Incito Cloud IaaS ICSMAPI reference

灵动云计算-灵动系统监控API

参考手册

目录

[概述 4](#_Toc406675118)

[API 说明 4](#_Toc406675119)

[文档修订记录 4](#_Toc406675120)

[Package phyMachine 5](#_Toc406675121)

[Class PhyCpuUtil 5](#_Toc406675122)

[初始化方法( \_\_init\_\_) 5](#_Toc406675123)

[类实例化 5](#_Toc406675124)

[getMonitorKeys 6](#_Toc406675125)

[getValueByMonitorKey 7](#_Toc406675126)

[getTrendsValue 8](#_Toc406675127)

[getCpuSystemUtil 10](#_Toc406675128)

[getPeriodCpuSystemUtil 11](#_Toc406675129)

[getCpuUserUtil 12](#_Toc406675130)

[getCpuIdleUtil 13](#_Toc406675131)

[getCpuDiskIOUtil 14](#_Toc406675132)

[getProcessLoad1min 15](#_Toc406675133)

[getProcessLoad5min 16](#_Toc406675134)

[getProcessLoad15min 17](#_Toc406675135)

[Class PhySystemStatus 19](#_Toc406675136)

[初始化方法( \_\_init\_\_) 20](#_Toc406675137)

[类实例化 20](#_Toc406675138)

[getPhySystemUptime 20](#_Toc406675139)

[getPhySystemLocaltime 21](#_Toc406675140)

[getPhySystemLoggedInUser 22](#_Toc406675141)

[getPhySystemMaxOFile 23](#_Toc406675142)

[getPhySystemMaxProcesses 24](#_Toc406675143)

[getPhyTotalMemory 25](#_Toc406675144)

[getPhyAvailableMemory 26](#_Toc406675145)

[getPhyTotalRootDiskSize 27](#_Toc406675146)

[getPhyAvailableRootDiskSize 28](#_Toc406675147)

[getPhyAvailRootDiskPercent() 29](#_Toc406675148)

[getPhyNetworkTotal 30](#_Toc406675149)

[getPhyNetworkIn 31](#_Toc406675150)

[getPhyNetworkOut 32](#_Toc406675151)

[getPhyAgentPing 33](#_Toc406675152)

[getPhyMd5Passwd 34](#_Toc406675153)

[getPhyCommonStatus 35](#_Toc406675154)

[getMonitorExtendDefine 36](#_Toc406675155)

[Package phyService 37](#_Toc406675156)

[Class ServiceStatus 37](#_Toc406675157)

[初始化方法( \_\_init\_\_) 37](#_Toc406675158)

[类实例化 37](#_Toc406675159)

[getMonitorService 38](#_Toc406675160)

[getServiceStatusByKey 39](#_Toc406675161)

[getWebStatus 40](#_Toc406675162)

[getMongodbStatus 41](#_Toc406675163)

[getNovaNetworkStatus 42](#_Toc406675164)

[getMemcacheStatus 43](#_Toc406675165)

[getCommonServiceStatus 44](#_Toc406675166)

[getPingStatus 45](#_Toc406675167)

[getSSHStatus 46](#_Toc406675168)

[getHttpStatus 47](#_Toc406675169)

[getSmtpStatus 48](#_Toc406675170)

[getPop3Status 49](#_Toc406675171)

[getCurrentLoadStatus 50](#_Toc406675172)

[getRootPartitionStatus 51](#_Toc406675173)

[getUserDefinedStatus 52](#_Toc406675174)

[Package virMachine 53](#_Toc406675175)

[Class VirNovaUtil 53](#_Toc406675176)

[初始化方法( \_\_init\_\_) 53](#_Toc406675177)

[类实例化 53](#_Toc406675178)

[getAllMeters 54](#_Toc406675179)

[getMeterNameByResId 55](#_Toc406675180)

[getTrendsData 56](#_Toc406675181)

[getDaysSample 58](#_Toc406675182)

[getCpuUsedTime 59](#_Toc406675183)

[getDiskReadBytesRate 60](#_Toc406675184)

[getCpuUtil 61](#_Toc406675185)

[getDiskReadBytes 62](#_Toc406675186)

[getDiskWriteBytes 63](#_Toc406675187)

[getNetworkInBytes 64](#_Toc406675188)

[getNetworkOutBytes 65](#_Toc406675189)

[Class VirGlanceUtil 66](#_Toc406675190)

[初始化方法( \_\_init\_\_) 67](#_Toc406675191)

[类实例化 67](#_Toc406675192)

[getCommonService 67](#_Toc406675193)

[getGlanceImageExists 68](#_Toc406675194)

[getGlanceImageSize 69](#_Toc406675195)

[getGlanceImageUpdate 70](#_Toc406675196)

[getGlanceImageDelete 71](#_Toc406675197)

[getGlanceImageDownload 72](#_Toc406675198)

[getGlanceImageServe 73](#_Toc406675199)

## 概述

ICSMAPI 包含对物理机，虚拟机， 服务资源监控.

整合zabbixAPI，nagiosAPI， ceilometerAPI

使用开发语言: Python

## API 说明

ICSMAPI(Incito cloud system monitor api)

灵动云系统监控API

包名修改下，隐藏底层细节

X,y,z 关系

## 文档修订记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 修订时间 | 修订方式 | 修改人 | 审核人 | 备注 |
| 1 | 2014-10-13 | 新增 | 崔明文 |  | 文档初稿 |
| 2 | 2014-10-15 | 修改 | 崔明文 |  | 添加返回值说明信息 |
| 3 | 2014-10-16 | 修改，新增 | 崔明文 |  | 取消区域id 参数.调整返回值，新增一段时期监控api |
| 4 | 2014-10-19 | 修改 | 崔明文 |  | 修改PythonAPI 类似Java 风格的API, 添加包的概念 |
| 5 | 2014-10-24 | 修改 | 崔明文 |  | 修改包名，类名，隐藏细节 |
| 6 | 2014-11-18 | 新增 | 崔明文 |  | 添加phyService 包中getMonitorService，getServiceStatusByKey |
| 7 | 2014-11-20 | Add | 崔明文 |  | 添加getTrendsData，修改**getDaysSample** |

