# VmwareAPI(扼要)

## interface.general(查询操作)

### 类---AllHostNameAndMOR 类功能说明：获取指定主机下的全部虚拟机和模板及其状态

**a.函数扼要：**

#### [AllHostNameAndMOR\* getAllHostNameAndMORInstance()](#_1.AllHostNameAndMOR*_getAllHostName) ---获取该类的单例对象

#### [void getAllHostNameString()](#_2.void_getAllHostNameString()) ---获取全部主机的名称

#### [void getAllHostNameStringSignal(QString)](#_3._void_getAllHostNameStringSignal() ---接收全部主机的名称的数据的信号

### 类---DatastoresByHost 类功能说明：获取指定主机所可以使用的全部存储器

**a.函数扼要：**

#### [static DatastoresByHost\* getDatastoresByHostInstance()](#_1.static__DatastoresByHost*) ---获取单例对象

#### [void getDataStores(QString hostName)](#_2._void_) ---获取指定主机下的全部的存储器

#### [void getDataStoresSignal(QString)](#_3._void_) ---接收指定主机下的全部的存储器的信号

#### [void getDatastoreSubForder(QString hostName)](#_4.void_getDatastoreSubForder(QStrin) ---取得存储器的一级文件夹

#### [void getDatastoreSubForderSignal(QString)](#_5.void_getDatastoreSubForderSignal() ---接收数据

### 类---HostInfo 类功能说明：获取指定主机下的全部虚拟机和模板及其状态

a.函数扼要：

#### [static HostInfo\* getHostInfoInstance()](#_1.static__HostInfo*) ---获取单例对象

#### [void getHostInfoString(QString hostName)](#_2.void__getHostInfoString(QString) ---获获取指定主机的详情信息

#### [void getHostInfoStringSignal(QString)](#_3._void__1) ---接收机的详情信息数据的信号

### 类---IsoFile 类功能说明：用于获取创建虚拟机时所需要的镜像文件

a.函数扼要：

#### [static IsoFile\* getIsoFileInstance();](#_1.static__IsoFile*) ---获取单例对象

#### [void getIsoFile();](#_2.void__getIsoFile();) ---获取全部镜像文件

#### [void getIsoFileSignal(QString);](#_3.void__getIsoFileSignal(QString);) ---接收用于获取创建虚拟机时所需要的镜像文件数据的信号

### 类---VirtualMachineInfo 类功能说明：获取虚拟机详情信息

a.函数扼要：

#### [static VirtualMachineInfo\* getVirtualMachineInfoInstance();](#_1._static_) ---获取单例对象

#### [void getVirtualMachineInfoString(QString hostName,QString vmName);](#_2._void__1) ---获取指定主机下的虚拟机的详情信息

#### [void getVirtualMachineInfoStringSignal(QString);](#_3._void__2) ---接收定主机下的虚拟机的详情信息数据的信号

### 类--- VMNameAndState 类功能说明：获取指定主机下的全部虚拟机和模板及其状态

a.函数扼要：

#### [static VMNameAndState\* getVMNameAndStateInstance();](#_1.static__VMNameAndState*) ---获取单例对象

#### [void getVMNameAndStateString(QString hostName)](#_2.void__getVMNameAndStateString(QSt) ---获取制定主机下的的全部虚拟机的名称和状态

#### [void getVMNameAndStateStringSignal(QString)](#_3.void_getVMNameAndStateStringSigna) ---接收获取制定主机下的的全部虚拟机的名称和状态数据的信号

### 类---VNCInfo 类功能说明:获取VNC信息

#### [void getVNCInfo(QString hostName,QString vmName)](#_1.void_getVNCInfo(QString_hostName,)---获取指定虚拟机的VNC端口，密码

#### [void getVNCInfoSignals(QString)](#_2.void_getVNCInfoSignals(QString)) ---接收数据

## interface.host(主机操作)

### 类--- HostOperation 类功能说明：完成主机的基本操作（添加，移除，进入维护模式，推出维护模式，断开，重新连接）

a.函数扼要：

#### [static HostOperation\* getHostOperationInstance();](#_1.static__HostOperation*) ---获取单例对象

#### [void addHostToCluster(QString clusterName,QString hostName,QString userName,QString password) ---添加主机到集群](#_2.void__addHostToCluster(QString)

