Intel Accelerated Storage Manager

System Plugin REST API

INTEL CONFIDENTIAL

September 2017
Document Revision 1.3

Contents

1. Introduction 2
2. URI structure 2
3. Response handling 2
4. OAuth parameters 3
5. Resources 3
   5.1 Plugin version 3
   5.2 List of available plugins 3
   5.3 Host general system information 4
   5.4 Host CPU information 4
   5.5 Memory information 5
   5.6 Host device and partitions 5
   5.7 Host performance information 7

1. Introduction

This document is a reference for Intel ASM Service API. Scope of this API covers access to system resources of host machine where agent is installed.

2. URI structure

Access to resources is provided via URI paths. Client applications should send requests over the HTTP protocol using standard GET method.

Intel ASM URI structure:

```
[protocol]://[host]:[port]/v1/info/[resource name]
```

Where:

[protocol]	Communication protocol – http/https.
[host]	Host machine ip address or name where agent is installed.
[port]	Port number on which agent service is listening.
v1	Stands for first version of API definition.
info	Namespace for all resources related to host system information.
[resource name]	Name for resource to be returned.

3. Response handling

REST API returns responses in wrapper format presented below.

Response format			

data	{Object} or {Array}	Contains resource response as described in this documentation. Can be either an array or an object.	Error response.
			{
			"data": null,
			"status": {
			"errorId": "INVALID_PARAMETER",
			"errorMessage": "Request could not be parsed,
			"errorDescription": "Incorrect Json formatting",
			"success": false
			}
			}

status	{Object}	Contains the request status object. If request failed, additional error information is attached.	

{}.errorId (Optional)	{String}	Identifier of thrown error.	Example for successful request:
			{
			"data": {
			"agentVersion": "1.1.0.0",
			"pluginVersion": "1.0.0.0"
			},
			"status": {
			"success": true

{}.errorMessage (Optional)	{String}	General error message.	
{}.errorDescription (Optional)	{String}	Extended error description.	

{}.success	{Boolean}	Request status, true if requests succeeded, otherwise false.	} }

4. OAuth parameters

To get access to System Plugin resources the OAuth scope parameter must contain value "info" - *scope="info"*.

5. Resources

Below is a list of all available resources for Intel ASM service.

5.1 Plugin version

```
[GET] /v1/info/version
```

Gets version of currently installed agent service.
Example:
curl http://127.0.0.1:8080/v1/info/version

Response {Object}

agentVersion	{String}	Installed Intel ASM service version. Using the following structure MAJOR.MINOR.BUILD.REVISION
pluginVersion	{String}	Installed system plugin version. Using the following structure MAJOR.MINOR.BUILD.REVISION.


```
{
  "data":
  {
    "agentVersion": "1.1.12.0",
    "pluginVersion": "1.0.1.0"
  },
  "status": {
    "success": true
  }
}
```


5.2 List of available plugins

```
[GET] /v1/info/products
```

Get all services available on this node according to plugins loaded from agent and products plugins directories.
Example:
curl http://127.0.0.1:8080/v1/info/products

Response {Array}

name	{String}	Service name.
id	{String}	Service identifier.
state	{String}	Plugin state [ENABLED, DISABLED]. If plugin cannot communicate with product, this parameter will be DISABLED, otherwise ENABLED.
version	{String}	Plugin version.


```
{
  "data": [
  {
    "name": "Intel(R) Product Software",
    "id": "intel_product",
    "state": "ENABLED",
    "version": "1.6.0.0"
  }],
  "status": {
    "success": true
  }
}
```

5.3 Host general system information

```
[GET] /v1/info/system
```

System basic information, like OS, architecture and host name.
Example:
curl http://127.0.0.1:8080/v1/info/system

Response {Object}

type	{String}	System type ["WINDOWS" | "LINUX"].
arch	{String}	System architecture ["x86" | "x64"].
name	{String}	Full system name in human readable format.
release	{String}	System release.
hostname	{String}	Computer name.
domain	{String}	Computer domain. If computer is not a part of a domain, the result should be an empty string.
serialNumber	{String}	Motherboard serial number used to identify machine.