## Package phyMachine

phyMachine package包含的类有PhyCpuUtil，PhySystemStatus, PhyBase

### Class PhyCpuUtil

PhyCpuUtil 继承 PhyBase

类PhyCpuUtil 包含方法摘要：

|  |  |  |
| --- | --- | --- |
| 方法名 | 返回值 | 方法说明 |
| **getMonitorKeys** | list | 获取所有监控的key 值 |
| **getValueByMonirotKey** | map | 根据监控key 获取监控值 |
| **getTrendsValue** | list | 获取一段时期内的值 |
| getCpuSystemUtil | map | 获取Cpu系统使用率 |
| getPeriodCpuSystemUtil | list | 获取一段时间Cpu 系统使用率 |
| getCpuUserUtil | map | 获取Cpu用户使用率 |
| getCpuIdleutil | map | 获取Cpu 空闲使用率 |
| getCpuDiskIOUtil | map | 获取Cpu磁盘IO负载使用率 |
| getProcessLoad1min | map | 获取单个Cpu 平均1分钟负载 |
| getProcessLoad5min | map | 获取单个Cpu 平均5分钟负载 |
| getProcessLoad15min | map | 获取单个Cpu 平均15分钟负载 |

### 初始化方法( \_\_init\_\_)

调用父类(PhyBase) 的初始化方法, 设置父类实例属性hostip (String), 调用父类方法getHostId 设置本类实例属性hid (String)

def **\_\_init\_\_**(*self*, hostip=*None*):

super(PhyCpuUtil,*self*).\_\_init\_\_(hostip=hostip)

*self*.hid = *self*.getHostId(hostip)

### 类实例化

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| hostip | string | None | 主机ip地址 |

phyCpuInstance = PhyCpuUtil(hostip = ’10.66.49.19’)

### getMonitorKeys

#### 功能描述:

根据所有监控的keys( 监控项目列表)

#### 参数：

无

#### 返回值：

list

获取当前IP 所有的监控列表

#### 定义:

getMonitorKeys(self)

#### 样例：

phyInstance = PhyCpuUtil(hostip = ’10.66.49.176’)

rlt = phyInstance. getMonitorKeys()

print rlt

[u'zabbix.getBackupStatus', u'agent.ping', u'vm.memory.size[available]', u'vfs.file.cksum[/etc/passwd]', u'system.cpu.switches', u'system.cpu.util[,idle]', u'system.cpu.util[,interrupt]', u'system.cpu.util[,iowait]', u'system.cpu.util[,nice]', u'system.cpu.util[,softirq]', u'system.cpu.util[,steal]', u'system.cpu.util[,system]', u'system.cpu.util[,user]', u'vfs.fs.size[/,free]', u'vfs.fs.size[/boot,free]', u'vfs.fs.size[/bakdata,free]', u'vfs.fs.size[/,pfree]', u'vfs.fs.size[/boot,pfree]', u'vfs.fs.size[/bakdata,pfree]', u'vfs.fs.inode[/,pfree]', u'vfs.fs.inode[/boot,pfree]', u'vfs.fs.inode[/bakdata,pfree]', u'system.swap.size[,free]', u'system.swap.size[,pfree]', u'system.boottime', u'system.localtime', u'system.hostname', u'agent.hostname', u'net.if.in[vlan103]', u'net.if.in[usb0]', u'net.if.in[bond0]', u'net.if.in[eth0]', u'net.if.in[eth1]', u'net.if.in[eth2]', u'net.if.in[eth3]', u'net.if.in[virbr0]', u'net.if.in[vnet0]', u'net.if.in[br103]', u'system.cpu.intr', u'kernel.maxfiles', u'kernel.maxproc', u'net.if.in[eth1,]', u'net.if.out[eth1,]', u'net.if.total[eth1,]', u'system.users.num', u'proc.num[]', u'proc.num[,,run]', u'net.if.out[vlan103]', u'net.if.out[usb0]', u'net.if.out[bond0]', u'net.if.out[eth0]', u'net.if.out[eth1]', u'net.if.out[eth2]', u'net.if.out[eth3]', u'net.if.out[virbr0]', u'net.if.out[vnet0]', u'net.if.out[br103]', u'system.cpu.load[percpu,avg1]', u'system.cpu.load[percpu,avg15]', u'system.cpu.load[percpu,avg5]', u'system.uname', u'system.uptime', u'vfs.fs.size[/,total]', u'vfs.fs.size[/boot,total]', u'vfs.fs.size[/bakdata,total]', u'vm.memory.size[total]', u'system.swap.size[,total]', u'vfs.fs.size[/,used]', u'vfs.fs.size[/boot,used]', u'vfs.fs.size[/bakdata,used]', u'agent.version']

获取当前ip 所有的监控 列表

#### 异常处理：

失败返回 信息” get monitor key failed”

返回-1

成功时返回 list

### getValueByMonitorKey

#### 功能描述:

根据监控项目获取物理服务器相应的监控值

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 监控key |

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | 当前cpu 系统占用率的百分比 |
| lastC | int | 当前时间戳 |

#### 定义:

getValueByMonitorKey(self, mkey=None)

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip = ’10.66.49.19’)

rlt = phyCpuInstance. getValueByMonitorKey(‘system.cpu.util[,system]*’*)

print rlt

{'lastvalue': 0.7852, 'lastclock': 1413274213}

lastvalue: 表示当前cpu 系统占用率的百分比， 单位%

lastclock: 表示当前时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

### getTrendsValue

#### 功能描述:

获取一段时期内的监控值

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 监控key |
| timeFrom | int | 无 | 开始时间戳 |
| timeEnd | int | 无 | 结束时间戳 |

#### 返回值：

list

例如：[{u'value\_min': 7638646784L, u'clock': 1416384000, u'value\_avg': 7640265113L, u'value\_max': 7641698304L}, {u'value\_min': 7638646784L, u'clock': 1416384000, u'value\_avg': 7640265113L, u'value\_max': 7641698304L}, {u'value\_min': 7638646784L, u'clock': 1416384000, u'value\_avg': 7640265113L, u'value\_max': 7641698304L}, {u'value\_min': 7638646784L, u'clock': 1416384000, u'value\_avg': 7640265113L, u'value\_max': 7641698304L}]

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| clock | int | 时间戳 |
| Value\_min | int | 当前时间戳最小值 |
| Value\_avg | int | 当前时间戳平均值 |
| Value\_max | int | 当前时间戳最大值 |

