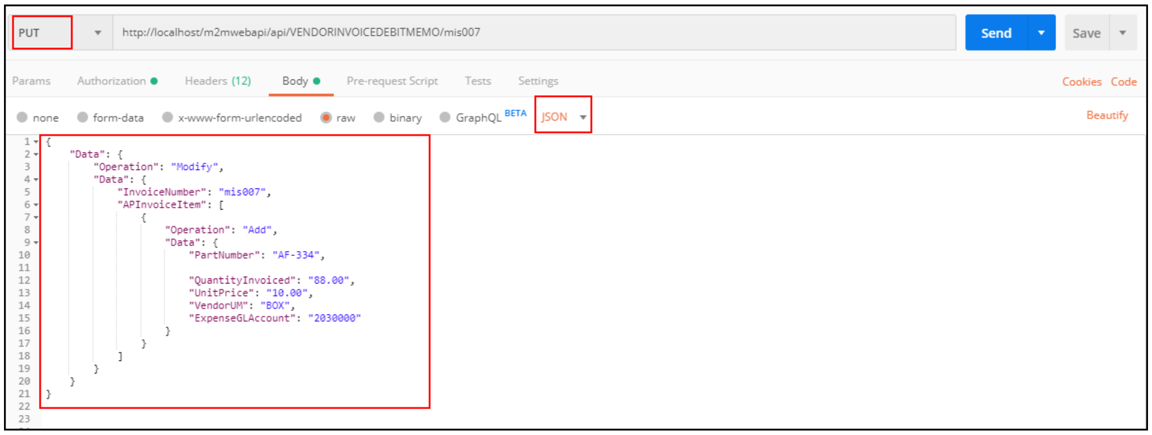
Body: raw, JSON

```
{
    "CopyPath":"Normal Invoice/Vendor",
    "Data":
    {
       "InvoiceNumber" :"mis007"
    }

}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

To modify and add Line Items to the newly created Vendor Invoice, send a PUT request with the URL as illustated in the screen shot.

Figure 34: PUT Request - Modifying data in Vendor Invoice



PUT http://localhost/m2mwebapi/api/VENDORINVOICEDEBITMEMO/mis007

Body: raw, JSON

```
{
    "Data": {
        "Operation": "Modify",
        "Data": {
            "InvoiceNumber": "mis007",
            "APInvoiceItem": [
                {
                    "Operation": "Add",
                    "Data": {
                        "PartNumber": "AF-334",

                        "QuantityInvoiced": "88.00",
                        "UnitPrice": "10.00",
                        "VendorUM": "BOX",
                        "ExpenseGLAccount": "2030000"
                    }
                }
            ]
        }
    }
}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

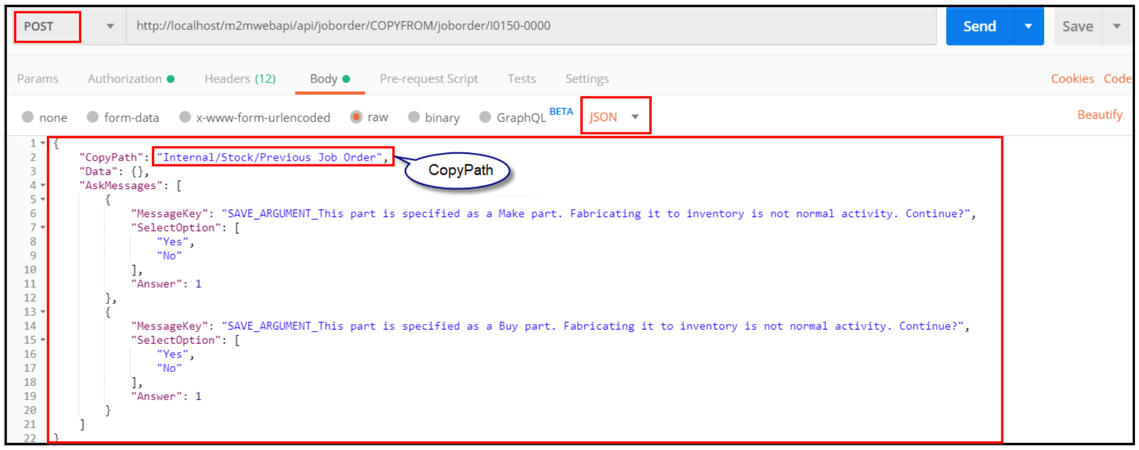
Copying from Job Order

Creating a new Job Order:

Endpoint: https://<<server>>/M2MWebAPI/api/joborder/COPYFROM/joborder/<<joborder number>>

Screen shot illustrates a request to create an Internal Stock Previous Job Order from an existing job order where the CopyPath line in the JSON body is `Internal/Stock/Previous Job Order`. Since this is a POST request, the request body contains the data being sent to the API. This is in JSON format.