Windows example:

```
{
  "data":
  {
    "type": "WINDOWS",
    "arch": "x64",
    "name": "Windows 8.1 Enterprise",
    "release": "6.3.9600",
    "hostname": "srv-db",
    "domain": "srv.lab.comp.com",
    "serialNumber": "23441214323"
  },
  "status": {
    "success": true
  }
}
```

Linux example:

```
{
  "data":
  {
    "type": "LINUX",
    "arch": "x64",
    "name": "SUSE Linux Enterprise Server
12 SP1",
    "release": "3.12.51-60.20-default",
    "hostname": "lnx-srv-db",
    "domain": "",
    "serialNumber": "23441214323",
  },
  "status": {
    "success": true
  }
}
```

5.4 Host CPU information

```
[GET] /v1/info/cpu
```

List of physical CPUs and its' usage information.
Example:
curl http://127.0.0.1:8080/v1/info/cpu

Response {Array}

cpuId	{String}	CPU identifier.
vendorId	{String}	CPU vendor id.
modelName	{String}	CPU model name.

```
{
   "data:": [
   {
     "cpuId": "CPU0",
     "vendorId": "GenuineIntel",
```

frequency	{Number}	CPU nominal frequency (MHz).
coresNumber	{Number}	Number of cores.
procNumber	{Number}	Number of logical processors.


```
    "modelName":
      "Intel® Xeon® CPU E5-2697 v2 @ 2.70GHz",
    "frequency": 2709,
    "coresNumber": 2,
    "procNumber": 4
  }],
  "status": {
    "success": true
  }
}
```


5.5 Memory information

[GET] /v1/info/memory

System memory and usage information.
Example:
curl http://127.0.0.1:8080/v1/info/memory

Response {Object}

memory	{Array}	Array of physical memory objects.
[].bankLabel	{String}	Physical bank label.
[].capacity	{String}	Physical memory capacity (bytes).
[].deviceLocator	{String}	Label of physical memory socket.
[].manufacturer	{String}	Manufacturer name. If it's unknown, the result is "UNKNOWN".
[].type	{String}	Memory type see http://schemas.dmtf.org/wbem/cim-html/2/CIM_PhysicalMemory.html


```
{
 "data":
 {
   "memory": [
   {
     "bankLabel": "BANK 0",
     "capacity": "8589934",
     "deviceLocator": "ChannelA-
DIMM0",
     "manufacturer": "Hyundai",
     "type": "RAM"
   }],
 },
 "status": {
   "success": true
 }
}
```


5.6 Host device and partitions

[GET] /v1/info/device

Returns list of all disks with partitions from the machine.
Example:
curl http://127.0.0.1:8080/v1/info/device

Response {Array}

device	{String}	Device identifier. On Linux it is TEMPORARY representation within /dev pseudo file system. On Windows it is the unique name of physical device.
serialNumber	{String}	Device serial number.
model	{String}	Device model.
type	{String}	Device type ["SSD"|"HDD"|"NAS"|"UNKNOWN"|"NVME"|"NONE"].

Windows Example


```
{
 "data": [
 {
   "device": "\\.\PHYSICALDRIVE0",
   "serialNumber": "CVPO044000B",
   "model": "INTEL SSDSA2M120G",
   "type": "SSD",
   "size" "120031511040",
   "partitions": [
   {
     "name": "C:",
     "label": "Windows",
```

size	{String}	Size of the disk drive in bytes.
partitions (Optional)	{Array}	Array of disks partitions objects.
[].label	{String}	Partition label.
[].name	{String}	Partition name.
[].mountPoints (Optional)	{Array}	String array of places where partition is logically attached.
[].fileSystem	{String}	File system name.
[].size	{String}	Partition size (bytes).
[].freeSpace	{String}	Free partition space left (bytes).
[].system	{Boolean}	Determines if partition is a system partition. On Linux, partition with mount point "/".


```
      "mountPoints": [
         "C:",
         "E:\System"
       ],
      "fileSystem": "ntfs",
      "size": "118987599872",
      "freeSpace": "7987599865",
      "system": true
    }],
  }
  "status": {
    "success": true
  }
}
```


Linux Example


```
{
 "data": [
 {
   "device": "/dev/sda",
   "serialNumber": "CVPO044000BW120",
   "model": "INTEL SSDSA2M120G2",
   "type": "SSD",
   "size": "120031511040",
   "partitions": [
   {
     "label": "",
     "name": "/dev/sda1",
     "fileSystem": "ntfs",
     "size": "1000204886016",
     "freeSpace": "0",
     "system": false
   },
   {
     "label": "",
     "name": "/dev/sda2",
     "fileSystem": "fat16",
     "size": "104857600000",
     "freeSpace": "0",
     "system": false
   },
   {
     "label": "",
     "name": "/dev/sda3",
     "mountPoints":
     [
       "/home/tuser/data"
     ],
     "fileSystem": "xfs",
     "size": "524288000",
     "freeSpace": "404288000",
     "system": false
   },
   {
     "label": "",
     "name": "/dev/sda4",
     "fileSystem": "",
     "size": "1024",
     "freeSpace": "0",
     "system": false
   },
   {
     "label": "",
     "name": "/dev/sda5",
     "fileSystem": "lvm2_member",
     "size": "685104955392",
     "freeSpace": "0",
     "system": false
```

```
    }],
  },
  {
   "device": "/dev/rhel",
   "model": "LVM VOLUME GROUP",
   "serialNumber": "HO0ddh-CxKp-S5Wd",
   "size": "140481386427152",
   "type": "NONE"
   "partitions": [
    {
      "fileSystem": "xfs",
      "freeSpace": "0",
      "label": "",
      "mountPoints": [
        "/"
      ],
      "name": "/dev/mapper/rhel--root",
      "size": "52038729728",
      "system": true
    },
    {
      "fileSystem": "swap",
      "freeSpace": "0",
      "label": "",
      "mountPoints": [
        "SWAP"
      ],
      "name": "/dev/mapper/rhel--swap",
      "size": "1073741824",
      "system": false
    }]
  },
  {
   "device": "/dev/sdb",
   "serialNumber": "CVPO040BW12",
   "model": "INTEL SSDSA2M120G2",
   "type": "SSD",
   "size": "120031511040",
  },
  "status": {
    "success": true
  }
}
```