#### 定义:

getTrendsValue(self, mkey, timeFrom, timeEnd)

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip = ’10.66.49.19’)

rlt = phyCpuInstance. getTrendsValue(‘vm.memory.size[available]’, 1416157200, 1416388861)

print rlt

[{u'value\_min': 7638646784L, u'clock': 1416384000, u'value\_avg': 7640265113L, u'value\_max': 7641698304L}, {u'value\_min': 7638646784L, u'clock': 1416384000, u'value\_avg': 7640265113L, u'value\_max': 7641698304L}, {u'value\_min': 7638646784L, u'clock': 1416384000, u'value\_avg': 7640265113L, u'value\_max': 7641698304L}]

value\_min : 最小值

value\_avg: 平均值

value\_max: 最大值

clock: 表示时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 list

### getCpuSystemUtil

#### 功能描述:

获取物理服务器cpu系统使用率

#### 参数：

无

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | 当前cpu 系统占用率的百分比 |
| lastC | int | 当前时间戳 |

#### 定义:

getCpuSystemUtil(self)

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip = ’10.66.49.19’)

rlt = phyCpuInstance. getCpuSystemUtil()

print rlt

{'lastvalue': 0.7852, 'lastclock': 1413274213}

lastvalue: 表示当前cpu 系统占用率的百分比， 单位%

lastclock: 表示当前时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPeriodCpuSystemUtil

#### 功能描述:

获取物理服务器一段时期内cpu系统使用率

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| timeFrom | int | None | 起始时间戳 |
| timeUntil | int | None | 结束时间戳 |

#### 返回值：

list [ {‘value’: float, ‘clock’: int }, {‘value’: float, ‘clock’: int },………. , {‘value’: float, ‘clock’: int }]

#### 定义:

getPeriodCpuSystemUtil(self, timeFrom,

timeUntil)

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt =phyCpuInstance.getPeriodCpuSystemUtil(1413306000, 1413363600)

print rlt

[{u'value': 0.6945, u'clock': 1413363553}, {u'value': 0.6945, u'clock': 1413363553}, {u'value': 0.6945, u'clock': 1413363553}, {u'value': 0.6945, u'clock': 1413363553}, {u'value': 0.7448, u'clock': 1413363493}, {u'value': 0.7448, u'clock': 1413363493}, ……..

…………., {u'value': 0.2426, u'clock': 1413306013}, {u'value': 0.2426, u'clock': 1413306013}]

value: 表示当前cpu 系统占用率的百分比， 单位% clock: 表示当前时间戳.

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

如果初始化连接API 接口失败时：

失败消息如下,

Network error!

### getCpuUserUtil

#### 功能描述:

获取物理服务器cpu用户使用率

#### 参数：

无

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | cpu 用户占用率的百分比 |
| lastC | int | 当前时间戳 |

#### 定义:

getCpuUserUtil(self):

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt = phyCpuInstance. getCpuUserUtil()

print rlt

{'lastvalue': 2.6104 , 'lastclock': 1413274214}

lastvalue : 表示cpu 用户使用的百分比，单位%

lastclock：表示时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

如果初始化连接API 接口失败时：

失败消息如下,

Network error!

### getCpuIdleUtil

#### 功能描述:

获取物理服务器cpu空闲使用率

#### 参数：

无

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | 当前cpu 系统占用率的百分比 |
| lastC | int | 当前时间戳 |

#### 定义:

getCpuIdleUtil(self):

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt = phyCpuInstace. getCpuIdleUtil()

print rlt

{'lastvalue': 96.2516, 'lastclock': 1413274214}

lastvalue: 表示当前cpu 空闲百分比， 单位%

lastclock: 表示时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

如果初始化连接API 接口失败时：

失败消息如下,

Network error!

### getCpuDiskIOUtil

#### 功能描述:

获取物理服务器磁盘IO负载

#### 参数：

无

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | 当前磁盘IO使用率的百分比 |
| lastC | int | 当前时间戳 |

#### 定义:

getCpuDiskIOUtil(self):

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt = phyCpuInstace. getCpuDiskIOUtil()

print rlt

{'lastvalue': 0.1840, 'lastclock': 1413274214}

lastvalue： 磁盘io 使用百分比， 单位%

lastclock： 时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

如果初始化连接API 接口失败时：

失败消息如下,

Network error!

### getProcessLoad1min

#### 功能描述:

获取物理服务器平均1分钟负载

#### 参数：

无

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | cpu 平均1分钟负载 |
| lastC | int | 当前时间戳 |

#### 定义:

getProcessLoad1min(self):

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt = phyCpuInstance. getProcessLoad1min()

print rlt

{'lastvalue': 0.0525, 'lastclock': 1413275644}

lastvalue: 表示平均一分钟负载值

lastclock : 表示时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

如果初始化连接API 接口失败时：

失败消息如下,

Network error!

### getProcessLoad5min

#### 功能描述:

获取物理服务器平均5分钟负载

#### 参数：

无

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | cpu 平均5分钟负载 |
| lastC | int | 时间戳 |

#### 定义:

getProcessLoad5min(self):

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt = phyCpuInstace. getProcessLoad5min()

print rlt

{'lastvalue': 0.0700, 'lastclock': 1413275644}]

lastvalue: 表示平均 5分钟负载

lastclock: 时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

如果初始化连接API 接口失败时：

失败消息如下,

Network error!

### getProcessLoad15min

#### 功能描述:

获取物理服务器15分钟负载

#### 参数：

无

#### 返回值：

map

例如：{‘lastvalue’: lastV , ‘lastclock’: lastC}

|  |  |  |
| --- | --- | --- |
| 返回值描述 | 类型 | 描述 |
| lastV | float | cpu 平均15分钟负载 |
| lastC | Int | 时间戳 |

#### 定义:

getProcessLoad15min(self):

#### 样例：

phyCpuInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt = phyCpuInstace. getProcessLoad15min()

print rlt

{'lastvalue': 0.0575}, {'lastclock': 1413275644}

lastvalue: 表示cpu 平均15 分钟负载

lastclock: 时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

如果初始化连接API 接口失败时：

失败消息如下,

Network error!