#### [void removeHostFromCluster(QString hostName);](#_3.void__removeHostFromCluster(QStri) ---将主机从集群中移除

#### [void enterMaintenanceMode(QString hostName);](#_4.void__enterMaintenanceMode(QStrin) ---将主机进入维护模式

#### [void exitMaintenanceMode(QString hostName);](#_5.void__exitMaintenanceMode(QString) ---将主机退出维护模式

#### [void disconnectHost(QString hostName);](#_6.void__disconnectHost(QString) ---断开与主机的连接

#### [void reconnectHost(QString hostName);](#_7.void__reconnectHost(QString) ---重新与主机进行连接

#### [void addHostToClusterSignal(QString);](#_8.void_addHostToClusterSignal(QStri) ---添加主机到集群的信号

#### [void removeHostFromClusterSignal(QString);](#_9.void_removeHostFromClusterSignal() ---将主机从集群中移除的信号

#### [void enterMaintenanceModeSignal(QString);](#_10.void_enterMaintenanceModeSignal() ---将主机进入维护模式的信号

#### [void exitMaintenanceModeSignal(QString);](#_11.void_exitMaintenanceModeSignal(Q) ---将主机退出维护模式的信号

#### [void disconnectHostSignal(QString);](#_12.void_disconnectHostSignal(QStrin) ---断开与主机的连接的信号

#### [void reconnectHostSignal(QString);](#_13.void_reconnectHostSignal(QString) ---重新与主机进行连接的信号

### 类--- SiteFirewallDefaultPolicy 类功能说明: 获取指定指定主机的默认防火墙规则

a.函数摘要

#### [void enableFirewallDefaultPolicy(QString hostName)](#_1.String_enableFirewallDefaultPolic)---设置防火前的默认策略true(deny)

#### [void undoFirewallDefaultPolicy(QString hostName)](#_2.String_undoFirewallDefaultPolicy()---设置防火前的默认策略false(permit)

#### void enableFirewallDefaultPolicySignals(QString)---接收数据

#### void undoFirewallDefaultPolicySignals(QString)---接收数据

## interface.vm(虚拟机操作)

### 类--- VMClone 类功能说明: 完成虚拟的克隆,从模板部署虚拟机

1. 函数扼要:

#### [static VMClone\* getVMCloneInstance();](#_1.static__VMClone*) ---获取单例对象

#### [void cloneVM(QString hostName, QString vmName, QString cloneName);](#_2.void_cloneVM(QString_hostName,) ---克隆虚拟机(开机或关机)

#### [void cloneTemplate(QString hostName,QString templateName, QString cloneName);](#_3._void_cloneTemplate(QString) --- 克隆模板 (模板是虚拟机的主镜像)

#### [void cloneVMToTemplate(QString hostName,QString vmName, QString cloneName) ;](#_4.void_cloneVMToTemplate(QString_ho) --- 虚拟机克隆为模板

#### [void cloneTemplatToVM(QString clusterName, QString hostName,QString templateName, QString cloneName) ;](#_5._void_cloneTemplatToVM(QString) --- 从模板部署虚拟机

#### [void vmToTemp( QString hostName,QString vmName);](#_6.void_vmToTemp(_QString) --- 将虚拟机转化为模板

#### [void markAsVM(QString clusterName,QString hostName, QString templateName);](#_7.void_markAsVM(QString_clusterName) --- 将模板转化为虚拟机

#### void cloneVMSignal(QString); ---接收数据

#### void cloneTemplateSignal(QString); ---接收数据

#### void cloneVMToTemplateSignal(QString); ---接收数据

#### void cloneTemplatToVMSignal(QString); ---接收数据

#### void vmToTempSignal(QString); ---接收数据

#### void markAsVMSignal(QString); ---接收数据

### 类--- VMCreate 类功能说明: 完成虚拟的创建

1. 函数扼要:

#### [static VMCreate\* getVMCreateInstance();](#_1.static__VMCreate*) ---获取单例对象

#### [void createVirtualMachine(QString hostname,QString vmName, QString guestOsId,](#_2.void__createVirtualMachine(QStrin)

