

Knowledgebase > CustomerPortal (fka WebPortal) > Miscellaneous > Web API Example Usage with Postman

# Web API Example Usage with Postman

Megan De Freitas - 2024-11-22 - Miscellaneous

#### Notes

For purposes of this demonstration, the SalesPad WebAPI is installed locally on port 5501. This means that all example requests shown will start with http://localhost:5501. If following along with this documentation, replace this with the server name and port number for your particular installation.

References to the API Help Page are referring the help page which can be found at your installation location/Help (in our case, http://localhost:5510/Help).

SalesPad WebAPI supports JavaScript Object Notation (JSON). For more information on JSON, check out <a href="http://www.json.org/">http://www.json.org/</a>.

#### **Postman Rest Client**

SalesPad developers have found a tool named Postman to be indispensable in testing and developing the SalesPad WebAPI. Postman is a Chrome App, but can be run as a desktop application as well. You can download the Packaged App <u>here</u>.

You may use any tools / code of your choice to send requests to the SalesPad WebAPI, but for the purposes of this demonstration we will be using Postman.

#### Creating a Session / Logging In

**NOTE:** You must create a session before attempting to request anything from the API. If you do not send a valid Session- ID with subsequent requests, your requests will be denied. (It is also assumed that you will use a user account that has admin (or all) permissions for all request types.)

To create a session, you must send a request to GET api/Session. This request must contain an Authorization header for <u>Basic Authentication</u> (as specified on the API help page): Username and password should be in the format of username:password, using the colon as separator, prefixed with the keyword Basic and encoded in Base64.

In Postman, this is a simple process, as a Basic Auth tab is provided which will prompt you for Username and Password, and do the rest for you. Enter your user information, then click **Refresh headers**. This will create an Authorization header with the appropriate value:

Usemame	e sa			Note						
Password	ı			The authorizatio	n header will b dded as a cust	tom				
			Show Passwo	ord header.						
	Clear	Refresh he	aders	3						
ave helper data to reques	t 🕑									
ave helper data to reques	t 💌						GET	•	URL params	Headers (1
ave helper data to reques tp://localhost:5501/a wuthorization	t ₽	Basic c2	E6c2E=			Add preset 🕶	GET Manage p	• resets	URL params	Headers (1
Save helper data to reques ttp://localhost:5501/a Authorization -leader	t 🗹	Basic c2 Value	E6c2E=			Add preset •	GET Manage p	• resets	URL params	Headers (1

Next, enter your server name, port, and api/session (in our case, http://localhost:5501/api/session), make sure the Http Request Method is GET, and click **Send**:

Normal Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No environment •				
Useman	ne sa			Note				
Passwo	rd ••			The authorization header will be generated and added as a custom				
		(	Show Passwor	j header.				
	Clear	Refresh he	aders					
Save helper data to reque	st 🖉							
http://localhost:5501/	api/session	-1				GET T	2 params	Headers (1)
http://localhost:5501/ Authorization	api/session	-1 Basic c2	?E6c2E=	A	dd preset 🔻	GET THE Manage presets	2 params	Headers (1)
http://localhost:5501/ Authorization Header	api/session	Basic c2 Value	2E6c2E=	A	dd preset 👻	GET • • • • • • • • • • • • • • • • • • •	2 params	Headers (1)

The response should pop up below, and look something like this (If it doesn't look as nice, make sure the Pretty tab is selected):

Body	Cookies	Headers (15)	Tests	STATUS 20	о ок	TIME 49 ms	
Pr	etty Ra	aw Preview	23	٩	⊪	JSON ▼	
{							
ļ	"Session] "SystemGu	ID": "908a791	e-a4af-4d	7f-bb59-1d	66137	a4973",	
	"Secu	rity_Group":	"Admin",				
	"Secu	rity": null,					
	"Is_E	xternal_Grou	p": false	,			
	"Layo	ut_Path": ""	,				
	"User	FieldData":	null,				
	"User	FieldNames":	null,				
	"Core	SaveOnly": f	alse,				
	"Part	iallyLoaded"	: false,				
	"User	FriendlyName	": "System	m Group",			
	"Noti	fications":	nu11				

The response will contain information about the user, such as the Security Group it belongs to, Object and Property Permissions, Group Permissions, and any User-Customer associations. However, in our case, all we really care about is the SessionID value given, as this is our ticket to make further requests. This value must be used with a Session-ID header for all following requests:

Normal	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No environment •
Enter r	equest URL h	iere			
Sessio	on-ID		908a79f	fe-a4af-4d71	f-bb59-1d66137
Heade	er		Value		
		_		-	
Send	Previ	ew Pre-req	uest script	Tests	Add to collection

