1) SNMP WALK from API:

Expected input:

{

“FQDN” : str

"ip": str,

"snmpPort": int,

"snmpVersion": str

"snmpCommand": str,

"oid": str,

"outputFormat": str,

"peerIp": str,

"communityString": str,

"authenticationUsername": str,

"authenticationProtocol": str,

"authenticationPassphrase": str,

"encryptionProtocol": str,

"encryptionPassphrase": str

}

Expected output:

SNMP Command executed successfully or not. (JSON format)

Notes:

It would be very useful to run SNMP using the device's FQDN if the IP address is not specified in the input, with a priority given to searching by IP. Additionally, SNMP should ideally be executed using the credentials stored in SevOne, rather than requiring the user to input them manually.

2) SNMP TRACEROUTE from API:

Expected input:

{

“FQDN” : str,

"ip": str,

"peerIp": str

}

Expected output:

Traceroute command executed successfully or not. (JSON format)

Notes:

Would be very useful to run the SNMP by FQDN of the device if the IP is not detailed in the input, prioritizing the search by IP.

3) Get DEVICE MANAGER as a csv from API:

Expected input: Desired columns and subcolumns to export, i.e: Name, Objects, Metadata (and desired subfields).

Expected output: Device manager with all devices with the selected columns. Response in CSV format.

4) Get ALL policies:

Expected input:

1. Optional filter by id or name.
2. Page and size
3. Field for filter the response output fields.

Expected output:

{

"success": bool,

"policies": [

```
{
  "id": int,
  "name": str,
  "isDeviceGroup": bool,
  "groupIdList": [
    str
  ],
  "isMemberOfAny": bool,
  "pluginId": str,
  "objectTypeId": str,
  "objectSubTypeId": str,
  "severity":str,
  "triggerExpression": str,
  "clearExpression": str,
  "userEnabled": bool,
  "mailTo": str,
  "mailOnce": bool,
  "mailPeriod": int,
  "lastUpdated": int,
  "useDefaultTraps": bool,
  "useDeviceTraps": bool,
  "useCustomTraps": bool,
  "triggerMessage": str,
  "clearMessage": str,
  "appendConditionMessages": bool,
  "useDeviceWorkHours": bool,
  "description": str,
  "folderId": 2,
  "folderName": str,
  "isFolderEditable": bool,
  "technologyType": str,
  "hasRelativeTime": bool,
  "hasAbsoluteTime": bool,
```

```
    "type": str,
    "flowViewId": str,
    "flowDirection": str,
    "flowFilterId": str,
    "flowTriggerDuration": str,
    "flowClearDuration": str,
    "flowTriggerId": str,
    "flowClearId": str,
    "webhookTriggerId": str,
    "webhookClearId": str,
    "isServiceAlerts": str,
    "serviceProfileList": [
      str
    ],
    "flags": str
  }
        ],
        "total": str
        }
```


5) Get ALL object rules:

Expected input:

1. Optional filter by id or name.
2. Page and size
3. Field for filter the response output fields.

Expected output:


```
 "data": [
  {
    "id": str,
    "rank": str,
    "ruleType": str,
    "deviceGroupId": str,
```

```
    "objectTypeId": str,
    "objectTypeName": str,
    "objectSubtypeId": str,
    "comment": str,
    "enabled": bool,
    "subtypeName": str,
    "subtypeDescription": str,
    "pluginId": str,
    "pluginName": str,
    "deviceGroupName": str,
    "editable": bool,
    "nameMatch": bool,
    "descriptionMatch": bool,
    "conditions": [
     {
       "fieldName": str,
       "doMatch": stsr,
       "caseSensitive": str,
       "expression": str
     },
     {
       "fieldName": str,
       "doMatch": str,
       "caseSensitive": str,
       "expression": str
     }
    ],
    "nameExpression": str,
    "caseSensitive": bool,
    "descriptionExpression":str
   }
  ],
  "totalCount": int
}
```

6) CREATE object rules:

Expected input: Same input fields as UI

Expected output: Object rule successfully created or not.

7) Get OBJECT MANAGER as a csv from API:

Expected input:

- Desired filters (same as UI)
- Desired columns and subcolumns to export, i.e: Device, Object, Object type)

Expected output: Object manager response in CSV format.