Figure 35: POST Request - Copying data from other record



http://localhost/m2mwebapi/api/joborder/COPYFROM/joborder/I0150-0000

```
{
    "CopyPath": "Internal/Stock/Previous Job Order",
    "Data": {},
    "AskMessages": [
        {
            "MessageKey": "SAVE_ARGUMENT_This part is specified as a Make part. Fabricating it to inventory is not normal activity. Continue?",
            "SelectOption": [
                "Yes",
                "No"
            ],
            "Answer": 1
        },
        {
            "MessageKey": "SAVE_ARGUMENT_This part is specified as a Buy part. Fabricating it to inventory is not normal activity. Continue?",
            "SelectOption": [
                "Yes",
                "No"
            ],
            "Answer": 1
        }
    ]
}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

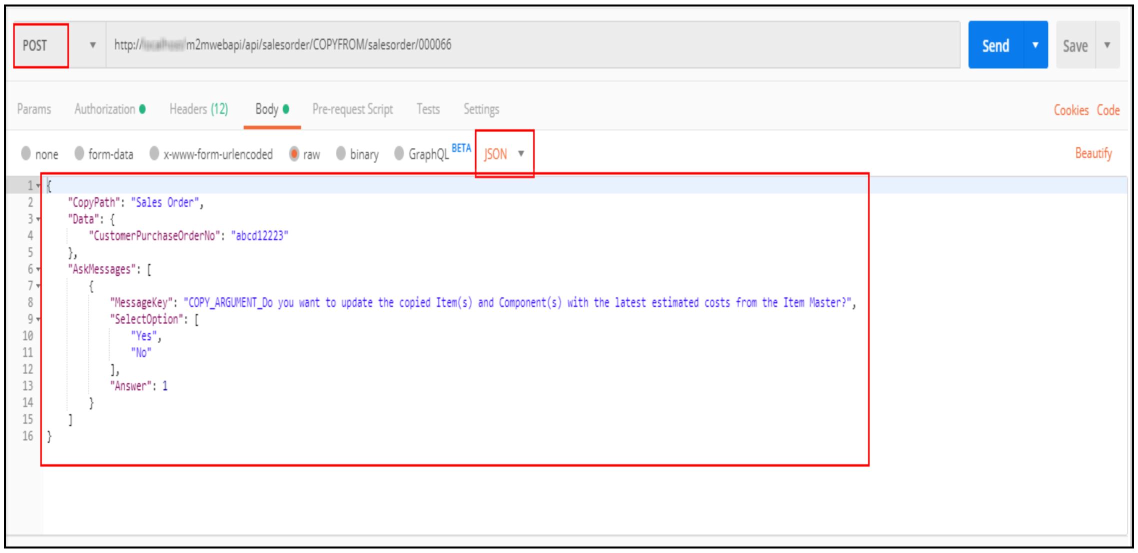
Copying data from same record

Creating a sales order from an existing sales order.

Endpoint: https://<<server>>/M2MWebAPI/api/salesorder/COPYFROM/salesorder/<<sales order number>>

The screen shot illustrates a request to create a sales order from an existing sales order. Since this is a POST request, the request body contains the data being sent to the API. This is in JSON format.

Figure 36: POST Request - Copying data from same record



http://[redacted]m2mwebapi/api/salesorder/COPYFROM/salesorder/000066

```
{
    "CopyPath": "Sales Order",
    "Data": {
        "CustomerPurchaseOrderNo": "abcd12223"
    },
    "AskMessages": [
        {
            "MessageKey": "COPY_ARGUMENT_Do you want to update the copied Item(s) and Component(s) with the latest estimated costs from the Item Master?",
            "SelectOption": [
                "Yes",
                "No"
            ],
            "Answer": 1
        }
    ]
}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

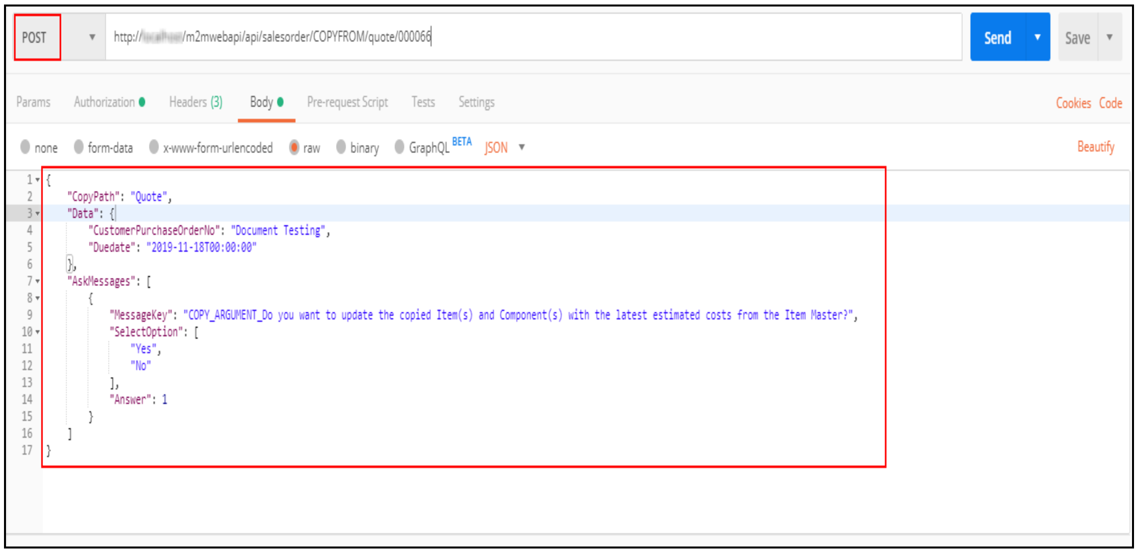
Copying data from other record

Creating a sales order from an existing quote.

Endpoint: https://<<server>>/M2MWebAPI/api/salesorder/COPYFROM/quote/<<quote number>>

The screen shot illustrates a request to create a sales order from an existing quote. Since this is a POST request, the request body contains the data being sent to the API. This is in JSON format.