### Class PhySystemStatus

PhySystemStatus 继承 PhyBase

类PhySystemStatus 包含方法getPhyTotalRootDiskSize (获取根磁盘大小),

getPhyTotalDataDiskSize (获取/bakdata 磁盘大小)，

getPhyAvailRootDiskSize (获取可用根磁盘大小),

getPhyAvailRootDiskPercent (获取可用根磁盘百分比),

getPhyAvailDataDiskPercent (获取可用/bakdata 磁盘百分比),

getPhyTotalMemory (获取总内存大小)，

getPhyAvailMemory (获取可用内存大小)，

getPhySystemUptime(获取系统运行时间)

getPhySystemLocaltime( 获取当前系统时间)

getPhySystemLoggedInUser(获取当前在线用户)

getPhySystemMaxOFile( 当前系统内核最大打开文件数)

getPhySystemMaxProcesses(当前系统内核最大进程数)

getPhyAllHosts(获取所有主机ip)

getPhyNetworkIn(带宽流入流量)

getPhyNetworkOut(带宽流出流量)

getPhyNetworkTotal(总带宽流量)

getPhyAgentPing(agentPing 主机状态)

getPhyMd5Passwd( 密码文件crc值)

getPhyCommonStatus ( 通用函数)

getMonitorExtendDefine( 未来其他监控需求)

### 初始化方法( \_\_init\_\_)

调用父类(PhyBase) 的初始化方法, 设置父类实例属性hostip (String), 调用父类方法getHostId 设置本类实例属性hid (String)

def **\_\_init\_\_**(*self*, hostip=*None*):

super(PhySystemStatus,*self*).\_\_init\_\_(hostip=hostip)

*self*.hid = *self*.getHostId(hostip)

### 类实例化

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| hostip | string | None | 主机ip地址 |

phySystemInstance = PhySystemStatus(hostip = ’10.66.49.19’)

### getPhySystemUptime

#### 功能描述:

获取物理服务器运行时间

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: timestamp, ‘lastclock’: timestamp}

#### 定义:

getPhySystemUptime(self):

#### 样例：

phySystemInstance = PhyCpuUtil(hostip='10.66.49.19’)

rlt = phySystemInstace. getPhySystemUptime(self)

print rlt

{'lastvalue': 14248337.0, 'lastclock': 1414392981}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhySystemLocaltime

#### 功能描述:

获取物理服务器当前时间

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: timestamp, ‘lastclock’: timestamp}

#### 定义:

getSystemLocaltime(self):

#### 样例：

phySystemInstance = PhyCpuUtil(hostip='10.66.49.19')

rlt = phySystemInstace. getSystemLocaltime(self)

print rlt

{'lastvalue': 1414392978.0, 'lastclock': 1414392976}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhySystemLoggedInUser

#### 功能描述:

获取物理服务器当前登录用户数

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: float, ‘lastclock’: timestamp}

#### 定义:

getSystemLoggedInUser(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhySystemLoggedInUser(self)

print rlt

{'lastvalue': 0.0, 'lastclock': 1414392982}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhySystemMaxOFile

#### 功能描述:

获取物理服务器内核最大打开文件数

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: float, ‘lastclock’: timestamp}

#### 定义:

getPhySystemMaxOFile(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhySystemMaxOFile(self)

print rlt

{'lastvalue': 1630330.0, 'lastclock': 1414391757}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhySystemMaxProcesses

#### 功能描述:

获取物理服务器内核最大进程数

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: float, ‘lastclock’: timestamp}

#### 定义:

getPhySystemMaxProcesses(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstance.getPhySystemMaxProcesses(self)

print rlt

{'lastvalue': 32768.0, 'lastclock': 1414391758}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyTotalMemory

#### 功能描述:

获取物理服务器总内存大小

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: 总内存字节数, ‘lastclock’: timestamp}

#### 定义:

getPhyTotalMemory(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstance.getPhyTotalMemory(self)

print rlt

{'lastvalue': 16816041984.0, 'lastclock': 1414391785}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyAvailableMemory

#### 功能描述:

获取物理服务器可用内存

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: 字节数, ‘lastclock’: timestamp}

#### 定义:

getPhyAvailableMemory(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstance.getPhyAvailableMemory(self)

print rlt

{'lastvalue': 5092212736.0, 'lastclock': 1414392984}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyTotalRootDiskSize

#### 功能描述:

获取物理服务器根分区磁盘大小

#### 参数：

无

#### 返回值：

Map {‘lastvalue’: 字节数, ‘lastclock’: timestamp}

#### 定义:

getPhyTotalRootDiskSize (self):

#### 样例：

phySystemInstance = PhySystemStatus (hostip='10.66.49.19')

rlt = phySystemInstance. getPhyTotalRootDiskSize (self)

print rlt

{'lastvalue': 379655434240.0, 'lastclock': 1414391695}

value: 根分区磁盘总大小， 单位是字节

clock: 当前时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyAvailableRootDiskSize

#### 功能描述:

获取物理服务器可用根分区磁盘大小

#### 参数：

无

#### 返回值：

Map

#### 定义:

getPhyAvailableRootDiskSize (self):

#### 样例：

physysinstance = PhySystemStatus(hostip='10.66.49.19')

rlt = physysinstace. getPhyAvailableRootDiskSize(self)

print rlt

{'lastvalue': 269980667904.0, 'lastclock': 1414393009}

lastvalue: 根分区可用磁盘大小， 单位是字节

lastclock: 当前时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyAvailRootDiskPercent()

#### 功能描述:

获取物理服务器根分区可用磁盘百分比

#### 参数：

无

#### 返回值：

Map

#### 定义:

getPhyAvailableRootDiskSize (self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhyAvailableRootDiskSize (self)

print rlt

{'lastvalue': 74.1482, 'lastclock': 1414393012}

lastvalue： 可用根分区磁盘百分比， 单位是百分号%

lastclock: 当前时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyNetworkTotal

#### 功能描述:

获取物理服务器网卡总流量

#### 参数：

无

#### 返回值：

Map

#### 定义:

getPhyNetworkTotal(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstance.getPhyNetworkTotal(self)

print rlt

{'lastvalue': 1049664.0, 'lastclock': 1414394995}

lastvalue： 总流量字节数

lastclock: 时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyNetworkIn

#### 功能描述:

获取物理服务器网卡流入流量

#### 参数：

无

#### 返回值：

Map

#### 定义:

getPhyNetworkIn(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhyNetworkIn(self)

print rlt

{'lastvalue': 52512.0, 'lastclock': 1414394995}

lastvalue : 流入总字节数

lastclock: 当前时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyNetworkOut

#### 功能描述:

获取物理服务器流出流量字节数

#### 参数：

无

#### 返回值：

Map

#### 定义:

getPhyNetworkOut(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhyNetworkOut(self)

print rlt

{'lastvalue': 999504.0, 'lastclock': 1414395035}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyAgentPing

#### 功能描述:

获取物理服务器存活状态

#### 参数：

无

#### 返回值：

Map

#### 定义:

getPhyAgentPing(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhyAgentPing(self)

print rlt

{'lastvalue': 1.0, 'lastclock': 1414400695}

lastvalue: 1 存活状态， 0 非存活状态

lastclock: 时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyMd5Passwd

#### 功能描述:

获取物理服务器密码文件安全性

#### 参数：

无

#### 返回值：

Map

#### 定义:

getPhyMd5Passwd(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhyMd5Passwd(self)

print rlt

{'lastvalue': 512005931.0, 'lastclock': 1414398983}

lastvalue ： 密码文件的CRC 值

lastclock: 时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getPhyCommonStatus

#### 功能描述:

根据参数key获取物理服务器相应监控信息

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 监控项目 |

#### 返回值：

Map

#### 定义:

getPhyCommonStatus(self, mkey):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getPhyCommonStatus(self, ‘vfs.file.cksum[/etc/passwd]*’*)

print rlt

{'lastvalue': 512005931.0, 'lastclock': 1414398983}

lastvalue ： 密码文件的CRC 值

lastclock: 时间戳

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

### getMonitorExtendDefine

#### 功能描述:

获取物理服务器自定义监控项目

#### 参数：

无

#### 返回值：

Map

#### 定义:

getMonitorExtendDefine(self):

#### 样例：

phySystemInstance = PhySystemStatus(hostip='10.66.49.19')

rlt = phySystemInstace. getMonitorExtendDefine(self)

print rlt

{'lastvalue': xxxxx, 'lastclock': xxxxxx}

#### 异常处理：

如果没有该监控项时返回： -1, 成功时返回 map

当初始化连接接口API失败时：

失败消息如下,

Network error!

## Package phyService

phyService package包含的类有ServiceBase, ServiceStatus

### Class ServiceStatus

ServiceStatus 继承 ServiceBase

类ServiceStatus 包含方法如下:

|  |  |
| --- | --- |
| 方法名 | 描述 |
| getMonitorService | 获取当前主机的所有监控项目 |
| getServiceStatusByKey | 通过服务名获取相应服务状态 |
| getCommonServiceStatus | 通用服务状态(通过制定的监控项, 获取状态) |
| getWebStatus | 自定义网站运行状态 |
| getMongodbStatus | Mongodb 运行状态 |
| getPingStatus | 主机运行状态 |
| getNovaNetworkStatus | Nova-network 服务运行状态 |
| getMemcacheStatus | Memcache 运行状态 |
| getSSHStatus | Ssh 服务运行状态 |
| getCurrentUserStatus | 当前用户数 |
| getSmtpStatus | Smtp 服务运行状态 |
| getPop3Status | Pop3 服务运行状态 |
| getCurrentLoadStatus | 当前负载情况 |
| getRootPartionStatus | 当前根磁盘使用情况 |
| getHttpStatus | 当前主机web 服务运行情况 |
| getUserDefinedStatus | 用户自定义服务扩展 |

### 初始化方法( \_\_init\_\_)

调用父类(ServiceBase) 的初始化方法, 设置父类实例属性hostip (String), 调用父类方法getHostId 设置本类实例属性hid (String)

def **\_\_init\_\_**(*self*, hostip=*None*):

super(ServiceStatus,*self*).\_\_init\_\_(hostip=hostip)

*self*.hid = *self*.getHostId(hostip)

### 类实例化

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| Hostip | string | None | 主机ip地址 |

SI = ServiceStatus(hostip = ’10.66.49.21’)

### getMonitorService

#### 功能描述:

获取物理服务器当前主机的所有监控项目

#### 参数：

无

#### 返回值：

list

#### 定义:

getMonitorService(self):

#### 样例：

SI= ServiceStatus(hostip='10.66.49.21')

rlt =SI.getMonitorService(self)

print rlt

[u'HTTP', u'Current\_load', u'PING', u'Swap Usage', u'Current Load', u'Current Users', u'Root Partition', u'SSH', u'Total Processes', u'incitoWeb']

成功时返回当前主机所有的监控项目， 失败时返回 -1

连接失败时消息如下：

connect server request error

返回 -1

### getServiceStatusByKey

#### 功能描述:

根据服务名字获取服务状态

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 需要监控的服务名字 |

#### 返回值：

map

#### 定义:

getServiceStatusByKey(self, mkey=’LY\_novaConductor’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.18')

rlt = SI.getServiceStatusByKey(‘LY\_novaConductor’)

print rlt

{'current\_state': u'OK', 'plugin\_output': u'LY nova-conductor run OK'}

current\_state: OK， 服务运行正常, Critical 表示服务不正常

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getWebStatus

#### 功能描述:

获取物理服务器指定web 可用性

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | incitoWeb | [www.incito.com.cn](http://www.incito.com.cn) 可用性 |

#### 返回值：

map

#### 定义:

getWebStatus(self, mkey=’incitoWeb’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getWebStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': 'web OK \xe4\xbb\x8e\xe4\xb8\xb4\xe6\xb2\x82\xe6\xb5\x8b\xe8\xaf\x95\xe6\x9c\xba\xe7\x9b\x91\xe6\x8e\xa7incitoweb\xe6\xad\xa3\xe5\xb8\xb8 - <http://www.incito.com.cn>'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getMongodbStatus

#### 功能描述:

获取物理服务器mongodb 服务状态

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | mongodb | Mongodb 服务可用性 |

#### 返回值：

map

#### 定义:

getMongodbStatus(self, mkey=’mongodb’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.18')

rlt = SI.getMongodbStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'LY mongodb run OK'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getNovaNetworkStatus

#### 功能描述:

获取物理服务器NovaNetwork 服务状态

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | novaNetwork | novaNetwork 服务可用性 |