#### **Retrieve a Customer**

To retrieve Customer information, we can send a GET request to api/Customer endpoint. The API Help Page shows us that there are two kinds of GET requests we can make – one using OData, and one using the Primary Keys of the Object (in this case, the Customer\_Num):

# Customer

CRUD operations for Customer

API	Description
GET api/Customer	ODATA enabled get Customer requests.
GET api/Customer/{Customer_Num}	Get Customer requests.
DOST ani/Customor	Croate Oustamor objects

For our example, we will retrieve Customer Aaron Fitz Electrical, from the Sample Company setup provided by Dynamics GP.

We can do this using either endpoint:

• GET api/Customer/AARONFIT0001

OR

• GET api/customer?\$filter=Customer\_Num eq 'AARONFIT0001'

For the OData enabled endpoint, we simply filter to customers where the Customer\_Num is AARONFIT0001, and we end up with the same results. (See the API GettingStarted page for more information on OData).

For this request, you can use the Normal tab in Postman, as we don't need the Basic Authentication anymore. Begin by copying the SessionID value provided by the GET api/Session request, and paste it as the value for a Session-ID header (as shown previously). Then enter the URL you wish to use:

Normal	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No environr	ment 🕶				
http://lo	ocalhost:5501	/api/customer	AARONFIT(	0001				GET 🔻	URL params	Headers (1)
Sessio	on-ID		908a79	fe-a4af-4d7	f-bb59-1d66137		Add preset 🔻	Manage presets		
Heade	er		Value		E					
Send	Prev	iew Pre-req	uest <mark>scrip</mark> t	Tests	Add to collection					Reset

Again, make sure the Http Request Method is GET, and click **Send**. A successful response should look like the following:



All other GET requests should be made in a similar manner, using the SessionID created by the initial GET api/Session request. See the API Help Page for all available endpoints.

#### **Create a Customer**

To create a new Customer, we will use the POST Http Request Method. When you change this in Postman, an additional area for sending content with the request will pop up. We want to use the raw type, and it looks best when formatted as JSON.

(When you select JSON, you may notice that Postman will add an additional Content-Type header of application/json. Feel free to delete it or leave it as you see fit. This is normally what would describe the request body type, but since SalesPad WebAPI only supports JSON, it is not necessary (but also does not hurt anything).)

ormal	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No environment	•			
http://lo	calhost:5501	1/api/customer					POST -	1) params	Headers (2)
Sessio	n-ID		908a79	fe-a4af-4d7f-	bb59-1d66137	Add preset 🕶	Manage presets		
Conter	nt-Type		applicat	ion/json					
Heade	r		Value						
form-	data x-ww	vw-form-urlenco	ded raw	<b>→</b> 2	JSON (application/json)	•			
1					Text				
					Text (text/plain)				
					JSON (application/jso	on)			
					Javascript (applicatio	on/javascript)			
					XML (application/xml	I)			
					XML (text/xml)				
Send	<ul> <li>Prev</li> </ul>	riew Pre-req	uest script	Tests	HTML (text/html)				Reset

**NOTE:** Generally, a SalesPad Object can be created with as little as the Primary Key(s). However, we encourage you to provide as much information as possible for better accuracy and fewer errors.

For example, we can create a Customer by only giving the Customer Number we wish to use:

ormai	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No envir	onment 🕶					
nttp://lo	ocalhost:5501	/api/customer						POST	•	URL params	Headers (2)
Sessi	on-ID		908a79	fe-a4af-4d7f-b	b59-1d66137		Add preset 🔻	Manage p	resets		
Conte	nt-Type		applicat	ion/json		≥≑×					
Heade	er		Value			=					
form	data x-ww	w-form-urlenco	ded raw	binary	ISON (applicatio	n/json) 🔻					
1 - i 2	{ "Custome	r_Num": "TestC	ustomer",								
	1										

However, this Customer will only have a few default properties filled out; you will likely not be able to do anything useful with this Customer until you have updated it with more information.