Figure 37: POST Request - Copying data from other record



http://[redacted]/m2mwebapi/api/salesorder/COPYFROM/quote/000066

```
{
    "CopyPath": "Quote",
    "Data": {
        "CustomerPurchaseOrderNo": "Document Testing",
        "Duedate": "2019-11-18T00:00:00"
    },
    "AskMessages": [
        {
            "MessageKey": "COPY_ARGUMENT_Do you want to update the copied Item(s) and Component(s) with the latest estimated costs from the Item Master?",
            "SelectOption": [
                "Yes",
                "No"
            ],
            "Answer": 1
        }
    ]
}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

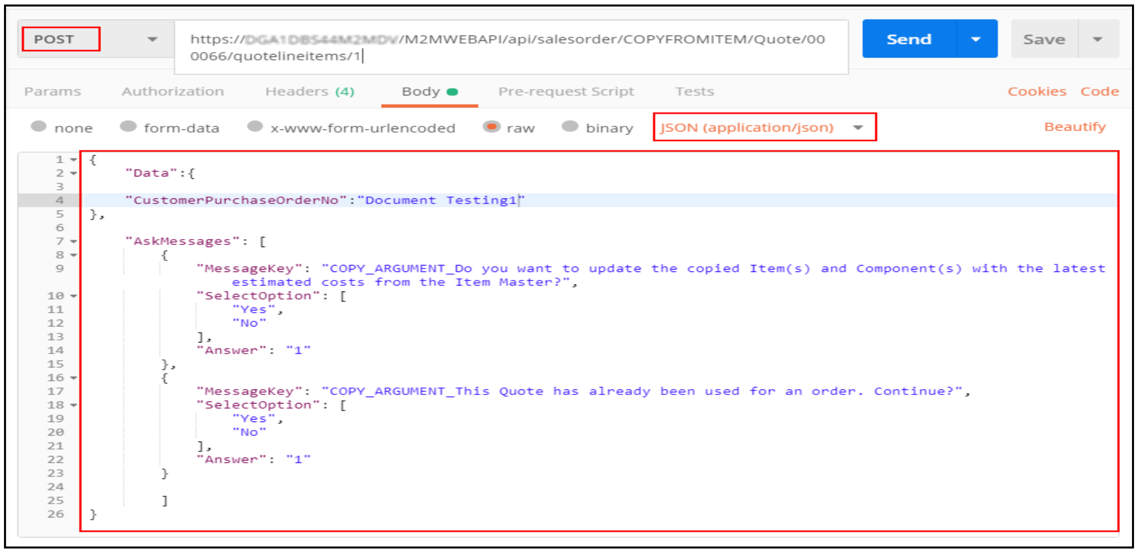
Copying specific item from item level of other record

Copying specific items from item level of a quote to a sales order.

Endpoint: https://<<server>>/M2MWebAPI/api/salesorder/COPYFROMITEM/quote/<<quote number>>/quote line items/<<quote line item number>>

The screen shot illustrates a request to copy a specific line item from a quote to a sales order. Since this is a POST request, the request body contains the data being sent to the API. This is in JSON format.

Figure 38: POST Request - Copying specific item from item level of other record



https://[redacted]/M2MWEBAPI/api/salesorder/COPYFROMITEM/Quote/000066/quotelineitems/1

JSON (application/json)

```
{
    "Data":{
    
    "CustomerPurchaseOrderNo":"Document Testing1"
},

    "AskMessages": [
        {
            "MessageKey": "COPY_ARGUMENT_Do you want to update the copied Item(s) and Component(s) with the latest
                estimated costs from the Item Master?",
            "SelectOption": [
                "Yes",
                "No"
            ],
            "Answer": "1"
        },
        {
            "MessageKey": "COPY_ARGUMENT_This Quote has already been used for an order. Continue?",
            "SelectOption": [
                "Yes",
                "No"
            ],
            "Answer": "1"
        }

        ]
}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

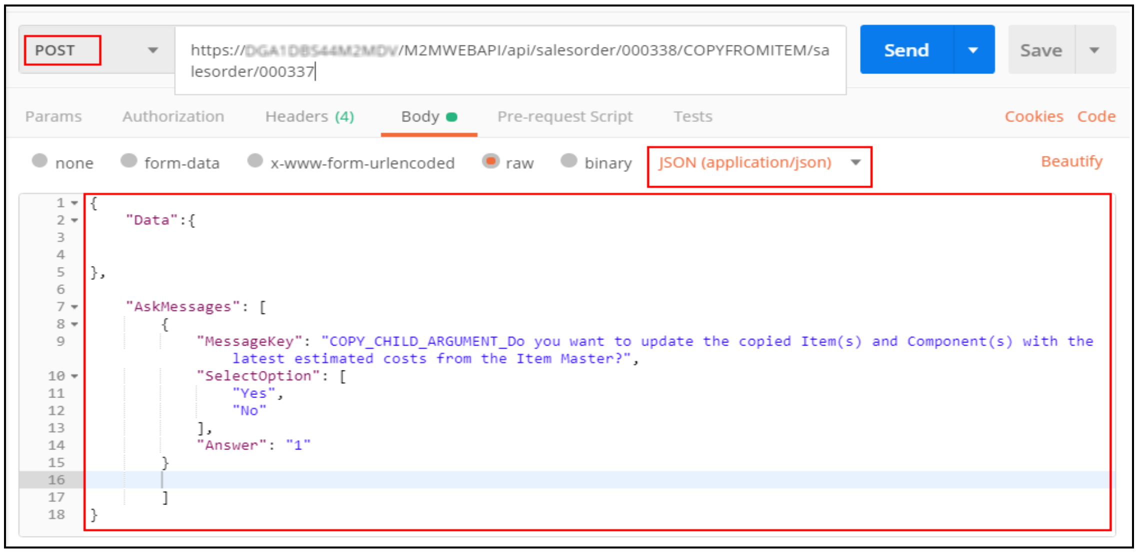
Copying line items from other record

Copying sales order line level items from an existing sales order to another sales order.

Endpoint: https://<<server>>/M2MWebAPI/api/salesorder/<<sales order number>>/COPYFROMITEM/salesorder/<<sales order number>>

The screen shot illustrates a request to copy the sales order line level items from an existing sales order to another sales order. Since this is a POST request, the request body contains the data being sent to the API. This is in JSON format.