[QString numCpus,QString diskSize,QString vmMemorySize,](#_2.void__createVirtualMachine(QStrin)

[QString numNics,QString dataStore,QString iso,](#_2.void__createVirtualMachine(QStrin)

[QString startVM);](#_2.void__createVirtualMachine(QStrin)  ---创建虚拟机

#### void createVirtualMachineSignal(QString); ---接收数据

### 类---VMOtherOps 类功能说明: 完成虚拟机的一些额外的操作

a.函数扼要:

#### [static VMOtherOps\* getVMOtherOpsInstance();](#_1.static__VMOtherOps*) ---获取单例对象

#### [void destroyVm(QString hostName,QString vmName);](#_2.void__destroyVm(QString)---删除虚拟机

#### [void unregisterVm(QString hostName,QString vmName);](#_3.void__unregisterVm(QString)---从清单中移除虚拟机

#### [void renameVm(QString hostName,QString vmName, QString newVmName);](#_4.void__renameVm(QString)---虚拟机重命名

#### void destroyVmSignal(QString); ---接收数据

#### void unregisterVmSignal(QString); ---接收数据

#### void renameVmSignal(QString); ---接收数据

### 类--- VMotion 类功能说明: 完成虚拟的迁移和重定向

a.函数扼要:

#### [static VMotion\* getVMotionInstance();](#_1.static__VMotion*) ---获取单例对象

#### [void migrateVM(QString vmname, QString tcluster,QString tHost,](#_2._void__2)

[QString srcHost, QString priority); -](#_2._void__2)-- 对虚拟机进行迁移操作

#### [relocateVMOnSameHost(QString vmname,QString tcluster,QString tDS, QString srcHost,QString priority);-](#_3._void__3)-- 对虚拟机在同一主机内进行重定位操作

#### [relocateVMOnDifferentHost(QString vmname,QString tcluster,QString tHost,QString tDS,QString srcHost,QString priority);](#_4.void__relocateVMOnDifferentHost(Q)--- 对虚拟机在不同主机内进行重定位操作

#### void migrateVMSignal(QString); ---接收数据

#### void relocateVMOnSameHostSignal(QString); ---接收数据

#### void relocateVMOnDifferentHostSignal(QString); ---接收数据

### 类--- VMPowerOps 类功能说明: 完成虚拟的基本操作

a.函数扼要:

#### [static VMPowerOps\* getVMPowerOpsInstance();](#_1.static__VMPowerOps*) ---获取单例对象

#### [void powerOnVM(QString hostName,QString vmName);-](#_2.void__powerOnVM(QString)-- 给指定的虚拟机打开电源

#### [void powerOffVM(QString hostName, QString vmName);-](#_3.void__powerOffVM(QString)-- 给指定的虚拟机关闭电源

#### [void resetVM(QString hostName,QString vmName);](#_4.void__resetVM(QString) --- 复位指定的虚拟机

#### [void suspendVM(QString hostName,QString vmName);](#_5.void__suspendVM(QString)--- 挂起指定的虚拟机

#### [void rebootVM(QString hostName,QString vmName);](#_6.void__rebootVM(QString)--- 重启客户机

#### [void shutdownVM(QString hostName,QString vmName);-](#_7.void__shutdownVM(QString)-- 关闭客户机

#### [void standbyVM(QString hostName,QString vmName);](#_8.void__standbyVM(QString)--- 启动客户机操作系统待机

#### void powerOnVMSignal(QString); ---接收数据

#### void powerOffVMSignal(QString); ---接收数据

#### void resetVMSignal(QString); ---接收数据

#### void suspendVMSignal(QString); ---接收数据

#### void rebootVMSignal(QString); ---接收数据

#### void shutdownVMSignal(QString); ---接收数据

#### void standbyVMSignal(QString); ---接收数据

### 类--- VMSnapshot 类功能说明: 完成虚拟的查看快照、创建快照、删除指定快照、删除全部快照、恢复指定快照、恢复到当前快照

a.函数扼要:

#### [static VMSnapshot\* getVMSnapshotInstance();](#_1.static__VMSnapshot*) ---获取单例对象

#### [void createSnapshot(QString hostName,QString vmName,](#_2.void__createSnapshot(QString)

[QString snapshotName,QString desc)](#_2.void__createSnapshot(QString)---创建虚拟机快照

#### [void listSnapshot(QString hostName, QString vmName)-](#_3._void__4)--列举指定虚拟机的当期的全部快照

#### [void revertSnapshot(QString hostName,QString vmName,](#_4.void__revertSnapshot(QString)

[QString snapshotName)](#_4.void__revertSnapshot(QString) ---恢复到指定的快照

#### [void revertToCurrentSnapshot(QString hostName,QString vmName);](#_5.void__revertToCurrentSnapshot(QSt) ---恢复到当前快照

#### [void removeAllSnapshot(QString hostName,QString vmName);](#_6.void__removeAllSnapshot(QString) ---删除全部快照

#### [void removeSnapshot(QString hostName,QString vmName,QString snapshotName,](#_7.void_removeSnapshot(QString_hostN)

[QString removeChild);](#_7.void_removeSnapshot(QString_hostN) ---移除指定的快照

#### void createSnapshotSignal(QString); ---接收数据

#### void listSnapshotSignal(QString); ---接收数据

#### void revertSnapshotSignal(QString); ---接收数据

#### void revertToCurrentSnapshotSignal(QString); ---接收数据

#### void removeAllSnapshotSignal(QString); ---接收数据

#### void removeSnapshotSignal(QString); ---接收数据

# VmwareAPI(详细)

## interface.general(查询操作)

### 类---AllHostNameAndMOR 类功能说明：获取指定主机下的全部虚拟机和模板及其状态

函数详细说明：

#### [1.AllHostNameAndMOR\* getAllHostNameAndMORInstance()](#_getAllHostNameAndMORInstance()__)

功能：获取该类的单例对象

返回值：AllHostNameAndMOR对象

参数：

#### [2.void getAllHostNameString()](#_void__)

功能：获取全部主机的名称

返回值：空

参数：

#### [3. void getAllHostNameStringSignal(QString)](#_void___1)

功能：接收数据

返回值：空

参数：

### 类---DatastoresByHost 类功能说明：获取指定主机所可以使用的全部存储器

函数详细说明：

#### [1.static DatastoresByHost\* getDatastoresByHostInstance()](#_static_DatastoresByHost*_)

功能：获取单例对象

返回值：DatastoresByHost 指针

参数：

#### [2. void getDataStores(QString hostName)](#_void___2)

功能：获取指定主机下的全部的存储器

返回值：空

参数：hostName——主机名

#### [3. void getDataStoresSignal(QString);](#_void__getDataStoresSignal(QString))

功能：接收数据

返回值：空

参数：

#### [4.void getDatastoreSubForder(QString hostName)](#_void_getDatastoreSubForder(QString_)

功能：取得存储器的一级文件夹

返回值：空

参数： hostName ---主机名

#### [5.void getDatastoreSubForderSignal(QString)](#_void_getDatastoreSubForderSignal(QS)

功能：接收数据

返回值：空

参数：

### 类---HostInfo 类功能说明：获取指定主机下的全部虚拟机和模板及其状态

函数详细说明：

#### [1.static HostInfo\* getHostInfoInstance();](#_static__HostInfo*)

功能：获取单例对象

返回值：HostInfo指针

参数：

#### [2.void getHostInfoString(QString hostName);](#_void___4)

功能：获获取指定主机的详情信息

返回值：空

参数：hostName——主机名

#### [3. void getHostInfoStringSignal(QString);](#_void___3)

功能：接收数据

返回值：空

参数：

### 类---IsoFile 类功能说明：用于获取创建虚拟机时所需要的镜像文件

函数详细说明：

#### [1.static IsoFile\* getIsoFileInstance();](#_static__IsoFile*)

功能：获取单例对象

返回值：IsoFile指针

参数：

#### [2.void getIsoFile();](#_void___5)

功能：获取全部镜像文件

返回值：空

参数：

#### [3.void getIsoFileSignal(QString);](#_void___6)

功能：接收数据

