

## MSM 2.3 - API additions

- List vSites
  - GET
  - Response:
  - Response codes
  - Example
  - Testing
- Query Customer vSite
  - GET
  - Response:
  - Response codes
  - Example
  - Testing
- Query Customer Subnets
  - GET
  - Response:
  - Response codes
  - Example
  - Testing
- Get Customer privateKey
  - GET
  - Response:
  - Response codes
  - Example
  - Testing
- ResizeVM
  - PUT
  - REQUEST
  - Response codes
  - Testing
- List Flavors
  - GET
  - Response:
  - Response codes
  - Example
  - Testing
- List Images
  - GET
  - Response:
  - Response codes
  - Example
  - Testing
- Create Customer
  - POST
  - Response:
  - Response codes
  - Testing
- Delete Customer
  - DELETE
  - Response codes
  - Testing
- List Customers
  - GET
  - Response:
  - Response codes
  - Example
  - Testing
- Query customer
  - GET
  - RESPONSE
  - Response codes
- Init Customer vsite
  - POST
  - Request
  - Response codes
  - Testing
- Delete Customer vsite
  - DELETE
  - Response codes

- Testing
- List Customers vSites
  - GET
  - Response:
  - Response codes
  - example
  - Testing
- List OGR/Customer vSite OGR/CPE
  - GET
  - RESPONSE:
  - Response codes
  - Testing
- List All OGR/vSite OGR/CPE
  - GET
  - RESPONSE:
  - Response codes
  - Testing
- Associate Routes in a customer vSite
  - POST
  - Request
  - Response codes
  - EXAMPLE
- Disassociate routes in a customer vSite
  - POST
  - Response codes
  - EXAMPLE
- CreateVM
  - PUT
  - REQUEST
  - RESPONSE
  - Response codes
  - EXAMPLE
- DeleteVM
  - DELETE
  - Response codes
  - EXAMPLE
- ShutdownVM
  - POST
  - Response codes
  - Testing
- BootVM
  - POST
  - Response codes
  - Testing
- RebootVM
  - POST
  - Response codes
  - Testing
- syncVM
  - POST
  - Response codes
  - Testing
- GetVM
  - POST
  - RESPONSE
  - Response codes
  - EXAMPLE
- Query route
  - GET
  - Response:
  - Response codes
  - Example
  - Testing

## List vSites

List vSites in OpenStack vSite

# GET

msm/v1.0/vSite

## Response:

```
{
  "vSites": [
    {
      "name" : <string>
    }
  ]
}
```

## Response codes

200 - OK

## Example

```
{
  "vSites": [
    {
      "name" : "reston"
    },
    {
      "name" : "hongkong"
    }
  ]
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X GET
http://localhost:8080/msm/v1.0/vSite
{
  "vSites":[
    {
      "name":"vSite2"
    }
  ]
}
200
```

# Query Customer vSite

Query a customer from a vSite in OpenStack

## GET

msm/v1.0/customer/{custId}/vSite/{vSiteId}

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The vSite ID

## Response:

```
{
  "custId" : <id:string>,
  "vSite" : {
    "vSiteId": <id:string>,
    "status": <status:string>,
    "subnets" : [
      {
        "cidr" : <cidr:string>,
        "vms" : [
          {
            "uuid" : <uuid:sting>,
            "name" : <name:string>,
            "fixedIP" : <ip:string>,
            "floatIP " : <ip:string>,
            "srId" : <srid:string>
          }
        ]
      }
    ]
  }
}
```

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## Example

```
{
  "custId" : "coke",
  "vSite" : {
    "vSiteId": "reston",
    "status": "active",
    "subnets" : []
  }
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X GET
http://localhost:8080/msm/v1.0/customer/customer_1/vSite/vSite2
{
  "vSite":{
    "vSiteId": "vSite2",
    "status": "active",
    "subnets": [
      {
        "vms": [
          {
            "fixedIP": "192.168.2.105",
            "floatIP": "",
            "name": "VM1",
            "uuid": "15297ef3-8934-4242-a2ec-62423c012659",
            "srId": "00002"
          }
        ],
        "cidr": "192.168.2.0/24"
      }
    ]
  },
  "custId": "customer_1"
}
200
```

