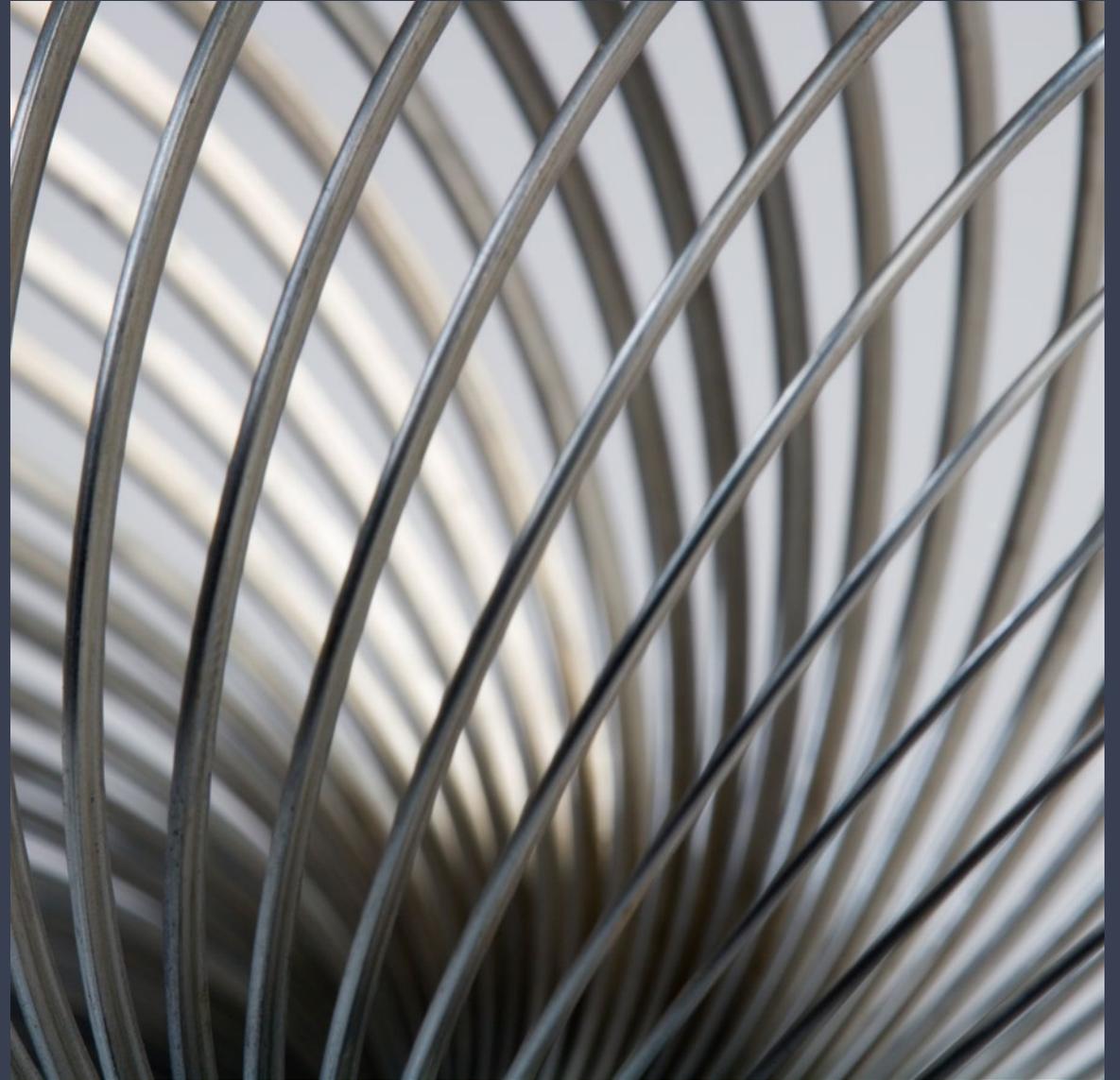


NUGM 2024

MANAGE 2000 AS  
A REST SERVICE  
PROVIDER

Peter Newby, Epicor



# AGENDA

1. The End Game
2. REST Fundamental Concepts
3. Manage 2000 M2K.REST.SERVICES
4. An Introduction to Postman
  1. Selecting and Configuring the Environment
  2. Authenticating User Credentials
  3. Testing Services
5. Copying REST calls from Postman to a web project
  1. Accessing Manage 2000 without a web server
  2. Accessing Manage 2000 without a browser



---

# THE END GAME

Accessing ERP Data from Out There

Way out there



# POSSIBLE REST SERVICE INTEGRATIONS

---

- Power BI, Excel
- CAD (Computer Aided Design)
- MES (Manufacturing Execution System)
- NODE.js microservices hub
- Parent company / sister company systems