返回值：空

参数：

### 类---VirtualMachineInfo 类功能说明：获取虚拟机详情信息

函数详细说明：

#### [1. static VirtualMachineInfo\* getVirtualMachineInfoInstance();](#_static__VirtualMachineInfo*)

功能：获取单例对象

返回值：VirtualMachineInfo指针

参数：

#### [2. void getVirtualMachineInfoString(QString hostName,QString vmName);](#_void___7)

功能：获取指定主机下的虚拟机的详情信息

返回值：空

参数： hostName---主机名 vmName---虚拟机名

#### [3. void getVirtualMachineInfoStringSignal(QString);](#_void___8)

功能：接收定主机下的虚拟机的详情信息数据的信号

返回值：空

参数：

### 类--- VMNameAndState 类功能说明：获取指定主机下的全部虚拟机和模板及其状态

b.函数详细说明：

#### [1.static VMNameAndState\* getVMNameAndStateInstance();](#_static__VMNameAndState*)

功能： 获取单例对象

返回值：VMNameAndState指针

参数：

#### [2.void getVMNameAndStateString(QString hostName);](#_void___9)

功能： 获取制定主机下的的全部虚拟机的名称和状态

返回值：空

参数：hostName---主机名

#### [3.void getVMNameAndStateStringSignal(QString);](#_void___10)

功能： 接收获取制定主机下的的全部虚拟机的名称和状态数据的信号

返回值：空

参数：

### 类---VNCInfo 类功能说明:获取VNC信息

#### [1.void getVNCInfo(QString hostName,QString vmName)](#_void_getVNCInfo(QString_hostName,QS)

功能:获取指定虚拟机的VNC端口，密码

返回值:

参数:hostName---主机名,vmName—虚拟机名

#### [2.void getVNCInfoSignals(QString)](#_void_getVNCInfoSignals(QString)_)

功能:接收数据

返回值:

参数:

## interface.host(主机操作)

### 类---HostOperation类功能说明：完成主机的基本操作（添加，移除，进入维护模式，推出维护模式，断开，重新连接）

b.函数详细说明：

#### [1.static HostOperation\* getHostOperationInstance();](#_static__HostOperation*)

功能： 获取单例对象

返回值：HostOperation指针

参数：

#### [2.void addHostToCluster(QString clusterName,QString hostName,QString userName,QString password);](#_void__addHostToCluster(QString)

功能： 添加主机到集群

返回值：空

参数：hostName--主机名 userName---用户名 password---密码

#### [3.void removeHostFromCluster(QString hostName);](#_void___11)

功能： 将主机从集群中移除

返回值：空

参数：hostName---主机名

#### [4.void enterMaintenanceMode(QString hostName);](#_void___12)

功能： 将主机进入维护模式

返回值：空

参数：hostName---主机名

#### [5.void exitMaintenanceMode(QString hostName);](#_void___13)

功能： 将主机退出维护模式

返回值：空

参数：hostName---主机名

#### [6.void disconnectHost(QString hostName);](#_void___14)

功能： 断开与主机的连接

返回值：空

参数：hostName---主机名

#### [7.void reconnectHost(QString hostName);](#_void___15)

功能： 重新与主机进行连接

返回值：空

参数：hostName---主机名

#### [8.void addHostToClusterSignal(QString);](#_void___16)

功能：添加主机到集群的信号

返回值：空

参数：

#### [9.void removeHostFromClusterSignal(QString);](#_void___17)

功能：将主机从集群中移除的信号

返回值：空

参数：

#### [10.void enterMaintenanceModeSignal(QString);](#_void___18)

功能： 将主机进入维护模式的信号

返回值：空

参数：

#### [11.void exitMaintenanceModeSignal(QString);](#_void___19)

功能： 将主机退出维护模式的信号

返回值：空

参数：

#### [12.void disconnectHostSignal(QString);](#_void___20)

功能： 断开与主机的连接的信号

返回值：空

参数：

#### [13.void reconnectHostSignal(QString);](#_void___21)

功能： 重新与主机进行连接的信号

返回值：空

参数：

### 类--- SiteFirewallDefaultPolicy 类功能说明: 获取指定指定主机的默认防火墙规则

1. 函数详细信息

#### [1.String enableFirewallDefaultPolicy(String hostName)](#_void_enableFirewallDefaultPolicy(QS)

功能: 设置防火前的默认策略true(deny)