## Query Customer Subnets

List Subnets of a customer in OpenStack

### GET

**msm/v1.0/customer/{custId}/subnet**

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems

### Response:

```
{
  "custId" : <id:string>,
  "vSite" : {
    "vSiteId": <id:string>,
    "subnets": [
      {
        "cidr" : <cidr:string>
      }
    ]
  }
}
```

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## Example

```
{
  "custId" : "coke",
  "vSite" : {
    "vSiteId": "reston",
    "subnets": [
      {
        "cidr" : "10.10.10.0/24"
      }
    ]
  }
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X GET
http://localhost:8080/msm/v1.0/customer/customer_1/subnet
{
  "vSite":[
    {
      "vSiteId":"vSite2",
      "subnets":[
        {
          "cidr":"192.168.2.0/24"
        }
      ]
    }
  ],
  "custId":"customer_1"
}

200
```

## Get Customer privateKey

Get the privateKey of a customer

### GET

**msm/v1.0/customer/{custId}/privateKey**

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems

### Response:

```
{
  "privateKey" : <string>,
}
```

### Response codes

```
200 - OK
400 - Bad Request
```

### Example

```
{
  "privateKey": "-----BEGIN RSA PRIVATE
KEY-----\nMIICXAIBAAKBgQDW7CwAxDYjmqeYJlY0Jiaai0UteN0gtMI4Jo/spRaft8ufiwow\nT9X81eOZxM
xK/2XBAVmErPXfGmV0nESfyrZfm/7QxxkWnflYRszNX83hA9IK6cIP\n2cR8ZQjk1bJEwDeBiV6MghJqgc2qBU
JtbflsmHsaXWNlOU7wxxXYmn9FpwIDAQAB\nAoGAVTwuq1MlgXNJOh+HoYDso3XORp1jXSAy0mAAvkIuucMDf0
9+B/Dxlach747L\n0oZ9TVECTop9rDBkwh1jXdfgAUroK9siJFbt9yQRzj3/wRkjkIHX0Xzz3a6viXV\nB8wt
bB84dL6t4KEM+RzV94HYAY/w3K0tgB03nsvRWMqFoBkCQQDq63KK9xsyVPPz\nnud8NQVe7GCZqzy3ilhzx+GyB
WiJs+KF/bUaMeh6QLNFs0TFNPicjzE1c+8ni+Ydx\nPusyhiDlAkeA6jVZ949i6zwLrtpK+q5ZTtOFgrcaSbrB
K+i36Z9cI9tw3Ei30ahW\nq7XN6RwF3lpAANuNfL4UUnwt4Qg89l0/mwJBAI3UTphFrh7KGWrJijgHfj8Pyonr
\nwWkudhVEhcXmtCgUR/SV9eT0CaXD2UCYX+qgo3pia0a7IV0uh65AbWfYh10CQBiv\n3vfVSVv6TFD1+YlK72
b7rMsGeeLjqJD0Ss55Wcd9lHu jvpx+3gPjefzku0oIwU5V\n1hl1RdVrsc/mQYLynP5kCQGtobqtCUC+ug/IQ6M
XMQC0M9qfyF349K0LhKS38KP2B\nnsxowAOBN8dhLdJzUwFCri+PemPH2EYixK9tTjF2JHHg=\n-----END RSA
PRIVATE KEY-----\n"
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X GET
http://localhost:8080/msm/v1.0/customer/customer_1/privateKey
{
  "privateKey": "-----BEGIN RSA PRIVATE
KEY-----\nMIICXAIBAAKBgQDW7CwAxDYjmqeYJlY0Jiaai0UteN0gtMI4Jo/spRaft8ufiwow\nT9X81eOZxM
xK/2XBAVmErPXfGmV0nESfyrZfm/7QxxkWnflYRszNX83hA9IK6cIP\n2cR8ZQjk1bJEwDeBiV6MghJqgc2qBU
JtbflsmHsaXWNlOU7wxxXYmn9FpwIDAQAB\nAoGAVTwuq1MlgXNJOh+HoYDso3XORp1jXSAy0mAAvkIuucMDf0
9+B/Dxlach747L\n0oZ9TVECTop9rDBkwh1jXdfgAUroK9siJFbt9yQRzj3/wRkjkIHX0Xzz3a6viXV\nB8wt
bB84dL6t4KEM+RzV94HYAY/w3K0tgB03nsvRWMqFoBkCQQDq63KK9xsyVPPz\nnud8NQVe7GCZqzy3ilhzx+GyB
WiJs+KF/bUaMeh6QLNFs0TFNPicjzE1c+8ni+Ydx\nPusyhiDlAkeA6jVZ949i6zwLrtpK+q5ZTtOFgrcaSbrB
K+i36Z9cI9tw3Ei30ahW\nq7XN6RwF3lpAANuNfL4UUnwt4Qg89l0/mwJBAI3UTphFrh7KGWrJijgHfj8Pyonr
\nwWkudhVEhcXmtCgUR/SV9eT0CaXD2UCYX+qgo3pia0a7IV0uh65AbWfYh10CQBiv\n3vfVSVv6TFD1+YlK72
b7rMsGeeLjqJD0Ss55Wcd9lHu jvpx+3gPjefzku0oIwU5V\n1hl1RdVrsc/mQYLynP5kCQGtobqtCUC+ug/IQ6M
XMQC0M9qfyF349K0LhKS38KP2B\nnsxowAOBN8dhLdJzUwFCri+PemPH2EYixK9tTjF2JHHg=\n-----END RSA
PRIVATE KEY-----\n" }
200
```