Body	Cookie	s Hea	aders (15)	Tests	STATUS 201 Created TIME 129 ms
Pr	etty	Raw	Preview	23	Q <b>⊪</b> JSON▼
{					
	"Custo	mer_Nur	m": "Test	Customer	r",
	"Custo	mer_Nar	me": "",		
	"Custo	mer_Cla	ass": "",		
	"Corpo	rate_C	ustomer_N	lum": "",	,
	"Short	Name"	: "",		
	"State	ment_Na	ame": "",		
	"Prima	ry_Addı	r_Code":	"",	
	"Prima	ry_Bill	l_To_Addr	_Code":	"",
	"Prima	ry_Shi	p_To_Addr	_Code":	"",
	"State	ment To	o ∆ddr Co	de" · ""	

**NOTE:** For Customers, SalesDocuments, and SalesLineItems, auto-numbering can be accomplished by not providing the Primary Key you wish to be created for you. For example, we can POST a customer with just a name, and we will see that the Primary Key (Customer\_Num) property is automatically generated:

{
 "Customer\_Name": "Testing Customer Number Generation",
}

A successful response should look similar to the following:

Body	Cookies	Headers (15)	Tests	STATU	201 Created	d TIME 139 ms
Pr	retty Ra	aw Previev	2	٩	·	JSON ▼
{	"Customer	r_Num": "000	001 ",			
	"Customer	r <mark>_Name":</mark> "Te	sting Cu	stomer Nu	mber Gene	ration ",
	"Customer	r_Class": "	",			
	"Corporat	te_Customer_	Num": "	",		
	"Short_Na	ame": " ",				
	"Statemer	nt_Name": "	",			
	"Primary	_Addr_Code":	" ",			
	"Primary	Bill To Add	r Code"			

See the API Help Page for JSON samples which include all available properties for POSTing each Object type: POST api/Customer

POST api/Customer	Create Customer objects
GET ani/Customer/ICustomer_Num1/SalasP	Request Information
	Request body formats
	application/json, text/json
	Sample:
	<pre>1 "Customer_Num": "sample string 1", "Customer_Name": "sample string 2", "Customer_Class": "sample string 3",</pre>
	"Corporate_Customer_Num": "sample string 4", "Short_Name": "sample string 5", "Statement_Name": "sample string 6",
	"Primary_Addr_Code": "sample string 7", "primary_Bill_To Addr Code", "sample string 8"

#### **Create a Sales Document**

Creating a Sales Document is similar to creating a Customer. The minimum required properties are Sales\_Doc\_ID, Sales\_Doc\_Type, and Customer\_Num. (You may provide your own Sales\_Doc\_Num, or else it will be automatically generated for you.)

{
 "Sales\_Doc\_ID": "STDORD",
 "Sales\_Doc\_Type": "ORDER",
 "Customer\_Num": "AARONFIT0001",
 }

Make sure that the Http Request Method is set to POST:

ormal	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No envir	onment -				
http://lo	ocalhost:5501	/api/salesdocu	ment					POST V	URL params	Headers (2)
Sessio	on-ID		908a791	e-a4af-4d7	f-bb59-1d66137		Add preset 🕶	Manage presets		
Conte	Content-Type			application/json						
Heade	er		Value							
form	data x-ww	w-form-urlenco	ded raw	binary	JSON (application	n/json) 🔻				
1 * 2 3 <i>i</i> 4 5	<pre>{     "Sales_Doc     "Sales_Doc     "Customer_] }</pre>	_ID": "STDORD" _Type": "ORDER Num": "AARONFI	, ", T0001",							

As you can see, since we have not supplied a Sales\_Doc\_Num value, the response shows that the Sales Document Number has been automatically generated for us:

Body	Cookies	Headers (15)	Tests	STATUS 201 Crea	ted TIME 156 ms						
Pr	etty R	aw Preview		۹ 🌗	JSON -						
{											
	"Sales_Doc_Type": "ORDER",										
- (	"Sales_Doc_Num": "ORDST2242",										
	"Sales_D	oc_ID": "STD	ORD",								
	"Doc_Dat	e": "2015-02	-03T00:00	:00.000-05:00",							
	"Actual_	Ship_Date":	2015-02-	03T00:00:00.000	-05:00",						
	"Fulfill	ment_Date":	2015-02-	03T00:00:00.000	-05:00",						
	"Source"	: "",									
	"Sales_Batch": "ORDER",										
	"Customer_Num": "AARONFIT0001",										
	"Custome	r_Name": "",									
	"Bill To	Address Cod	e". ""								

#### **Create a Sales Line Item**

When creating a Sales Line Item, minimum required properties are Sales\_Doc\_Type, Sales\_Doc\_Num, and Item\_Number. Line\_Num and Component\_Seq\_Num can be provided, otherwise they will be automatically generated:

Add them to the LineItems property in Sales Document when POSTing the document (each Line Item Object in the LineItems array should at least contain the minimum required properties listed in the following point):

