**Main call parameters: assignment group=4d93ef121bcbe810594e0f6cdc4bcbf7 & state not equal to 3:**

url = 'https://accruentseattleoemdev.service-now.com/api/now/table/x\_ctv\_fac\_work\_order?&assignment\_group=4d93ef121bcbe810594e0f6cdc4bcbf7&state!=3'

Action: If work order number does not exist in help desk, create new request.

Else, check if any info has changed & update request with creation of follow-up (internal) that generates email to request owner.

**Comments call parameters: Applies to work orders that already exist. Parameter is sys\_id of ALL work orders with state of not 3 & element=comments**

url = 'https://accruentseattleoemdev.service-now.com/api/now/table/sys\_journal\_field?sysparm\_query=element\_id= loop of all work order sys\_ids &element=comments'

Action: This call will need to run after the main call to check for any new comments in the open requests. Comments are not visible in the main call. All comments return and so we will only want to update the new comments. Comments that we have sent will also appear so that may be a duplication issue or I’m not sure if we can check that and ignore them as already existing.

**Patch Calls –** We will only be sending comments pertaining to open requests or closing open requests when they are complete

**Patch call to send comments. Url specifies table & specific work order sys\_id**

url = 'https://accruentseattleoemdev.service-now.com/api/now/table/x\_ctv\_fac\_work\_order/4657eb161bcbe810594e0f6cdc4bcbf1'

response = requests.patch(url, auth=(user, pwd), headers=headers, data="<request><entry><comments>Some actual comment text here. May include dates – these will all be treated as strings </comments></entry></request>")

Action: Call initiated when helpdesk user makes a follow-up comment that is flagged as ‘client facing’.

**Patch call to close a request. Url specifies table & specific work order sys\_id**

url = 'https://accruentseattleoemdev.service-now.com/api/now/table/x\_ctv\_fac\_work\_order/4657eb161bcbe810594e0f6cdc4bcbf1'

response = requests.patch(url, auth=(user, pwd), headers=headers, data="<request><entry><state>3</state><x\_ctv\_sc\_close\_code>SWAG+F Request Complete</x\_ctv\_sc\_close\_code><close\_notes>Systems Source has closed this request.</close\_notes></entry></request>")

3 parameters are required to close a request. The values we pass here will be the same for all requests:

State = 3

X\_ctv\_sc\_close\_code = SWAG+F Request Complete

Close\_note = Systems Source has closed this request.

Action: The Patch call to close a request will initiate when a helpdesk user enters a ‘completed date’ into the help desk.