---

# REST FUNDAMENTAL CONCEPTS

Manage 2000 REST Infrastructure



# MANAGE 2000 REST INFRASTRUCTURE

---

- URL Endpoint – `http://{Webserver}/{Manage 2000 web site} /MT/JSONServices/{service id}`
- Submit a CID Authentication Ticket named `M2k_Auth-Token` in the HTTP Header
- Use HTTP GET or HTTP POST
- Supply GET parameters as Querystring variables and POST parameters as Form variables
- Return will contain JSON in the HTTP Body



# ACCESSING ERP DATA FROM OUT THERE

---

- From somewhere outside of Manage 2000
- You want to develop an application using information from inside Manage 2000
- You want a test bench for experimenting with access to information
- You want documentation explaining how to make REST calls to Manage 2000 and what the input and output strings will look like
- After testing the REST service, you want to copy and paste it into your project outside of Manage 2000



---

# INTRODUCTION TO POSTMAN



# WHAT IS POSTMAN?

---

- An open source REST development community collaboration platform
- A platform for designing and building APIs
- A documentation development platform
- A documentation presentation and publishing interface
- A REST source generator for Copy/Paste example code in multiple languages
  - From the documentation use the Language dropdown menu to select your client language
  - From the documentation method example request use the copy icon to copy code to the clipboard
- A REST service test bench



# M2K.REST.SERVICES IN POSTMAN

The screenshot displays the Postman interface for the 'MANAGE 2000 REST SERVICES' collection. The top navigation bar includes 'ENVIRONMENT Manage 2000 REST Environment', 'LAYOUT Double Column', and 'LANGUAGE cURL - cURL'. A 'Run in Postman' button is visible in the top right corner.

**Item Validation Service**

**POST Item Validation**

...../MT/JSONServices/ValidateItemService.ashx?FileName=IM&TableNbr=0&ItemID=110&Prefix

Item Validation may be used to validate any primary key values in Manage 2000, not just inventory 'Items'.  
Validation may result in key normalization in which case the returned key may differ from the submitted key.  
The standard Manage 2000 validation display value will be returned with the normalized key.

**AUTHORIZATION** API Key

This request is using API Key from collection **Manage 2000 REST Services**

**PARAMS**

FileName	IM	
TableNbr	0	If the filename is TM then you must specify the TableNbr, otherwise it is ignored.
ItemID	110	
Prefix	Used with TM table number prefixes and in certain other situations.	

**Example Request**

```
curl --location --request POST '...../MT/JSONServices/ValidateItemService.ashx?FileName=IM&TableNbr=0&ItemID=110&Prefix'
```

**Example Response**

```
json
{
  "oValidItem": {
    "FileName": "IM",
    "TableNbr": "0",
    "ItemID": "100|01",
    "OrigItemID": "110",
    "NewItemID": "100|01",
    "Valid": "True",
    "Display": "System Pentium 900MHz 17'",
    "OptionalSupercede": "False",
    "OptionalSupercede": "False"
  }
}
```

**OCONV Service**

**POST OCONV**

...../MT/JSONServices/oconv.ashx?Value=20015&Conversion=D4-

OCONV is a multi-value output formatting conversion which handles many different conversion code such as time, date, and decimal.

**AUTHORIZATION** API Key

This request is using API Key from collection **Manage 2000 REST Services**

**PARAMS**

Value	20015
-------	-------

**Example Request**

```
curl --location --request POST '...../MT/JSONServices/oconv.ashx?Value=20015&Conversion=D4-'
```

**Example Response**

```
json
{
  "oUdtConv": {
  }
}
```



# IMPORTING COLLECTIONS AND ENVIRONMENTS

---

- Sign up with Postman
- Create a blank workspace name Manage 2000
- Run M2K.REST.SERVICES
- Click on the orange “Run in Postman” button in the upper right.
- Select Postman for Web
- Select Manage 2000
- Click on orange “Import” button



---

# MANAGE 2000 REST FROM POSTMAN



# SELECTING THE MANAGE 2000 ENVIRONMENT

---

1. On Environments tab, Set Manage 2000 REST Environment as active
2. Set serviceroot variable to your webserver's main or train account website
3. On Collections tab, go to Manage 2000 REST Services > Login CID
4. Set variable UserId to your Manage 2000 web login
5. Set variable Password to your Manage 2000 web password
6. Click Send
7. Copy CIDToken result
8. on Environment tab, paste into Current value of the M2k\_Auth-Token variable
9. on the Collections tab, explore Manage 2000 REST Services methods



# M2K.REST.SERVICES CATEGORIES

---

1. Authentication
2. Application Server Cookies
3. Primary Key Selection
4. Data Validation and Formatting
5. Data Retrieval
6. Shopping Cart



# USING POSTMAN

---

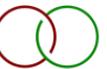
- Environments
  - Be sure to select the active environment
  - Make sure the « serviceroot » variable points to your webserver
- Collections
  - Expand Manage 2000 REST Services to types of services
  - Expand a particular type of service to see detail service endpoints
- Documentation
  - From a particular service endpoint, click on the ellipses and select « View documentation »
- Test Examples
  - Use the « Language » pull down to select the language the examples are presented in.