#### 返回值：

map

#### 定义:

getNovaNetwork(self, mkey=’novaNetwork’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.18')

rlt = SI.getNovaNetworkStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'LY nova-network run OK'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getMemcacheStatus

#### 功能描述:

获取物理服务器memcache 服务状态

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | memcached | Memcached 服务 可用性 |

#### 返回值：

map

#### 定义:

getMemcacheStatus(self, mkey=’memcached’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getMemcacheStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'LY memcached run OK'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getCommonServiceStatus

#### 功能描述:

获取物理服务器指定服务状态

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 指定服务 可用性 |

#### 返回值：

map

#### 定义:

getCommonServiceStatus(self, mkey=None):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getCommonServiceStatus(“PING”)

print rlt

{'current\_state': u'0', 'plugin\_output': u'PING OK - Packet loss = 0%, RTA = 0.46 ms'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getPingStatus

#### 功能描述:

获取物理服务器网络连通性ping

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | PING | 网络可用性 |

#### 返回值：

map

#### 定义:

getPingStatus(self, mkey=’PING’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getPingStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'PING OK - Packet loss = 0%, RTA = 0.03 ms'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getSSHStatus

#### 功能描述:

获取物理服务器ssh 可用性

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | SSH | 22 端口可用性 |

#### 返回值：

map

#### 定义:

getSSHStatus(self, mkey=’SSH’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getSSHStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'SSH OK - OpenSSH\_6.4 (protocol 2.0)'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getHttpStatus

#### 功能描述:

获取物理服务器cpu系统使用率

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | HTTP | 被监控机本身web可用性 |

#### 返回值：

map

#### 定义:

getHttpStatus(self, mkey=’HTTP’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getHttpStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'HTTP OK: HTTP/1.1 200 OK - 1273 bytes in 0.001 second response time'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getSmtpStatus

#### 功能描述:

获取物理服务器smtp 服务状态

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | smtp | Smtp可用性 |

#### 返回值：

map

#### 定义:

getSmtpStatus(self, mkey=’smtp’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getSmtpStatus()

print rlt

{'current\_state': string, 'plugin\_output': string}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getPop3Status

#### 功能描述:

获取物理服务器pop3 服务状态

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | pop3 | Pop3可用性 |

#### 返回值：

map

#### 定义:

getPop3Status(self, mkey=’pop3’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getPop3Status()

print rlt

{'current\_state': string, 'plugin\_output': string}

current\_state: 0 (字符串类型) 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getCurrentLoadStatus

#### 功能描述:

获取物理服务器当前负载情况

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | Current\_load | Cpu 负载情况 |

#### 返回值：

map

#### 定义:

getCurrentLoadStatus(self, mkey=’Current\_load’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getCurrentLoadStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'OK - load average: 0.00, 0.01, 0.05'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getRootPartitionStatus

#### 功能描述:

获取物理服务器根分区使用情况

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | Root Partition | 根分区使用情况 |

#### 返回值：

map

#### 定义:

getRootPartitionStatus(self, mkey=’Root Partition’):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getRootPartitionStatus()

print rlt

{'current\_state': u'0', 'plugin\_output': u'DISK OK - free space: / 5886 MB (63% inode=82%):'}

current\_state: 0 表示OK， 服务运行正常.

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

### getUserDefinedStatus

#### 功能描述:

获取用户自定义服务状态的监控

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | Xxxx 服务可用性 |

#### 返回值：

map

#### 定义:

getUserDefinedStatus(self, mkey=None):

#### 样例：

SI = ServiceStatus(hostip='10.66.49.21')

rlt = SI.getUserDefinedStatus()

print rlt

{'current\_state': string, 'plugin\_output': string }

current\_state: 0 表示OK， 服务运行正常., 1 表示服务运转异常

plugin\_output: 服务具体的描述信息.

出错时消息:

no such monitor service:

返回： -1

## Package virMachine

virMachine package包含的类有VirBase, VirNovaUtil, VirGlanceUtil

### Class VirNovaUtil

VirNovaUtil 继承 VirBase

类VirNovaUtil 包含方法

|  |  |  |
| --- | --- | --- |
| 方法名 | 描述 | 值类型 |
| getAllMeters | 获取所有的监控项目 | list |
| getMeterNameByResId | 根据资源id获取监控项 | list |
| getTrendsData | 获取历史数据 | list |
| getCpuUsedTime | Cpu 已经使用时间 | 累计值 |
| getDiskReadBytesRate | 平均每秒每虚拟机磁盘读速率 | 瞬时值 |
| getCpuUtil | Cpu 使用率 | 瞬时值 |
| getDiskReadBytes | 磁盘读字节数 | 累计值 |
| getDiskWriteBytes | 磁盘写字节数 | 累计值 |
| getNetworkInBytes | 网卡流入字节数 | 累计值 |
| getNetworkOutBytes | 网卡流出字节数 | 累计值 |

.

### 初始化方法( \_\_init\_\_)

调用父类(VirBase) 的初始化方法, 设置父类实例属性resource\_id (String)

def **\_\_init\_\_**(*self*, resource\_id=*None*):

super(VirNovaUtil,*self*).\_\_init\_\_(resource\_id=resource\_id)

### 类实例化

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| Resource\_id | string | None | 监控资源id |

vn = VirNovaUtil(resource\_id = ’*572e2069-1d61-4e0a-9a7d-fc9e726c22cb*’)

### getAllMeters

#### 功能描述:

获取虚拟服务器资源id 和监控项目的列表

#### 参数：

无

#### 返回值：

list

#### 定义:

getAllMeters(self):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getAllMeters()

print rlt

[{'name': u'image', 'resource\_id': u'f57e4f9f-6524-48d7-bd7b-8677085344d2'}, {'name': u'image.size', 'resource\_id': u'f57e4f9f-6524-48d7-bd7b-8677085344d2'}, {'name': u'image.download', 'resource\_id': u'f57e4f9f-6524-48d7-bd7b-8677085344d2'}, {'name': u'image.serve', 'resource\_id': u'f57e4f9f-6524-48d7-bd7b-8677085344d2'}, {'name': u'disk.write.bytes', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'cpu', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'disk.read.bytes', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'disk.write.requests.rate', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'disk.read.requests.rate', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'disk.write.bytes.rate', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'cpu\_util', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'disk.read.bytes.rate', 'resource\_id': u'4480b6a9-9c35-4edc-ae64-4380a76c6902'}, {'name': u'network.incoming.packets', 'resource\_id': u'nova-instance-instance-00000001-fa163ede05c5'},{'name': u'network.outgoing.packets', 'resource\_id': u'nova-instance-instance-00000001-fa163ede05c5'}]

name: 监控项目

resource\_id : 资源id

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getMeterNameByResId

#### 功能描述:

根据资源id 获取监控项目值

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| resid | string | None | 资源id |