Figure 39: POST Request - Copying line items from other record



https://[redacted]/M2MWEBAPI/api/salesorder/000338/COPYFROMITEM/salesorder/000337

JSON (application/json)

```
{
    "Data":{
    
    
},

    "AskMessages": [
        {
            "MessageKey": "COPY_CHILD_ARGUMENT_Do you want to update the copied Item(s) and Component(s) with the
                latest estimated costs from the Item Master?",
            "SelectOption": [
                "Yes",
                "No"
            ],
            "Answer": "1"
        }
        
        ]
}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

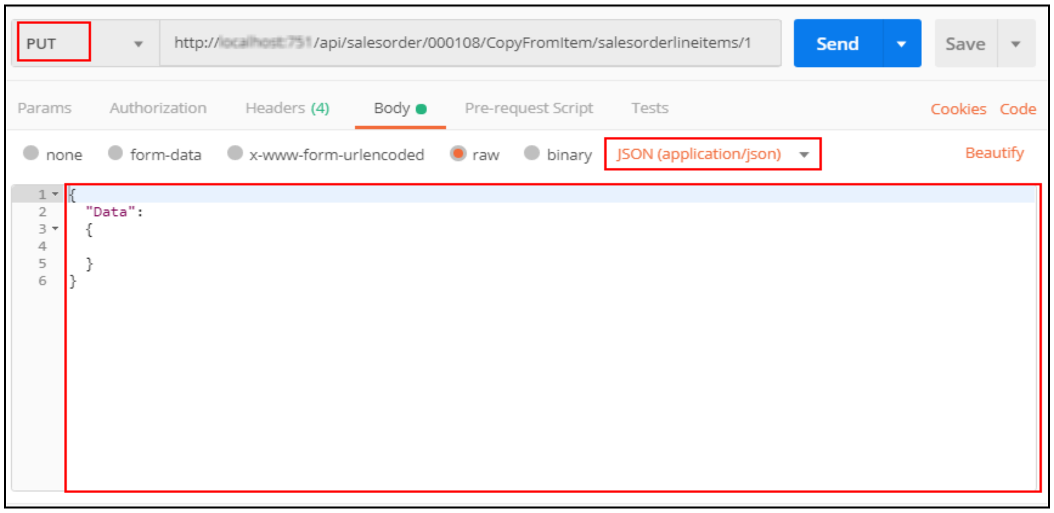
Copying specific line item from item level of same record

Copying a specific sales order line level item from an existing sales order to another sales order.

Endpoint: https://<<server>>/M2MWebAPI/api/salesorder/<<sales order number>>COPYFROMITEM/salesorder/salesorderlineitems/<<sales order line item number>>

The screen shot illustrates a request to copy an exisiting sales order line level item from an existing sales order to another sales order. Since this is a PUT request, the request body contains the data being sent to the API. This is in JSON format.

Figure 40: PUT - Copying specific line item from item level of same record



http://[redacted]/api/salesorder/000108/CopyFromItem/salesorderlineitems/1