## ResizeVM

Resize VM

## PUT

msm/v1.0/customer/<id>/resizeVM/<uuid>/vSite/<vSiteid>

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The id retrieved via get operation
uuid	<a href="#">xsd:string</a>	The vm uuid from the create or get operation

## REQUEST



```
{
  "flavorRef": <name: string>
}
```

## Response codes

```
200 - OK
400 - Bad Request
409 - Conflict
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X PUT -d
'{"flavorRef" : "m1.tiny"}'
http://localhost:8080/msm/v1.0/customer/testCustomerId/resizeVM/testUuid/vSite/reston

200
```

## List Flavors

List Flavors in OpenStack vSite

### GET

msm/v1.0/flavors/{vsite}

Parameter	Type	Description
vSite	<a href="#">string</a>	The name of the vSite

### Response:

```
{
  "flavors": [
    {
      "name" : <string>,
      "vcpu" : <integer>,
      "disk" : <integer>,
      "ram"  : <integer.>
      "swap" : <integer>
    }
  ]
}
```

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## Example

```
{
  "flavors": [
    {
      "name" : "testvm",
      "vcpu" : 4
      "disk" : 1000,
      "ram" : 10
    }
  ]
}
```

## Testing

```
time curl -w "\n%{http_code}\n" http://localhost:8080/msm/v1.0/flavors/{vsite}

{"flavors": [{"name": "testvm", "vcpu": "4", "disk": "1000", "ram": "10"}]}
200
```

## List Images

List Images in OpenStack vSite

### GET

msm/v1.0/images{vSite}

Parameter	Type	Description
vSite	<a href="#">string</a>	The name of the vSite

### Response:

```
{
  "images": [
    {
      "name" : <string>
    }
  ]
}
```

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## Example

```
{
  "images": [{
    "name": "Ubuntu LTS"
  }]
}
```