#### 返回值：

list

#### 定义:

getMeterNameByResId(self, resid=None):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getMeterNameByResId(‘*572e2069-1d61-4e0a-9a7d-fc9e726c22cb’*)

print rlt

[u'instance:core1', u'cpu', u'disk.write.bytes', u'disk.read.requests', u'instance', u'disk.write.requests', u'disk.read.bytes', u'cpu\_util', u'disk.write.bytes.rate', u'disk.read.requests.rate', u'disk.write.requests.rate', u'disk.read.bytes.rate', u'vcpus', u'memory', u'disk.ephemeral.size', u'disk.root.size']

根据资源id 获取监控项目

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getTrendsData

#### 功能描述:

获取虚拟服务器指定监控项目一段时期内的值

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 监控项目 |
| days | int | None | 天数 |
| timeFrom | int | None | 时间戳 |
| timeEnd | int | None | 时间戳 |

#### 返回值：

list

#### 定义:

getTrendsData(self, mkey=None, timeFrom=None, timeEnd=None, days=1):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getTrendsData(mkey=’disk.write.bytes.rate’,timeFrom=1416326400, timeEnd=1416467700, days=1)

print rlt

[{'period\_end': u'2014-11-19T16:00:00', 'min': 894.2933333333333, 'duration': 85812.0, 'max': 106905.6, 'sum': 260440.8610610586, 'avg': 1808.6170907017959, 'period\_start': u'2014-11-18T16:00:00', 'unit': u'B/s'}, {'period\_end': u'2014-11-20T16:00:00', 'min': 914.7733333333333, 'duration': 54609.0, 'max': 117937.49333333333, 'sum': 216259.92874098726, 'avg': 2350.6513993585572, 'period\_start': u'2014-11-19T16:00:00', 'unit': u'B/s'}]

获取1天内的磁盘写速率.

period\_end: utc 时间字符串，

min: 最小值

duratiton：采集间隔

max: 最大值

sum: 求和

avg: 均值

period\_start: 开始时间字符串

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getDaysSample

#### 功能描述:

获取虚拟服务器指定监控项目一段时期内的值

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 监控项目 |
| days | int | None | 间隔天数 |

#### 返回值：

list

#### 定义:

getDaysSample(self, mkey=None, days=1):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getDaysSample(‘disk.read.requests.rate’, 1)

print rlt

[{'period\_end': u'2014-10-27T13:57:56', 'min': 0.0, 'duration': 21003.0, 'max': 0.0, 'sum': 0.0, 'avg': 0.0, 'period\_start': u'2014-10-27T07:57:56', 'unit': u'request/s'}, {'period\_end': u'2014-10-27T19:57:56', 'min': 0.0, 'duration': 21002.0, 'max': 0.0, 'sum': 0.0, 'avg': 0.0, 'period\_start': u'2014-10-27T13:57:56', 'unit': u'request/s'}, {'period\_end': u'2014-10-28T01:57:56', 'min': 0.0, 'duration': 21003.0, 'max': 0.0, 'sum': 0.0, 'avg': 0.0, 'period\_start': u'2014-10-27T19:57:56', 'unit': u'request/s'}, {'period\_end': u'2014-10-28T07:57:56', 'min': 0.0, 'duration': 21003.0, 'max': 0.0, 'sum': 0.0, 'avg': 0.0, 'period\_start': u'2014-10-28T01:57:56', 'unit': u'request/s'}]

获取1天内的磁盘读请求速率.

period\_end: utc 时间字符串，

min: 最小值

duratiton：采集间隔

max: 最大值

sum: 求和

avg: 均值

period\_start: 开始时间字符串

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getCpuUsedTime

#### 功能描述:

获取虚拟服务器cpu已使用时间

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | cpu | Cpu 已用时间 |

#### 返回值：

list

#### 定义:

getCpuUsedTime(self, mkey=’cpu’):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getCpuUsedTime()

print rlt

[{'timestamp': u'2014-10-28T05:46:22', 'counter\_name': u'cpu', 'counter\_unit': u'ns', 'counter\_volume': 3259260000000.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getDiskReadBytesRate

#### 功能描述:

获取虚拟服务器磁盘平均每秒读速率

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | disk.read.bytes.rate | 磁盘每秒读速率 |

#### 返回值：

list

#### 定义:

getDiskReadBytesRate(self, mkey=’disk.read.bytes.rate’):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getDiskReadBytesRate()

print rlt

[{'timestamp': u'2014-10-28T07:56:23', 'counter\_name': u'disk.read.bytes.rate', 'counter\_unit': u'B/s', 'counter\_volume': 0.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getCpuUtil

#### 功能描述:

获取虚拟服务器cpu使用率

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | cpu\_util | Cpu 使用率 |

#### 返回值：

list

#### 定义:

getCpuUtil(self, mkey=’cpu\_util’):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getCpuUtil()

print rlt

[{'timestamp': u'2014-10-28T05:46:22', 'counter\_name': u'cpu\_util', 'counter\_unit': u'%', 'counter\_volume': 0.08833333333333333}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getDiskReadBytes

#### 功能描述:

获取虚拟服务器磁盘读请求字节数

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | disk.read.bytes | 累计磁盘读字节数 |

#### 返回值：

list

#### 定义:

getDiskReadBytes(self, mkey=’disk.read.bytes’):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getDiskReadBytes()

print rlt

[{'timestamp': u'2014-10-28T05:46:22', 'counter\_name': u'disk.read.bytes', 'counter\_unit': u'B', 'counter\_volume': 265694720.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getDiskWriteBytes

#### 功能描述:

获取虚拟服务器磁盘写请求字节数

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | disk.write.bytes | 累计磁盘写请求字节数 |