```
{
  "Data":
  {

  }
}
```

Upon successful processing of the request, the HTTP response code **200** is returned and the response body contains the data of the created sales order in JSON format.

Change Status

Example:

To change the status of a Sales Order with API

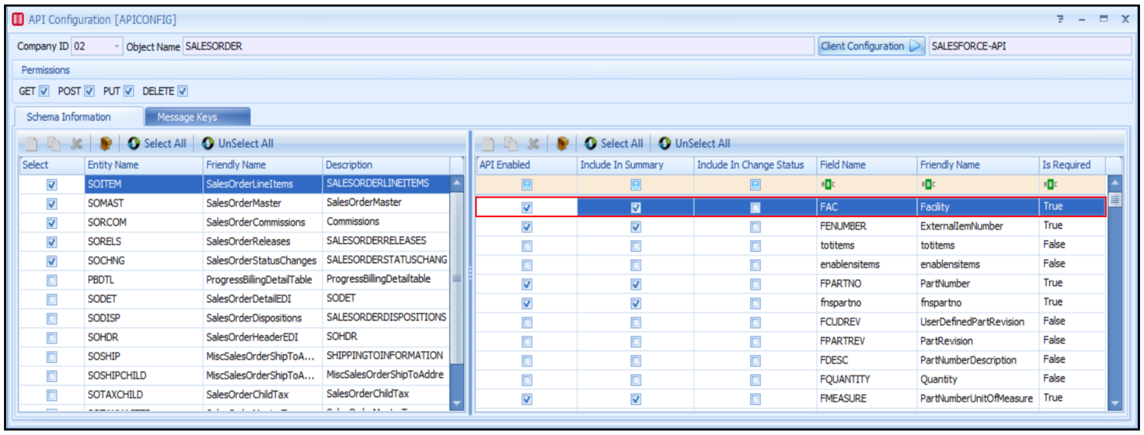
1. Open the **APICLIENT** screen and create a new client. If you want to use an existing client record, then go to step-2.
2. Open the **APICONFIG** screen and click **New**. If you already have configured the **APICONFIG** screen to enable Web API access to remote Web applications (clients or third party applications) for the required Business object (Eg. Sales Order, Account, Accounts Payable) and provided appropriate permissions, then go to step 7.
3. Select the company number and Object Name (Sales Order).
4. Specify the appropriate permissions (GET/POST/PUT/DELETE) based on the Object Name.

5. Click **Client Configuration** and select the **Client Name**.
6. Click **Save**.
7. In the **Schema Information** tab, select the check box to include the entities associated with the Object in the API response. In the right pane, select the check box to include the fields associated with the entity in the API response.

Note: To change the status of existing records in M2M using API calls, you must select the **API Enabled** check box for the **FSTATUS** field in the **APICONFIG** screen > **Schema Information** tab > Associated fields in the right pane.

Note: The check box **Include In Change Status** in the **APICONFIG** screen > **Schema Information** in the right pane, allows the user to edit the column along with changing status. By default the check box is enabled for the applicable Associated fields for Change Status.

Figure 41: APICONFIG screen



Company ID 02 Object Name SALESORDER Client Configuration SALESFORCE-API

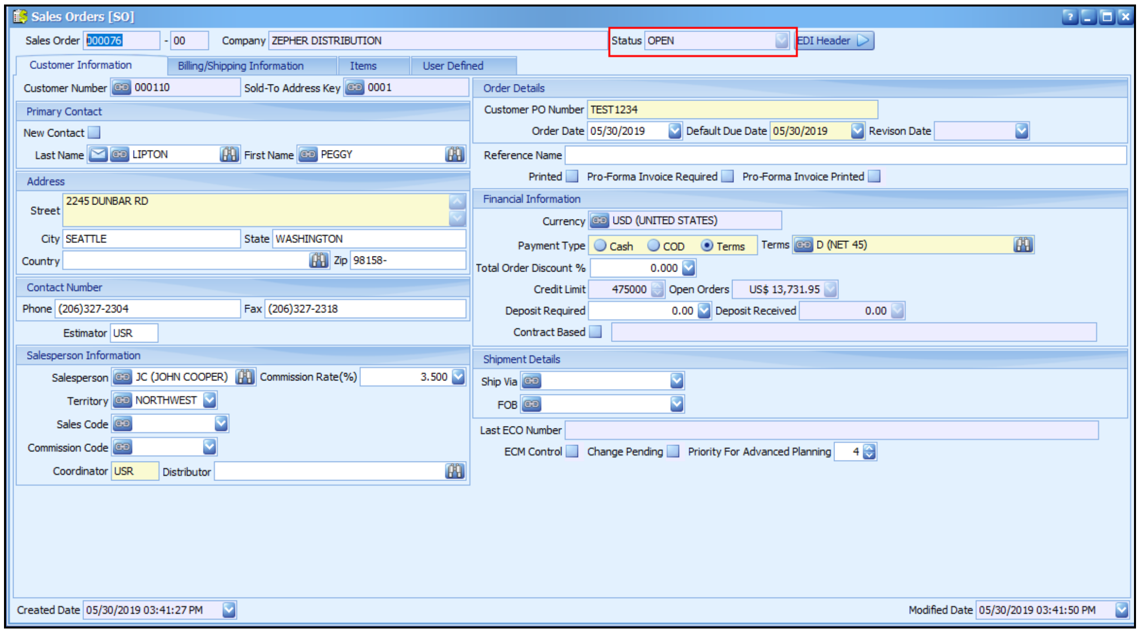
GET ✓ POST ✓ PUT ✓ DELETE ✓

Select	Entity Name	Friendly Name	Description
✓	SOITEM	SalesOrderLineItems	SALESORDERLINEITEMS
✓	SOMAST	SalesOrderMaster	SalesOrderMaster
✓	SORCOM	SalesOrderCommissions	Commissions
✓	SORELS	SalesOrderReleases	SALESORDERRELEASES
✓	SOCHNG	SalesOrderStatusChanges	SALESORDERSTATUSCHANG
☐	PBDTL	ProgressBillingDetailTable	ProgressBillingDetailtable
☐	SODET	SalesOrderDetailEDI	SODET
☐	SODISP	SalesOrderDispositions	SALESORDERDISPOSITIONS
☐	SOHDR	SalesOrderHeaderEDI	SOHDR
☐	SOSHIP	MiscSalesOrderShipToA...	SHIPPINGTOINFORMATION
☐	SOSHIPCHILD	MiscSalesOrderShipToA...	MiscSalesOrderShipToAddre
☐	SOTAXCHILD	SalesOrderChildTax	SalesOrderChildTax

API Enabled	Include In Summary	Include In Change Status	Field Name	Friendly Name	Is Required
✓	✓	☐	FAC	Facility	True
✓	✓	☐	FENUMBER	ExternalIemNumber	True
☐	☐	☐	totitems	totitems	False
☐	☐	☐	enablensitems	enablensitems	False
✓	✓	☐	FPARTNO	PartNumber	True