# EXAMPLE TESTING

---

1. Selecting Keys to a server cookie
2. Retrieving validation displays from the ids in the cookie



---

COPYING FROM  
POSTMAN TO  
VISUAL STUDIO  
PROJECT



# DROP CODE FROM POSTMAN INTO JAVASCRIPT

The image shows a side-by-side comparison of a REST client interface and its JavaScript implementation. On the left, a web browser displays a REST client interface for a REST API endpoint. The endpoint is `http://vm-mps-m2kwin11/Peter/MT/JSONServices/GetItemsAsJSON.ashx`. The request body is a JSON object with the following parameters: `{ "IDList": "&DisplayFileName=&FieldNames=Cust_Nbr&FieldNames=Name&FieldNames=City&FieldNames=State", "SelectStmnt": "WITH Name LIKE '...Office...'" }`. The response is a table of customer data.

Customer Nbr	Name	City	State	Zip Code
1016*1	Office Supply Co. Deutschland	Geoppingen	BAV	6632995
1040*1	Suburban Office Supply, UK	London		042996
1008*1	General Office Products, DL	Munich	BAV	5986222
1016	Office Supply Co.	ST LOUIS PARK MN		55426
1040	Suburban Office Supply	MINNEAPOLIS MN		55401-1278
1008	General Office Products	Murfesboro TN		37128

On the right, a remote desktop connection shows a Visual Studio Code editor with the JavaScript code that implements the REST client interface. The code uses the `fetch` API to send a GET request to the same endpoint as the REST client. The response is parsed as JSON and then formatted into an HTML table. The code includes error handling and console logging.

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <meta charset="utf-8" />
5   <title>Somewhere Outside of Manage 2000</title>
6   <script>
7     function DoFetch() {
8       console.log("doing the DoFetch function");
9       const RESTUrl = "http://vm-mps-m2kwin11/Peter/MT/JSONServices/GetItemsAsJSON.ashx";
10      const qsb4 = "SelectFileName=CM&SelectStmnt=";
11      const qsft = "&IDList=&DisplayFileName=&FieldNames=Cust_Nbr&FieldNames=Name&FieldNames=City&FieldNames=State";
12      const qs = qsb4 + document.getElementById('edtSelectStmnt').value + qsft;
13
14      fetch(RESTUrl + '?' + qs)
15        .then(response => {
16          if (!response.ok) { throw new Error('Invalid response from REST service' + response.status + ' ' +
17            return response.json()
18          })
19          .then(jsonResult => {
20            document.getElementById('divRESTQuery').innerHTML = '<h2>REST request to get data from ' + RESTUrl
21            const divRESTResults = document.getElementById('divRESTResults');
22            const tableHead = '<table><tr><th>Customer Nbr</th><th>Name</th><th>City</th><th>State</th><th>Zip
23
24            divRESTResults.innerHTML = tableHead + jsonResult.map(({ Cust_Nbr, Name, City, State, Zip }) => {
25              return `
26                <tr>
27                  <td>${Cust_Nbr}</td><td>${Name}</td><td>${City}</td><td>${State}</td><td>${Zip}</td>
28                </tr>
29              `;
30            }).join('') + '</table>';
31          })
32          .catch(console.warn);
33        }
34      </script>
35    </head>
36    <body>
37      <h1>From Somewhere Outside of Manage 2000</h1>
38      <div>
39        <span>Display CM :</span>
40        <input type="text" id="edtSelectStmnt" value="WITH Name LIKE 'Office...'" size=100 />
41        <input type="button" id="btnGo" value="Go" accesskey="g" onclick="DoFetch()" />
42      </div>
43      <div id="divRESTQuery"></div>
44      <hr/>
45      <div id="divRESTResults"></div>
46    </body>
47  </html>
48
```



# DROP CODE FROM POSTMAN INTO NODE.JS FILE

Example Request Item Validation ▾

```
javascript

var requestOptions = {
  method: 'POST',
  redirect: 'follow'
};

fetch("...../MT/JSONServices/ValidateItemService.ashx?FileName=IM&TableNbr=0&ItemID=110&Pre:
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', e))
```

[View More](#)

```
1 const serviceroot = "http://aus-dev-mnlab2022/CLASS_TRAIN";
2 const filename = process.argv[2];
3 const key = process.argv[3];
4
5 console.log('validate filename '+filename+' key '+key);
6
7 const requestOptions = {
8   method: "POST",
9   headers: {
10     "Content-Type": "application/x-www-form-urlencoded",
11     "M2k_Auth_Token": "1LR4VEWp76E%3d"
12   }
13 }
14
15 fetch(`${serviceroot}/MT/JSONServices/ValidateItemService.ashx?FileName=${filename}&TableNbr=0&ItemID=${key}`, requestOptions)
16   .then(response => response.json())
17   .then(result => console.log(JSON.stringify(result)))
18   .catch(error => console.log('error', error));
19
20
```



---

# QUESTIONS AND ANSWERS



NUGM 2024

THANK YOU