## Testing

```
time curl -w "\n%{http_code}\n" http://localhost:8080/msm/v1.0/images/{vsite}

{"images": [{"name": "vsiteImageA"}, {"name": "vsiteImageB"}]}
200
```

## Create Customer

Add a new unique customer name/id

### POST

**msm/v1.0/customer**

```
{
  "name": <string>
}
```

## Response:

```
{
  "privateKey": <string>
}
```

## Response codes

```
201 - Created
400 - Bad Request
409 - Conflict
{
  "message" : <string>
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X POST -d '{"name" :
"customer"}' http://localhost:8080/msm/v1.0/customer
```

200

## Delete Customer

Add a new unique customer name

## DELETE

msm/v1.0/customer/{customer}

Parameter	Type	Description
name	string	Unique customer name

## Response codes

```
200 - OK
400 - Bad Request
404 - Not Found
409 - Conflict
{
  "message" : <string>
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -X DELETE
http://localhost:8080/msm/v1.0/customer/{name}

200
```

## List Customers

List All Customers

### GET

**msm/v1.0/customer**

### Response:

```
{
  "customers": [
    {
      "name": <string>
    }
  ]
}
```

### Response codes

200 - OK

```
{
  "message" : <string>
}
```

### Example

```
{
  "customers": [{
    "name": "MEF001"
  }, {
    "name": "MEF002"
  }]
}
```

## Testing

```
time curl -w "\n%{http_code}\n" http://localhost:8080/msm/v1.0/customer

{"customers": [{"name": "MEF001"}, {"name": "MEF002"}]}
200
```

## Query customer

Query a customer in OpenStack

### GET

msm/v1.0/customer/{custId}

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems

### RESPONSE

```
{
  "custId" : <id:string>,
  "vSite" : [
    {
      "vSiteId": <id:string>,
      "status": <status:name>,
      "subnets" : [
        {
          "cidr" : <cidr:string>,
          "vms" : [
            {
              "uuid" : <uuid:string>,
              "name" : <name:string>,
              "fixedIP" : <ip:string>,
              "floatIP" : <ip:string>,
              "srId" : <srid:string>
            }
          ]
        }
      ]
    }
  ]
}
```

### Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

# Init Customer vsite

Adds a customer to a vsite

## POST

msm/v1.0/customer/{customer}/vSite/{vsite}

Parameter	Type	Description
customer	string	the customers unique name
vsite	string	vsite name

## Request

```
{
  "password" : <string>,                # password in clear text used in keystone
  (optional, will be generated)
  "defaultSubnet" : <string>,            # cidr for customer vsite
  "cpe" : [
    {
      "cpeSegmentationType" : <string>,   # will be "vlan" for now
      "cpeSegmentationId" : <string>,     # vlan id
      "cpeIpAddr" : <string>,              # ip of PE facing interface
      "cpeMask" : <string>,                # netmask of PE facing interface
      "cpeASN" : <string>,                 # ASN for this cpe
      "peerIP" : <string>,                 # PE router port IP
      "peerASN" : <string>,                # ASN of Peer PE Router
    }
  ]
}
```

## Response codes

```
200 - OK
400 - Bad Request
409 - Conflict
{
  "message" : <string>
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X POST -d '{"password" :
"admin", "defaultSubnet" : "10.0.0.0/24", "cpe" : [{ "cpeSegmentationType" : "vlan",
"cpeSegmentationId" : "1", "cpeIpAddr" : "10.0.0.10", "cpeMask" : "255.255.255.0", "cpeASN" :
"64511", "peerIP" : "10.2.10.10", "peerASN" : "1" }]}'
http://localhost:8080/msm/v1.0/customer/{customer}/vSite/{vsite}

200
```

## Delete Customer vsite

Deletes a customer from a vsite

### DELETE

`msm/v1.0/customer/{customer}/vSite/{site}`

Parameter	Type	Description
name	string	Unique customer name
vsite	string	vsite name

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X DELETE -d '{} '
http://localhost:8080/msm/v1.0/customer/custId/vSite/reston