✓	✓	☐	fnspartno	fnspartno	True
☐	☐	☐	FCUDREV	UserDefinedPartRevision	False
☐	☐	☐	FPARTREV	PartRevision	False
☐	☐	☐	FDESC	PartNumberDescription	False
☐	☐	☐	FQUANTITY	Quantity	False
✓	✓	☐	FMEASURE	PartNumberUnitOfMeasure	True

8. Click **Save** to save the record.
9. In M2M, create a new sales order or if you want to change the status of an existing sales order, go to step 10.

Figure 42: Sales Orders (SO) screen - Open Status



Sales Orders [SO]

Sales Order 000076 - 00 Company ZEPHER DISTRIBUTION Status OPEN EDI Header

Customer Number 000110 Sold-To Address Key 0001

Primary Contact

New Contact ☐

Last Name LIPTON First Name PEGGY

Address

Street 2245 DUNBAR RD

City SEATTLE State WASHINGTON

Country Zip 98158-

Contact Number

Phone (206)327-2304 Fax (206)327-2318

Estimator USR

Salesperson Information

Salesperson JC (JOHN COOPER) Commission Rate(%) 3.500

Territory NORTHWEST

Sales Code

Commission Code

Coordinator USR Distributor

Order Details

Customer PO Number TEST1234

Order Date 05/30/2019 Default Due Date 05/30/2019 Revison Date

Reference Name

Printed ☐ Pro-Forma Invoice Required ☐ Pro-Forma Invoice Printed ☐

Financial Information

Currency USD (UNITED STATES)

Payment Type Cash COD [✓ Terms] Terms D (NET 45)

Total Order Discount % 0.000

Credit Limit 475000 Open Orders US$ 13,731.95

Deposit Required 0.00 Deposit Received 0.00

Contract Based ☐

Shipment Details

Ship Via

FOB

Last ECO Number

ECM Control ☐ Change Pending ☐ Priority For Advanced Planning 4

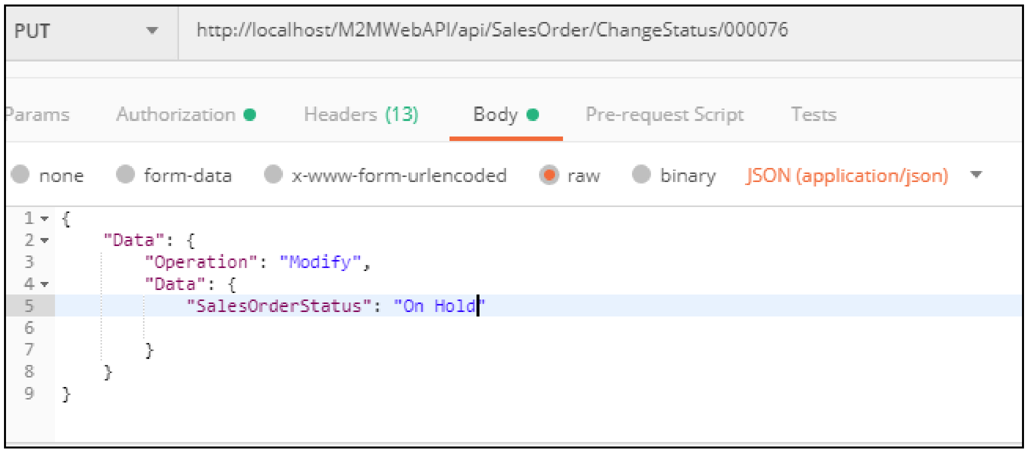
Created Date 05/30/2019 03:41:27 PM Modified Date 05/30/2019 03:41:50 PM

10. Switch to POSTMAN tool and create a POST request to get the access token to change the sales order status.
    Endpoint - https://<localservername>/M2MIDSERVER/identity/connect/token
11. In the request header, specify the company ID that you created in the **APICONFIG** screen.
12. Click **Send**.
    You will receive the response from the identity server that contains the access token, expiry time (in seconds) and token type.
13. To send a PUT request to change the sales order status, you must specify the following details:

- **Endpoint**: http://<<server>>/<contextpath>/api/<ObjectName>/<ChangeStatus>/<Value>/
  Example: http://<servername>/api/SalesOrder/ChangeStatus/000117/
- **Header**:
  - **Company ID**: Company ID as specified in the APICONFIG screen > Company ID field.
  - **ClientName**: Client Name as specified in the **APICONFIG** screen > **Client Configuration** window > **Client Name** field.
  - **Authorization**: access_token received as response from the previous POST operation.

14. In the request body, specify the new status of the sales order.

Figure 43: Change Status - Request Body

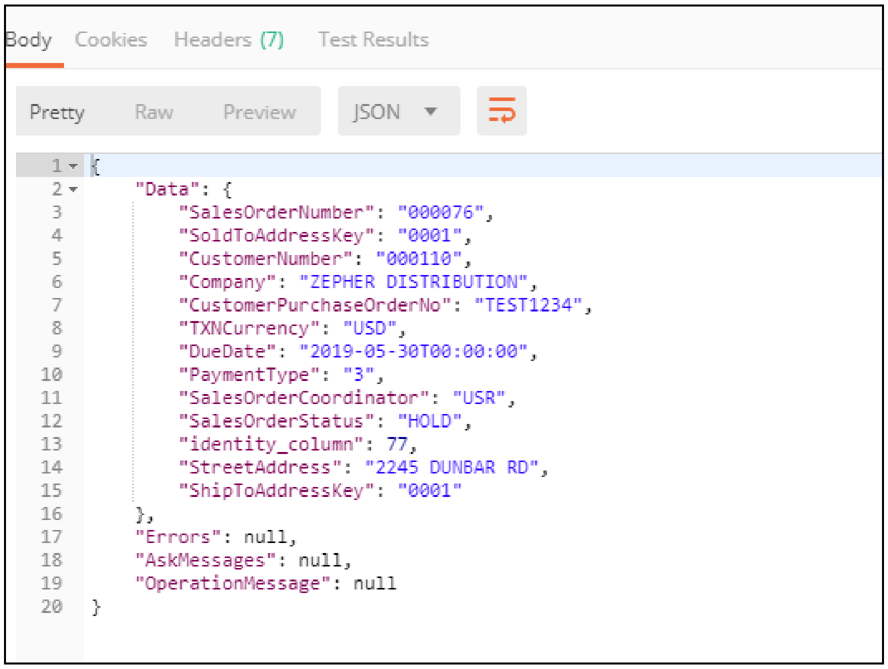


http://localhost/M2MWebAPI/api/SalesOrder/ChangeStatus/000076