返回值: 操作信息

参数: hostname---主机名

#### [2.String undoFirewallDefaultPolicy(String hostName)](#_void_undoFirewallDefaultPolicy(QStr)

#### 功能: 设置防火前的默认策略false(permit)

返回值: 操作信息

参数: hostname---主机名

## interface.vm(虚拟机操作)

### 类--- VMClone 类功能说明: 完成虚拟的克隆,从模板部署虚拟机

b.函数详细说明：

#### [1.static VMClone\* getVMCloneInstance();](#_static__VMClone*)

功能:获取单例对象

返回值:VMClone的指针

参数:

#### [2.void cloneVM(QString hostName, QString vmName, QString cloneName);](#_cloneVM(QString_hostName,_QString)

功能:克隆虚拟机(开机或关机)

返回值:

参数hostName---主机名,vmName---虚拟机名,cloneName---克隆机名

#### [3. void cloneTemplate(QString hostName,QString templateName, QString cloneName);](#_void___22)

功能:克隆模板 (模板是虚拟机的主镜像)

返回值:

参数: hostName---主机名, templateName ---模板名,cloneName---克隆机名

#### [4.void cloneVMToTemplate(QString hostName,QString vmName, QString cloneName) ;](#_void___23)

功能:虚拟机克隆为模板

返回值:

参数: hostName---主机名,vmName---虚拟机名,cloneName---克隆机名

#### [5. void cloneTemplatToVM(QString clusterName, QString hostName,QString templateName, QString cloneName) ;](#_void___24)

功能:从模板部署虚拟机

返回值:

参数: clusterName---集群名, hostName---主机名, templateName ---模板名,cloneName---克隆机名

#### [6.void vmToTemp( QString hostName,QString vmName);](#_void___25)

功能:将虚拟机转化为模板

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [7.void markAsVM(QString clusterName,QString hostName, QString templateName);](#_void___26)

功能: 将模板转化为虚拟机

返回值:

参数: clusterName---集群名, hostName---主机名, templateName ---模板名

### 类--- VMCreate 类功能说明: 完成虚拟的创建

b.函数详细说明:

#### [1.static VMCreate\* getVMCreateInstance();](#_static__VMCreate*)

功能:获取单例对象

返回值:VMCreate的指针

参数:

#### 2[.void createVirtualMachine(QString hostname,QString vmName,](#_void___27)

[QString guestOsId,QString numCpus,](#_void___27)

[QString diskSize,QString vmMemorySize,](#_void___27)

[QString numNics,QString dataStore,QString iso,](#_void___27)

[QString startVM);](#_void___27)

功能:创建虚拟机

返回值:

参数:hostname---主机名,vmName---虚拟机名,guestOsid---虚拟机操作系统id,

numCpus---cpu数量,diskSize---硬盘大小,vmMemorySize---虚拟机内存大小

numNics---虚拟机网卡数量,dataStore---数据存储位置,iso---虚拟机系统镜像

startVM---创建后是否启动虚拟机(startVM=1表示启动，否则表示不启动该虚拟机)

### 类---VMOtherOps 类功能说明: 完成虚拟机的一些额外的操作

b.函数详细说明:

#### [1.static VMOtherOps\* getVMOtherOpsInstance();](#_static__VMOtherOps*)

功能:获取单例对象

返回值: VMOtherOps指针

参数:

#### [2.void destroyVm(QString hostName,QString vmName);](#_void___28)

功能:删除虚拟机

返回值:

参数:hostname---主机名,vmName---虚拟机名

#### [3.void unregisterVm(QString hostName,QString vmName);](#_void___29)

功能:从清单中移除虚拟机

返回值:

参数: hostname---主机名,vmName---虚拟机名

#### [4.void renameVm(QString hostName,QString vmName, QString newVmName);](#_void___30)

功能:虚拟机重命名

返回值:

参数: hostname---主机名,vmName---虚拟机名,newVmName---新虚拟机名

### 类--- VMotion 类功能说明: 完成虚拟的迁移和重定向

b.函数详细说明:

#### [1.static VMotion\* getVMotionInstance();](#_VMotion*_getVMotionInstance();_)

功能:获取单例对象

返回值: VMotion指针

参数:

#### [2. void migrateVM(QString vmname, QString tcluster,QString tHost,](#_void___31)

[QString srcHost, QString priority);](#_void___31)

