**Sage 300 Web API – Using Postman**

In this exercise, you will learn how to make various Sage 300 Web API requests through a web API tester program called Postman. Before you begin, you should install Postman, a Chrome app, by visiting <https://chrome.google.com/webstore/search/postman> in a Chrome browser window. In addition to going through this tutorial, you may also want to import *Summit%20Demo.postman\_collection.json* (file located in the same folder as this document) into Postman to see the final results.

# Get Records

1. Start Postman and enter [**http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers**](http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers)in the request URL textbox.
2. Under Authorization, click the **Type** dropdown and select **Basic Auth**.
3. Enter **ADMIN** for both the **Username** and **Password**
4. Click **Send** and examine the result payload of customers
   * This is a single page (Max 100 count) of customers
5. Click **Save** and enter **Get Records** in the **Request Name** textbox
6. Enter **Sage Summit** in the **create new collection** textbox and click **Save**

# Get Records using a Range

1. Click the ellipsis “…” symbol next to the new request in the collections pane on the left and select **Duplicate**
2. Select the new copy and change the URL to [**http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers?$skip=5&$top=2**](http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers?$skip=5&$top=2)
3. Click **Send** and examine the result payload of customers
   * This results in the first 2 customers starting from the 6th record
4. Click **Save**
5. Click the ellipsis “…” symbol next to this request and select **Edit**.
6. Enter **Get Records using a Range** in the **Request Name** textbox and click **Update**

# Get Records using a Filter

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers?$filter=City eq 'San Francisco'** in the URL
2. Click **Send** and examine the resulting payload of customers
   * This results in a filtered result set of customers
3. Click **Save**
4. Edit the **Request Name** of this request and change it to **Get Records using a Filter**

# Get Records using a Complex Filter

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers?$filter=BalanceDueInCustomerCurrency gt 1000.00m and DateOfLastInvoice gt datetime'2016-12-31T12:00' and AccountType eq 'BalanceForward'**
2. Click **Send** and examine the resulting payload of customers
   * This results in a complex filtered result set of customers. Examine the way dates, decimals and enums are formatted in filter queries.
3. **Save** this request and rename it to **Get Records using a Complex Filter**

# Get a Record by Key

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers('1200')** as the URL
2. Click **Send** and examine the resulting single customer record
   * This results in a single customer record and is referred to as an entry.
3. **Save** this request and rename it to **Get a Record by Key**

# Create a Record

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers** as the URL
2. Change the request method from a **GET** to a **POST**
3. Select the **Headers** tab and enter **Content-Type** as the key and **application/json** as the value
4. Select the **Body** tab, click the **raw** radio button and enter:



1. Click **Send** and examine the resulting single customer record
   * The result is a single customer record reflecting what was created in Sage 300
2. **Save** this request and rename it to **Create a Record**

# Update a Record

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers('Z\_POSTMAN')** as the URL
2. Select the **Body** tab, click the **raw** radio button and enter:



1. Click **Send**. There is no response payload for this request.
2. **Save** this request and rename it to **Update a Record**

# Delete a Record

1. Duplicate this request and change the request method from a **POST** to a **DELETE**
2. Click **Send**. There is no response payload for a delete request.
3. **Save** this request and rename it to **Delete a Record**

# Create a Complex Record

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/InvoiceBatches** as the URL
2. Change the request method from a **DELETE** to a **POST**
3. Select the **Body** tab, click the **raw** radio button and enter:



1. Click **Send** and examine the resulting single AR invoice batch record
   * The result is a single AR invoice batch record reflecting what was created. Notice the new BatchNumber that was generated.
2. **Save** this request and rename it to **Create a Complex Record**

# Invoke Process Endpoint

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/CreateGLBatch($process)** as the URL
2. Select the **Body** tab, click the **raw** radio button and enter:



1. Click **Send**. There is no response payload for a process request.
2. **Save** this request and rename it to **Invoke Process Endpoint**

# Discover Available Resources

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/$metadata** as the URL
2. Change the request method from a **POST** to a **GET**
3. Click **Send**. The response is a listing of all endpoints and models exposed in AR.
4. **Save** this request and rename it to **Discover Available Resources**

# Dealing with Errors

1. Duplicate this request and enter **http://localhost/Sage300WebApi/-/SAMLTD/AR/Customers('1200')** as the URL
2. Change the request method from a **GET** to a **PATCH**
3. Select the **Body** tab, click the **raw** radio button and enter:



1. Click **Send**. The response a typical error payload.
2. **Save** this request and rename it to **Dealing with Errors**