```
{
    "Data": {
        "Operation": "Modify",
        "Data": {
            "SalesOrderStatus": "On Hold"

        }
    }
}
```

15. Click **Send**. You will receive the response data as shown in the following image:

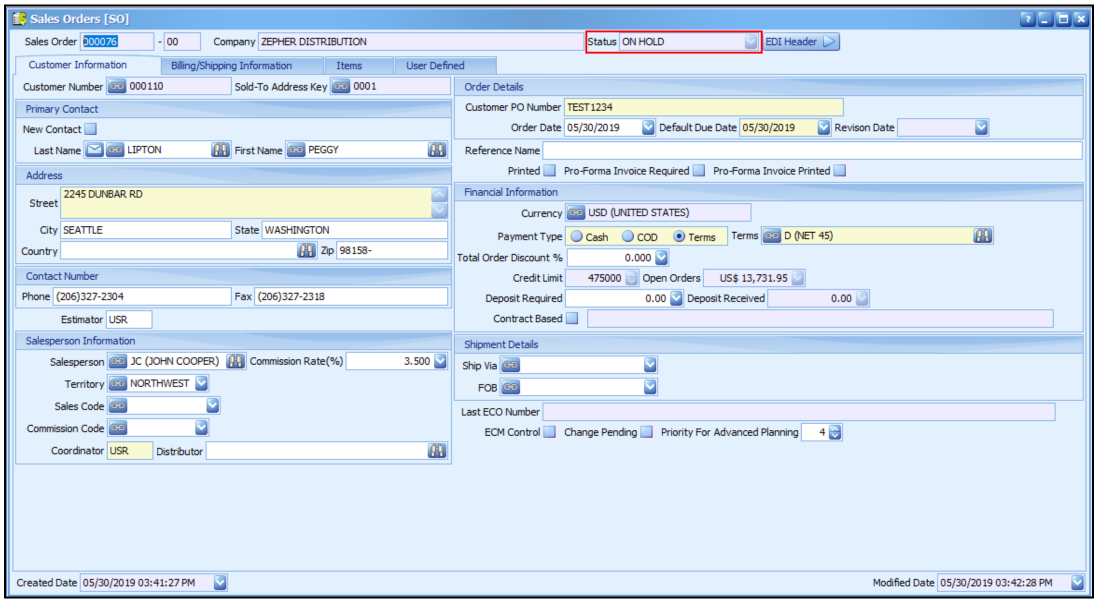
Figure 44: Change Status - Response Body