功能:对虚拟机进行迁移操作

返回值:

参数: vmname---虚拟机名, tcluster---迁移目标集群,tHost---迁移目标主机

srcHost---虚拟机源主机,priority---优先权

#### [3. void relocateVMOnSameHost(QString vmname,QString tcluster,QString tDS, QString srcHost,QString priority);](#_relocateVMOnSameHost(QString_vmname)

功能:对虚拟机在同一主机内进行重定位操作

返回值:

参数: vmname---虚拟机名, tcluster---迁移目标集群,tDS---迁移目标存储设备,

srcHost---虚拟机源主机,priority---优先权

#### [4.void relocateVMOnDifferentHost(QString vmname,QString tcluster,QString tHost,QString tDS,QString srcHost,QString priority);](#_relocateVMOnDifferentHost(QString_v)

功能:对虚拟机在不同主机内进行重定位操作

返回值:

参数: vmname---虚拟机名, tcluster---迁移目标集群, tHost---迁移目标主机,

tDS---迁移目标存储设备,srcHost---虚拟机源主机,priority---优先权

### 类--- VMPowerOps 类功能说明: 完成虚拟的基本操作

b.函数详细说明:

#### [1.static VMPowerOps\* getVMPowerOpsInstance();](#_static__VMPowerOps*)

功能:获取单例对象

返回值: VMPowerOps指针

参数:

#### [2.void powerOnVM(QString hostName,QString vmName);](#_void___32)

功能: 给指定的虚拟机打开电源

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [3.void powerOffVM(QString hostName, QString vmName);](#_void___32)

功能: 给指定的虚拟机关闭电源

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [4.void resetVM(QString hostName,QString vmName);](#_void_resetVM(QString_hostName,QStri)

功能:复位指定的虚拟机

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [5.void suspendVM(QString hostName,QString vmName);](#_void_suspendVM(QString_hostName,QSt)

功能:挂起指定的虚拟机

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [6.void rebootVM(QString hostName,QString vmName);](#_void_rebootVM(QString_hostName,QStr)

功能:重启客户机

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [7.void shutdownVM(QString hostName,QString vmName);](#_void_shutdownVM(QString_hostName,QS)

功能:关闭客户机

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [8.void standbyVM(QString hostName,QString vmName);](#_void_standbyVM(QString_hostName,QSt)

功能:启动客户机操作系统待机

返回值:

参数: hostName---主机名,vmName---虚拟机名

### 类--- VMSnapshot 类功能说明: 完成虚拟的查看快照、创建快照、删除指定快照、删除全部快照、恢复指定快照、恢复到当前快照

b.函数详细说明:

#### [1.static VMSnapshot\* getVMSnapshotInstance();](#_static__)

功能:获取单例对象

返回值: VMSnapshot的指针

参数:

#### [2.void createSnapshot(QString hostName,QString vmName,QString snapshotName,QString desc)](#_void_createSnapshot(QString_hostNam)

功能:创建虚拟机快照

返回值:

参数: hostName---主机名,vmName---虚拟机名,snapshotName---快照名,desc---描述

#### [3. void listSnapshot(QString hostName, QString vmName) ;](#_void_listSnapshot(QString_hostName,)

功能:列举指定虚拟机的当期的全部快照

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [4.void revertSnapshot(QString hostName,QString vmName,QString snapshotName);](#_void__revertSnapshot(QString)

功能:恢复到指定的快照

返回值:

参数: hostName---主机名,vmName---虚拟机名, snapshotName---快照名

#### [5.void revertToCurrentSnapshot(QString hostName,QString vmName);](#_void__revertToCurrentSnapshot(QStri)

功能:恢复到当前快照

返回值:

参数: hostName---主机名,vmName---虚拟机

#### [6.void removeAllSnapshot(QString hostName,QString vmName);](#_void__removeAllSnapshot(QString)

功能:删除全部快照

返回值:

参数: hostName---主机名,vmName---虚拟机名

#### [7.void removeSnapshot(QString hostName,QString vmName,QString snapshotName,QString removeChild);](#_void__removeSnapshot(QString)

功能:移除指定的快照

返回值:

参数: hostName---主机名,vmName---虚拟机名,snapshotName---快照名,

removeChild---移除的指定快照