5.7 Host performance information

```
[GET] /v1/info/performance
```

System performance information like cpu load and memory usage. Data parameters allows to limit response to queried data type only.

Example:

curl http://127.0.0.1:8080/v1/info/performance
curl http://127.0.0.1:8080/v1/info/performance/cpu
curl http://127.0.0.1:8080/v1/info/performance/memory
curl http://127.0.0.1:8080/v1/info/performance/partitions

```
Response {Object}
```

partitions	{Array}	List of partitions. See "Host device and partitions" section.	Example without data parameter
[].name	{String}	Partition name.	


```
{
 "data":
   {
     "partitions": [{
```

[].percentFreeSpace	{Number}	Partition free space in percent.
cpu	{Object}	CPU information object.
{}.total	{Number}	The average load of all CPUs in percent.
{}.cpus	{Array}	Array of physical processors.
{}.[].cpuId	{String}	Identifier of physical processor.
{}.[].frequency	{Number}	CPU current frequency (MHz).
{}.[].logicalProcessors	{Array}	Array of logical processors.
{}.[].[].id	{Number}	Logical processor identifier.
{}.[].[].value	{Number}	The average utilization of logical processor in percent.
memory	{Object}	Memory information.
{}.availableMemory	{String}	Current available physical memory in bytes.
{}.cachedMemory	{String}	Current cached memory in bytes. Memory user by kernel buffers and page cache and slabs (Linux). Number of bytes currently being used by the file system cache (Windows).
{}.memoryUsage	{Number}	Memory usage in percent. On Windows it's usage of committed virtual memory.


```
      "name" : "C:",
      "percentFreeSpace": 41
    }],
    "cpu":
    {
      "cpus": [
      {
        "cpuId": "CPU0",
        "frequency": 2709,
        "cores": [
        {
          "id": 0,
          "value": 10
        },
        {
          "id": 1,
          "value": 14
        },
        {
          "id": 2,
          "value": 11
        }],
      }],
      "total": 12
    },
    "memory": {
      "availableMemory": "9414782976",
      "cachedMemory": "5658992",
      "memoryUsage": 14
    }
  },
  "status": {
    "success": true
  }
}
```


Example for memory:


```
{
 "data":
 {
   "memory": {
     "availableMemory": "9414782976",
     "cachedMemory": "5658992",
     "memoryUsage": 14
   }
 },
 "status": {
   "success": true
 }
}
```


Example for partitions (Linux):


```
{
 "data":
 {
   "partitions": [
   {
     "name": "/dev/sda1",
     "percentFreeSpace": 0,
   },
   {
     "name": "/dev/sda2",
     "percentFreeSpace": 0,
   },
   {
     "name": "/dev/sda3",
     "percentFreeSpace": 80,
   },
```

```
    {
      "name": "/dev/sda4",
      "percentFreeSpace": 0,
    },
    {
      "name": "/dev/sda5",
      "percentFreeSpace": 0,
    },
    {
      "name": "/dev/mapper/rhel-swap",
      "percentFreeSpace": 0,
    },
    {
      "name": "/dev/mapper/rhel-root",
      "percentFreeSpace": 76,
    },
    {
      "name": "/dev/mapper/rhel-home",
      "percentFreeSpace": 68,
    }]
  },
  "status": {
    "success": true
  }
}
```


Example for cpu:


```
{
 "data":
 {
   "cpu":
   {
     "cpus": [
     {
       "cpuId": "CPU0",
       "frequency": 2709,
       "logicalProcessors": [
       {
         "id": 0,
         "value": 10
       },
       {
         "id": 1,
         "value": 14
       },
       {
         "id": 2,
         "value": 11
       }],
     }],
     "total": 12
   }
 },
 "status": {
   "success": true
 }
}
```