```
{
    "Data": {
        "SalesOrderNumber": "000076",
        "SoldToAddressKey": "0001",
        "CustomerNumber": "000110",
        "Company": "ZEPHER DISTRIBUTION",
        "CustomerPurchaseOrderNo": "TEST1234",
        "TXNCurrency": "USD",
        "DueDate": "2019-05-30T00:00:00",
        "PaymentType": "3",
        "SalesOrderCoordinator": "USR",
        "SalesOrderStatus": "HOLD",
        "identity_column": 77,
        "StreetAddress": "2245 DUNBAR RD",
        "ShipToAddressKey": "0001"
    },
    "Errors": null,
    "AskMessages": null,
    "OperationMessage": null
}
```

16. In **M2M** > **Sales Order** screen, the sales order status will be changed as shown in the following image:

Figure 45: Sales Order screen - On Hold Status


Sales Orders [SO]

Sales Order: 000076 - 00
Company: ZEPHER DISTRIBUTION
Status: ON HOLD

Customer Information | Billing/Shipping Information | Items | User Defined

Customer Number: 000110
Sold-To Address Key: 0001

Primary Contact
New Contact: ☐
Last Name: LIPTON
First Name: PEGGY

Address
Street: 2245 DUNBAR RD
City: SEATTLE
State: WASHINGTON
Country:
Zip: 98158-

Contact Number
Phone: (206)327-2304
Fax: (206)327-2318
Estimator: USR

Salesperson Information
Salesperson: JC (JOHN COOPER)
Commission Rate(%): 3.500
Territory: NORTHWEST
Sales Code:
Commission Code:
Coordinator: USR
Distributor:

Order Details
Customer PO Number: TEST1234
Order Date: 05/30/2019
Default Due Date: 05/30/2019
Revison Date:
Reference Name:
Printed: ☐ Pro-Forma Invoice Required: ☐ Pro-Forma Invoice Printed: ☐

Financial Information
Currency: USD (UNITED STATES)
Payment Type: Cash COD [✓ Terms]
Terms: D (NET 45)
Total Order Discount %: 0.000
Credit Limit: 475000
Open Orders: US$ 13,731.95
Deposit Required: 0.00
Deposit Received: 0.00
Contract Based: ☐

Shipment Details
Ship Via:
FOB:
Last ECO Number:
ECM Control: ☐ Change Pending: ☐ Priority For Advanced Planning: 4

Created Date: 05/30/2019 03:41:27 PM
Modified Date: 05/30/2019 03:42:28 PM

The following tables list the examples of Change Status and corresponding allowed new statuses for Quote, Sales Order, Job Order, and Purchase Order.

- **Quote**

Status	URL	Sample JSON

Open	http://localhost:751/api/Quote/ChangeStatus/000028	{
		"Data": {
		"Operation": "Modify",
		 "Data": {
		 "QuoteStatus": "Open"
		}
		 }
		}

Closed	http://localhost:751/api/Quote/ChangeStatus/000028	{
		"Data": {
		"Operation": "Modify",
		 "Data": {
		 "QuoteStatus": "Closed"
		}
		 }
		}

Awaiting Approval	http://localhost:751/api/Quote/ChangeStatus/000028	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"QuoteStatus": "Awaiting Approval"
		}
		}
		}

Cancelled	http://localhost:751/api/Quote/ChangeStatus/000028	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"QuoteStatus": "Cancelled"
		}
		}
		}

Quote statuses and corresponding allowed new statuses

- For **Quote Type**: **Customer**

Quote Status	Allowed New Status
Started	Awaiting Approval Open
Open	Cancelled Closed
Awaiting Approval	Open

Quote Status	Allowed New Status
Ordered	Closed Cancelled
Closed	Open
Cancelled	No Changes Allowed

- **Sales Order**

Status	URL	Sample JSON

Open	http://localhost:751/api/SalesOrder/ChangeStatus/000079	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"SalesOrderStatus":
		"OPEN"
		}
		}
		}

Closed	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"SalesOrderStatus":
		"CLOS"
		}
		}
		}

Awaiting Approval	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"SalesOrderStatus":
		"UNAP"
		}
		}
		}

Cancelled	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"SalesOrderStatus":
		"CANC"
		}
		}
		}

Status	URL	Sample JSON

On Hold	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"SalesOrderStatus":
		"HOLD"
		}
		}
		}

- **Sales Order Line Item**

To change the status of Line Items individually:

Status	URL	Sample JSON

Open	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"SalesOrderLineItems": [
		{
		"Operation": "Modify",
		"Data": {
		"SalesOrderNumber":
		"000065",
		"InternalItemNo": " 1",
		"LineItemStatus":"OPEN"
		}}
		]
		}
		}
		}

Closed	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"SalesOrderLineItems": [
		{
		"Operation": "Modify",
		"Data": {
		"SalesOrderNumber":
		"000065",
		"InternalItemNo": " 1",
		"LineItemStatus":"CLOS"
		}}
		]
		}
		}
		}

Status	URL	Sample JSON
Awaiting Approval	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "LineItemStatus":"UNAP" }}] } } }
Cancelled	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "LineItemStatus":"CANC" }}] } } }
On Hold	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "LineItemStatus":"HOLD" }}]

Status	URL	Sample JSON
		} } }

To change status of multiple Line items by passing array of items in body:

URL	Sample JSON

http://localhost:751/api/SalesOrder/ChangeStatus/000065	{
	"Data": {
	"Operation": "Modify",
	"Data": {
	"SalesOrderLineItems": [
	{
	"Operation": "Modify",
	"Data": {
	"SalesOrderNumber": "000065",
	"InternalItemNo": " 2",
	"LineItemStatus":"CLOS"
	} },
	{
	"Operation": "Modify",
	"Data": {
	"SalesOrderNumber": "000065",
	"InternalItemNo": " 7",
	"LineItemStatus":"OPEN"
	} },
	{
	"Operation": "Modify",
	"Data": {
	"SalesOrderNumber": "000065",
	"InternalItemNo": " 3",
	"LineItemStatus":"HOLD"
	} },
	{ "Operation": "Modify",
	"Data": {
	"SalesOrderNumber": "000065",
	"InternalItemNo": " 4",
	"LineItemStatus":"CANC"
	} }
	]
	}
	}
	}

- **Sales Order Releases**

To change the status of Releases individually:

Status	URL	Sample JSON
Open	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "SalesOrderReleases": [{ "Operation": "Modify", "Data": { "InternalItemNo": " 1", "ReleaseNumber": "002", "SalesOrderNumber": "000065", "ReleaseStatus":"OPEN" } }] } }] } } }
Closed	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "SalesOrderReleases": [{ "Operation": "Modify",

Status	URL	Sample JSON
		"Data": { "InternalItemNo": " 1", "ReleaseNumber": "002", "SalesOrderNumber": "000065", "ReleaseStatus":"CLOS" } }] } }] } } }
Awaiting Approval	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "SalesOrderReleases": [{ "Operation": "Modify", "Data": { "InternalItemNo": " 1", "ReleaseNumber": "002", "SalesOrderNumber": "000065", "ReleaseStatus":"UNAP" } }] } }] } }

Status	URL	Sample JSON
		}
Cancelled	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "SalesOrderReleases": [{ "Operation": "Modify", "Data": { "InternalItemNo": " 1", "ReleaseNumber": "002", "SalesOrderNumber": "000065", "ReleaseStatus":"CANC" } }] } }] } } }
On Hold	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "SalesOrderReleases": [{ "Operation": "Modify", "Data": {

Status	URL	Sample JSON
		"InternalItemNo": " 1", "ReleaseNumber": "002", "SalesOrderNumber": "000065", "ReleaseStatus":"HOLD" } }] } }] } } }

To change status of multiple Releases by passing array of items in body:

	URL	Sample JSON
	http://localhost:751/api/SalesOrder/ChangeStatus/000065	{ "Data": { "Operation": "Modify", "Data": { "SalesOrderLineItems": [{ "Operation": "Modify", "Data": { "SalesOrderNumber": "000065", "InternalItemNo": " 1", "SalesOrderReleases": [{ "Operation": "Modify", "Data": { "InternalItemNo": " 1", "ReleaseNumber": "002", "SalesOrderNumber": "000065", "ReleaseStatus":"OPEN" } }, { "Operation": "Modify", "Data": { "InternalItemNo": " 1", "ReleaseNumber": "001", "SalesOrderNumber": "000065", "ReleaseStatus":"CLOS" } }] } }] } } }

SO statuses and corresponding allowed new statuses

SO Status	Allowed New Status
Started	Open
	Awaiting Approval
Awaiting Approval	Open

SO Status	Allowed New Status
Open	Closed
	Cancelled
	On Hold
On Hold	Open
	Cancelled
	Closed
Closed	Open
Cancelled	Open

- **Job Order**

Status	URL	Sample JSON

Open	http://localhost:751/api/JobOrder/ChangeStatus/00056-0000	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"Status": "Open"
		}
		}
		}

Released	http://localhost:751/api/JobOrder/ChangeStatus/00056-0000	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"Status": "Released"
		}
		}
		}

Completed	http://localhost:751/api/JobOrder/ChangeStatus/00056-0000	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"Status": "Completed"
		}
		}
		}

Closed	http://localhost:751/api/JobOrder/ChangeStatus/00056-0000	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"Status": "Closed"
		}
		}
		}

Cancelled	http://localhost:751/api/JobOrder/ChangeStatus/00056-0000	{
		"Data": {
		"Operation": "Modify",
		"Data": {
		"Status": "Cancelled"
		}
		}
		}

Status	URL	Sample JSON
On Hold	http://localhost:751/api/JobOrder/ChangeStatus/00056-0000	{ "Data": { "Operation": "Modify", "Data": { "Status": "On Hold" } } }

JO statuses and corresponding allowed new statuses

JO Status	Allowed New Status
Started	Open
	Released
Open	Released
	Cancelled
Released	Open
	Completed
	Cancelled
	On Hold
Completed	Released
	Closed
Closed	Released
	Completed
Cancelled	Released
On Hold	Open
	Released
	Cancelled

Purchase Order

Status	URL	Sample JSON
Open	http://localhost/M2MWebAPI/api/PurchaseOrder/ChangeStatus/000074	{ "Data": { "Operation": "Modify", "Data": { "Status": "OPEN" } } }
Closed	http://localhost/M2MWebAPI/api/PurchaseOrder/ChangeStatus/000074	{ "Data": { "Operation": "Modify", "Data": { "Status": "CLOSED" } } }
Awaiting Approval	http://localhost/M2MWebAPI/api/PurchaseOrder/ChangeStatus/000074	{ "Data": { "Operation": "Modify", "Data": { "Status": "AWAITING" } } }
Cancelled	http://localhost/M2MWebAPI/api/PurchaseOrder/ChangeStatus/000074	{ "Data": { "Operation": "Modify", "Data": { "Status": "CANCELLED" } } }
On Hold	http://localhost/M2MWebAPI/api/PurchaseOrder/ChangeStatus/000074	{ "Data": { "Operation": "Modify", "Data": { "Status": "ON_ HOLD" } } }

PO statuses and corresponding allowed new statuses

PO Status	Allowed New Status
Started	Open
	Awaiting Approval
Awaiting Approval	Open
Open	Closed
	Cancelled
	On Hold
On Hold	Open
	Closed
	Cancelled
Closed	Open
	On Hold
Cancelled	No Changes Allowed