{
"Sales\_Doc\_Type": "ORDER",
"Sales\_Doc\_Num": "ORDST2242",
"Item\_Number": "A100",
}

Normal	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No e	environment 🔻				
http://lo	ocalhost:550	)1/api/salesline	item					POST V	URL params	Headers (2)
Sessi	on-ID		908a79	fe-a4af-4d7f	-bb59-1d66	137	Add preset 🕶	Manage presets		
Conte	nt-Type		applicat	ion/json					-	
Heade	er		Value							
form	-data x-w	ww-form-urlenc	oded raw	binary	JSON (applic	ation/json) 🔻				
1 * 2 3 <i>i</i> 4 5	K "Sales_Dc "Sales_Dc "Item_Num }	<_Type": "ORDE c_Num": "ORDST ber": "A100", view Pre-ren	R", 2242", quest script	Tests	Add to collec	tion				Reset
Body	Coo Pretty	kies He Raw	Prev	4) Tes	sts	STATU	s 201 Create	d TIME 19	0 ms	
{	"Sal "Sal "Lin "Com "Sou "Ite "Ite "Ite "Is_ "Is_	es_Doc_1 es_Doc_N e_Num": ponent_9 rce": "( m_Number m_Descri Non_Inve Dropship	Type": Num": " 16384, Seq_Num Open", ": "A1 iption" entory" o": fal	"ORDER ORDST2 ": 0, 00 ", : "Aud : fals	", 242 ", io Sys e,	, stem ",				
	"UOf "Uni "Bac	M_Schedu t_Of_Mea	ile": " asure": "Fack	PHONE "Each	1-10 ' ",	· ,				

The document and line items will be automatically updated with price, etc.

#### **Updating a Sales Document**

Updating Sales Document (or any other Object) properties is straight-forward. Simply look up the Object you wish to update, and send a PUT request to the specified URL with the properties you wish to update and their new values.

For example, to update a Sales Document, we can look up the PUT request URL format in the API Help Page:

GET apiroalesDocument/(oales\_Doc\_Type)/(oales\_Doc\_ivum)

## PUT api/SalesDocument/{Sales\_Doc\_Type}/{Sales\_Doc\_Num}

#### DELETE ani/SalesDocument//Sales\_Doc\_Tyne\//Sales\_Doc\_Num\

We can see that we have to send the Sales\_Doc\_Type and Sales\_Doc\_Num Primary Keys so that the API knows which object to update. In Postman, change the Http Request Method to PUT, fill in the URL, and supply the properties you wish to update. In our example, we will update the Customer\_PO\_Num:

# { "Customer\_PO\_Num": "New PO 1234", }

Normal	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	No environ	ment 🕶				
http://lo	ocalhost:5501	I/api/salesdocu	ment/ORDE	R/ORDST2	242			PUT •	URL params	Headers (2)
Sessio	on-ID		908a79	e-a4af-4d7	f-bb59-1d66137		Add preset 🕶	Manage presets		
Conte	nt-Type		applicat	ion/json						
Heade	er		Value							
form	data x-ww	vw-form-urlenco	ded raw	binary	JSON (application/j	son) 🔻				
1 * i 2 3	{ Custome	r_PO_Num": "New	v PO 1234",							
	1									
										ĥ
Send	<ul> <li>Prev</li> </ul>	iew Pre-requ	uest script	Tests	Add to collection					Reset

In the response, you can see that the Customer\_PO\_Num has been updated:



### Forwarding a Sales Document

Provided that you have workflow set up in SalesPad, it is easy to forward documents, as shown in the API Help Page:

## Workflow

Provides the ability to forward Sales Documents through SalesPad Workflow

API	Description
PUT api/Workflow/{Sales_Doc_Type}/{Sales_Doc_Num}/Forward	Forward a Sales Document to the next queue in workflow

In Postman, make sure the Http Request Method is set to PUT, and fill in the URL with the document you wish to forward (Note that no body is required in this request):

ormal	Basic Auth	Digest Au	rth OA	uth 1.0	OAuth 2.0	۲	No environ	ment 🔻				
http://lo	calhost:55	)1/api/Workf	low/ORE	ER/OR	DST2242/F	orward				PUT 🔻	URL params	Headers (2)
Sessio	n-ID			908a79f	e-a4af-4d7	'f-bb59-1	d66137		Add preset 🔻	Manage presets		
Conte	nt-Type			applicati	on/json							
Heade	r			Value								
form-	data x-w	ww-form-urle	encoded	raw	binary	JSON (a	pplication/j	son) 🔻				
1												
												A
Send	- Pre	view Pre	-request	script	Tests	Add to c	ollection					Reset

