

sevone-ingest-data

Ingest data [Device - Object - Indicators] into SevOne using V3 APIs

This project shares the code which can be used to ingest data into SevOne.

To ingest data, the following needs to be done

1. Update etc/config.json with the appliance details [Cluster Leader] "IPAddress": IP Address of the ClusterLeader "Username" : API Creds "Password" : API Cred password. You can enter the plain password for the first time as below

```
"Password": "<plainpassword",
```

The password will be encrypted upon running the code

"sshUserName": "", "sshPassword": ssh creds, You can enter the plain password for the first time as below

```
"sshPassword": "<plainpassword",
```

The password will be encrypted upon running the code "UseSSHKeys": 1 or 0 If the value is 1, sshkeys will be used to ssh. No username or password would be required. If value is 0, then username/pwd will be required to login "Type": "NMS"

If you want to ingest data for one device, use the below in your main function

```
objectDict = {
    "automaticCreation": True,
    "description": "Group members Count Metrics",
    "name": ShortDevGroupPath,
    "pluginName": "DEFERRED",
    "timestamps": [
        {
            "indicators": [
                {
                    "format": "GAUGE",
                    "name": "No of Stations",
                    "units": "Number",
                    "value": deviceGroupDict[group_id] ["NoOfStations"]
                }
            ],
            "timestamp": timestamp
        }
    ],
    "type": "Device Group Counts"
```

```

    }
    if objectDict["name"] != "":
        objectList.append(objectDict)
    logger.info(f"ObjectList: {json.dumps(objectList, indent=4)}")
    result =
    SevOne_appliance_obj.ingest_dev_obj_ind(automationDict["Name"], automationD
ict["IPToBeCreated"], objectList)

```

```json

If you want to ingest data for multiple devices, use the below in your main function

Create a DOI list as below

eg

```json

```

dev_ob_ind_list = [
    {
        "automaticCreation": True,
        "distributionOnAllPeers": True,
        "name": deviceName,
        "ip": deviceIp_to_be_created,
        "objects":
        [{
            "automaticCreation": True,
            "description": "Group members Count Metrics",
            "name": ShortDevGroupPath,
            "pluginName": "DEFERRED",
            "timestamps": [
                {
                    "indicators": [
                        {
                            "format": "GAUGE",
                            "name": "No of Stations",
                            "units": "Number",
                            "value": deviceGroupDict[group_id]["NoOfStations"]
                        }
                    ],
                    "timestamp": timestamp
                }
            ],
            "type": "Device Group Counts"
        }],
    },
    {
        "automaticCreation": True,
        "description": "Group members Count Metrics",
        "name": ShortDevGroupPath,
        "pluginName": "DEFERRED",
        "timestamps": [
            {
                "indicators": [
                    {
                        "format": "GAUGE",
                        "name": "No of Stations",
                        "units": "Number",
                        "value": deviceGroupDict[group_id]["NoOfStations"]
                    }
                ]
            }
        ]
    }
]

```

```

        }
    ],
    "timestamp": timestamp
}
],
    "type": "Device Group Counts"
}]
},
{
    "automaticCreation": True,
    "distributionOnAllPeers": True,
    "name": deviceName,
    "ip": deviceIp_to_be_created,
    "objects":
    [{
        "automaticCreation": True,
        "description": "Group members Count Metrics",
        "name": ShortDevGroupPath,
        "pluginName": "DEFERRED",
        "timestamps": [
            {
                "indicators": [
                    {
                        "format": "GAUGE",
                        "name": "No of Stations",
                        "units": "Number",
                        "value": deviceGroupDict[group_id] ["NoOfStations"]
                    }
                ],
                "timestamp": timestamp
            }
        ],
        "type": "Device Group Counts"
    },
    {
        "automaticCreation": True,
        "description": "Group members Count Metrics",
        "name": ShortDevGroupPath,
        "pluginName": "DEFERRED",
        "timestamps": [
            {
                "indicators": [
                    {
                        "format": "GAUGE",
                        "name": "No of Stations",
                        "units": "Number",
                        "value": deviceGroupDict[group_id] ["NoOfStations"]
                    }
                ],
                "timestamp": timestamp
            }
        ],
        "type": "Device Group Counts"
    }
]
}]

```

```
    }  
]
```

Call the below function in your main function

```
result = Sev0ne_appliance_obj.ingest_multi_dev_obj_ind(dev_ob_ind_list,  
max_threads=5):
```