#### 返回值：

list

#### 定义:

getDiskWriteBytes(self, mkey=’disk.write.bytes’):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getDiskWriteBytes()

print rlt

[{'timestamp': u'2014-10-28T05:46:22', 'counter\_name': u'disk.read.bytes', 'counter\_unit': u'B', 'counter\_volume': 265694720.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getNetworkInBytes

#### 功能描述:

获取虚拟服务器网络流入流量字节数

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | network.incoming.bytes | 累计流入流量字节数 |

#### 返回值：

list

#### 定义:

getNetworkInBytes(self, mkey=’network.incoming.bytes’):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getNetworkInBytes()

print rlt

[{'timestamp': u'2014-10-28T05:46:22', 'counter\_name': u'network.incoming.bytes', 'counter\_unit': u'B', 'counter\_volume': 265694720.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getNetworkOutBytes

#### 功能描述:

获取虚拟服务器流出流量字节数

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | network.outgoing.bytes | 累计流出流量字节数 |

#### 返回值：

list

#### 定义:

getNetworkOutBytes(self, mkey=’network.outgoing,bytes’):

#### 样例：

vn = VirNovaUtil(resource\_id='*572e2069-1d61-4e0a-9a7d-fc9e726c22cb* ')

rlt = vn.getNetworkOutBytes()

print rlt

[{'timestamp': u'2014-10-28T05:46:22', 'counter\_name': u'network.outgoing.bytes', 'counter\_unit': u'B', 'counter\_volume': 265694720.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### Class VirGlanceUtil

VirGlanceUtil 继承 VirBase

类VirGlanceUtil 包含方法

|  |  |  |
| --- | --- | --- |
| 方法名 | 描述 | 值类型 |
| getGlanceImageExists | Image 存在与否 | 累计值 |
| getGlanceImageSize | 已经更新image 大小 | 累计值 |
| getGlanceImageUpdate | Image 更新次数 | 累计值 |
| getGlanceImageDelete | Image 删除次数 | 累计值 |
| getGlanceImageDownload | Image 已经下载字节 | 累计值 |
| getGlanceImageServ | Image 已经使用字节 | 累计值 |

### 初始化方法( \_\_init\_\_)

调用父类(VirBase) 的初始化方法, 设置父类实例属性resource\_id (String),

def **\_\_init\_\_**(*self*, resource\_id=*None*):

super(VirGlanceUtil,*self*).\_\_init\_\_(resource\_id=resource\_id)

### 类实例化

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| resource\_id | string | None | 监控资源Id |

vg = VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

### getCommonService

#### 功能描述:

获取相应监控项目的值

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | None | 监控项目 |

#### 返回值：

list

#### 定义:

getCommonService(self， mkey=None):

#### 样例：

vg= VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

rlt = vg.getCommonService(self, ‘image’)

print rlt

[{'timestamp': u'2014-10-28T06:37:16', 'counter\_name': u'image', 'counter\_unit': u'image', 'counter\_volume': 1.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getGlanceImageExists

#### 功能描述:

获取虚拟服务器镜像存在与否

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | image | 监控image |

#### 返回值：

list

#### 定义:

getGlanceImageExists(self, mkey=’image’):

#### 样例：

vg= VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

rlt = vg.getGlanceImageExists()

print rlt

[{'timestamp': u'2014-10-28T06:37:16', 'counter\_name': u'image', 'counter\_unit': u'image', 'counter\_volume': 1.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getGlanceImageSize

#### 功能描述:

获取虚拟服务器已经更新镜像大小

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | image.size | 已经更新镜像大小 |

#### 返回值：

list

#### 定义:

getGlanceImageSize(self, mkey=’image.size’):

#### 样例：

vg= VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

rlt = vg. getGlanceImageSize(self)

print rlt

[{'timestamp': u'2014-10-28T06:37:16', 'counter\_name': u'image.size', 'counter\_unit': u'B', 'counter\_volume': 13167616.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getGlanceImageUpdate

#### 功能描述:

获取虚拟服务器image 已经更新次数

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | image.update | Image update |

#### 返回值：

list

#### 定义:

getGlanceImageUpdate(self, mkey=’image.update’):

#### 样例：

vg= VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

rlt = vg. getGlanceImageUpdate(self)

print rlt

[{'timestamp': u'2014-10-28T06:37:16', 'counter\_name': u'image.update', 'counter\_unit': u'image', 'counter\_volume': 1.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getGlanceImageDelete

#### 功能描述:

获取虚拟服务武器image 已经删除次数

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | image.delete | Image delete |

#### 返回值：

list

#### 定义:

getGlanceImageDelete(self, mkey=’image.delete’):

#### 样例：

vg= VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

rlt = vg. getGlanceImageDelete(self)

print rlt

[{'timestamp': u'2014-10-28T06:37:16', 'counter\_name': u'image.delete', 'counter\_unit': u'image', 'counter\_volume': 1.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getGlanceImageDownload

#### 功能描述:

获取虚拟服务器镜像已下载大小

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | image.download | Image downloadsize |

#### 返回值：

list

#### 定义:

getGlanceImageDownload(self, mkey=’image.download’):

#### 样例：

vg= VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

rlt = vg. getGlanceImageDownload(self)

print rlt

[{'timestamp': u'2014-09-04T09:25:02.296000', 'counter\_name': u'image.download', 'counter\_unit': u'B', 'counter\_volume': 13167616.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1

### getGlanceImageServe

#### 功能描述:

获取虚拟服务器镜像已经使用字节

#### 参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 默认值 | 描述 |
| mkey | string | image.serve | Image 已经使用大小 |

#### 返回值：

list

#### 定义:

getGlanceImageServe(self, mkey=’image.serve’):

#### 样例：

vg= VirGlanceUtil(resource\_id=’f57e4f9f-6524-48d7-bd7b-8677085344d2’)

rlt = vg. getGlanceImageServe(self)

print rlt

[{'timestamp': u'2014-09-04T09:25:02.296000', 'counter\_name': u'image.serve', 'counter\_unit': u'B', 'counter\_volume': 13167616.0}]

timestamp: utc 时间字符串，

counter\_name: 监控项目

counter\_unit: 单位

counter\_volume: 监控值

成功时返回: list

失败时返回: -1

初始化连接失败：

request url error!

返回值: -1