If successful, you will see a response similar to the following:

Body	Cooki	es He	aders (15) T	ests	STATUS	200 OK	TIME 61 ms							
P	retty	Raw	Preview	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	٩	<u>ا</u>	JSON ▼							
{	"Statu	usCode"	: "ОК",											
	"Erroi	"Code" : "CodeMe	ø, ssage": "No	Error",										
-	"Messa	ages":	[											
	"_			\ ORDS	T2242\ -			-\ The de	ocument wil	l be moved	to the	REVIEW	queue. \	\ "
	1													
}	-													

If the document is already in the last queue, you will likely get a message similar to the following:

Body	Cookie	s He	aders (15)	Tests	STATUS	500 Inte	rnal Server Erro	DI TIME	85 ms			
PI	retty	Raw	Preview	22	٩	<b>.</b>	JSON ▼					
{	"\$+>+14	Code	. "Inter	Server	Ennon"							
	"Error(	ode":	1000,	INTRE VEL	LITON ,							
	"Error(	odeMe	ssage": '	Exception	on - Gene	ral",						
_	"Messag	ges":	[									
	"Do	cumen	t Not For	warded.	Possible	reason	s include	being i	n the l	ast queue	of the	workflow."
	]											
}												

**Deleting a Sales Document** 

Deleting a Sales Document (or any other Object) is also straight-forward. Simply use the API Help Page to look up the Object you wish to delete, and send a DELETE request to the specified URL.

POT appoalesDocument/sales\_Doc\_Type//sales\_Doc\_ivum}

DELETE api/SalesDocument/{Sales\_Doc\_Type}/{Sales\_Doc\_Num}

OFT ani/CalasDecument/Leak/(Calas\_Dec\_Tupe)/(Calas\_Dec\_Num)

In Postman, flip the Http Request Method to DELETE, and fill in the URL with the Object you wish to delete (Note that no body is required in this request):

ormal	Basic Aut	h Dig	est Auth	OAuth 1.0	OAuth 2.0	No enviror	nment 🕶				
nttp://lo	calhost:55	01/api/\$	SalesDocu	ument/ORD	ER/ORDST	2242			DELETE V	URL params	Headers (2
Sessio	n-ID			908a79	fe-a4af-4d	f-bb59-1d66137		Add preset 👻	Manage presets		
Conter	nt-Type			applica	tion/json						
Heade	r			Value			<b></b>				
form-	data x-i	www-for	m-urlenco	ded raw	binary	JSON (application/	json) 🔻				
1											
Send	▼ Pr	eview	Pre-req	uest script	Tests	Add to collection					Reset

If successful, the response message should look like the following:

Body	Cooki	ies Hea	ders (15)	Tests	9	STATUS	200 OK	TIME 108 m	IS
Pr	etty	Raw	Preview	25		٩	<b>.</b>	JSON -	
{									
	"Statı	usCode":	"ОК",						
	"Error	rCode" :	0,						
	"Error	rCodeMes	sage": "	No Erro	r",				
_	"Messa	ages": [							
	"S	alesDoc	ument De	leted."					
	]								
}									

Deleting a Session / Logging Out

It is a good idea to delete your session when finished making requests. SalesPad WebAPI will automatically delete sessions older than 15 minutes, but during that time period you will still be logged in, and consuming a seat of SalesPad. Deleting your session will also delete any document locks you may have acquired during your session.

To delete your session, simply make a request to DELETE api/Session, with the Session-ID header as usual (again, no body is required):

rmal	Basic Auth	Digest Auth	OAuth 1.0	OAuth 2.0	● No environment ●				
ttp://lo	ocalhost:550	1/api/session					DELETE V	URL params	Headers (2)
essi	on-ID		908a79	fe-a4af-4d7f	f-bb59-1d66137	Add preset 🔻	Manage presets		
Conte	nt-Type		applica	tion/json					
leade	er		Value		#				
form	-data x-w	ww-form-urlenco	ded raw	binary	JSON (application/json) 🔻				
1									

If successful, you should get a response similar to the following:

Body	Cook	ies He	aders (15)	Tests		STATUS	200 OK	TIME 22 ms
Pr	etty	Raw	Preview	2	3	٩	<b>.</b>	JSON -
{								
	"Stat	usCode'	: "ОК",					
	"Erro	rCode":	0,					
	"Erro	rCodeMe	ssage":	"No Err	or",			
_	"Mess	ages":	[					
	"5	Session	and Sale	s Docu	ment	Locks	deleted	a."
	]							
}								