200
```

## List Customers vSites

List vsite for a customer

### GET

`msm/v1.0/customer/{customer}/vSite`

---



Parameter	Type	Description
name	string	Unique customer name

## Response:

```
{
  "vSites": [
    {
      "name": <string>,

      "status" : <string>      # active|partial active|initializing|deleting|error while
initialize phase 1|error while initialize phase 2|error while delete phase 1|error while delete
phase 2
    }
  ]
}
```

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## example

```
{
  "vSites": [{
    "name": "Reston"
  }, {
    "name": "HongKong"
  }, {
    "name": "Frankfurt"
  }]
}
```

## Testing

```
time curl -w "\n%{http_code}\n" http://localhost:8080/msm/v1.0/customer/{customer}/vSite
```

```
{"vSites": [{"name": "reston"}, {"name": "hongkong"}]}
200
```

# List OGR/Customer vSite OGR/CPE

query vSite OGR/CPE for customer

## GET

msm/v1.0/customer/{customer}/vSite/{vSite}/cpe

Parameter	Type	Description
name	string	Unique customer name
vSite	string	vSite name

## RESPONSE:

```
{
  "cpe" : [
    {
      "id" : <string>,
      "peerIP" : <string>,
      "peerASN" : <string>,
      "cpeSegmentationType" : <string>,
      "cpeSegmentationId" : <string>,
      "cpeNetworkUUID" : <string>,
      "cpeIpAddr" : <string>,
      "cpeMask" : <string>,
      "cpeASN" : <string>,
      "containerUUID" : <string>,
      "deployedNode" : <string>,
      "mgmtNetworkUUID" : <string>,
      "mgmtAddr" : <string>,
      "mgmtVlan" : <string>,
      "dataNetworkUUID" : <string>,
      "dataIP" : <string>,
      "dataVlan" : <string>,
      "aggregationZone" : <string>,
      "status" : <string>
    }
  ]
}
```

# only one will be accepted in this release.

# maybe null if not set

# maybe null if not set

# will be "vlan" for now

# vlan id

# network uuid

# ip of PE facing interfacee

# netmask of PE facing interfac

# ASN for this cpe

# openstack vm uuid

# Physical Node virtual cpe iss deployed on

# openstack uuid for mgmt ogr network

# CIDR for mgmt ogr network

# vlan id for mgmt ogr network

# openstack uuid for data ogr network

# CIDR for data ogr network

# vlan id for data ogr network

# nova zone for ogr compute node

# ogr status

## Response codes

200 - OK

400 - Bad Request

```
{
  "message" : <string>
}
```

## Testing

```
time curl -w "\n%{http_code}\n"
http://localhost:8080/msm/v1.0/customer/{customer}/vSite/{vSite}/cpe

{"cpe" : [{"id": "10","peerIP": "10.1.0.10","peerASN": "ASN","cpeSegmentationType":
"vlan","cpeSegmentationId": "1","cpeIpAddr": "10.1.0.10","cpeMask": "255.255.255.0","cpeASN":
"2"}]}
200
```

## List All OGR/vSite OGR/CPE

List All OGRs in a vSite

### GET

msm/v1.0/vSite/{vSite}/cpe

Parameter	Type	Description
vSite	string	vSite name

### RESPONSE:

```
[
{
  "cpe" : [                                     # only one will be accepted in this release.
  {
    "id" : <string>,
    "peerIP" : <string>,                        # maybe null if not set
    "peerASN" : <string>,                      # maybe null if not set
    "cpeSegmentationType" : <string>,          # will be "vlan" for now
    "cpeSegmentationId" : <string>,            # vlan id
    "cpeNetworkUUID" : <string>,               # network uuid
    "cpeIpAddr" : <string>,                    # ip of PE facing interface
    "cpeMask" : <string>,                     # netmask of PE facing interface
    "cpeASN" : <string>,                      # ASN for this cpe
    "containerUUID" : <string>,               # openstack vm uuid
    "deployedNode" : <string>,                # Physical Node virtual cpe iss deployed on
    "mgmtNetworkUUID" : <string>,             # openstack uuid for mgmt ogr network
    "mgmtAddr" : <string>,                    # CIDR for mgmt ogr network
    "mgmtVlan" : <string>,                   # vlan id for mgmt ogr network
    "dataNetworkUUID" : <string>,            # openstack uuid for data ogr network
    "dataIP" : <string>,                     # CIDR for data ogr network
    "dataVlan" : <string>,                   # vlan id for data ogr network
    "aggregationZone" : <string>,            # nova zone for ogr compute node
    "status" : <string>,                     # ogr status
  }
]
}
```

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## Testing

```
time curl -w "\n%{http_code}\n" http://localhost:8080/msm/v1.0/vSite/{vSite}/cpe

{{"cpe":[{"id":"160","peerIP":"169.254.0.1","peerASN":"3491","cpeSegmentationType":"vlan","cpeSegmentationId":"2020","cpeNetworkUUID":"8993c2f6-b641-4ff7-b5c1-6041bf8b556b","cpeIpAddr":"169.254.0.3","cpeMask":"255.255.255.248","cpeASN":"65100","containerUUID":"c69adee2-9213-4a1a-bf37-ae6df3013867","deployedNode":"os-compute07","mgmtNetworkUUID":"6dcb7867-d936-4709-bc6e-2ead2d1202f0","mgmtAddr":"172.16.6.5","mgmtVlan":"2001","dataNetworkUUID":"c3c9dd73-3083-4bdb-8dc6-24ad52d74e03","dataIP":"172.16.36.5","dataVlan":"2002","aggregationZone":"azogr1","status":"online"}]}}

200
```

## Associate Routes in a customer vSite

associate OpenStack route into MPLS

### POST

**msm/v1.0/customer/<custId>/associateRoute**

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteIdA	<a href="#">xsd:string</a>	The id of the vSite to be connected (can be an OpenStack vSite). vSite ids are externally managed names.
subnetA	<a href="#">xsd:string</a>	The cidr of the subnet to be connected.
vSiteIdB	<a href="#">xsd:string</a>	The id of the second vSite to be connected (can be an OpenStack vSite or a non-OpenStack vSite). "00000" if not an openstack vSite. eg. customer actual vSite.
subnetB	<a href="#">xsd:string</a>	The cidr of the second subnet to be connected.

## Request

```

{
  "connections": [
    {
      "vSiteIdA": <id:string>,
      "subnetA": <cidr:string>,
      "vSiteIdB": <id:string>,
      "subnetB": <cidr:string>
    }
  ]
}

```

## Response codes

```

200 - OK
400 - Bad Request
{
  "message" : <string>
}

```

## EXAMPLE

msm/v1.0/customer/coke/associateRoute

```

{
  "connections": [
    {
      "vSiteIdA": "reston",
      "subnetA": "10.0.0.0/24",
      "vSiteIdB": "hongkong",
      "subnetB": "10.20.0.0/24"
    }
  ]
}

```

msm/v1.0/customer/coke/associateRoute

```

{
  "connections": [
    {
      "vSiteIdA": "reston",
      "subnetA": "10.0.0.0/24",
      "vSiteIdB": "00000",
      "subnetB": "192.168.0.0/24"
    }
  ]
}

```

# Disassociate routes in a customer vSite

Disassociate an OpenStack route from MPLS

## POST

msm/v1.0/customer/<custId>/disassociateRoute

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteIdA	<a href="#">xsd:string</a>	The id of the vSite to be connected (can be an OpenStack vSite). vSite ids are externally managed names.
subnetA	<a href="#">xsd:string</a>	The cidr of the subnet to be connected.
vSiteIdB	<a href="#">xsd:string</a>	The id of the second vSite to be connected (can be an OpenStack vSite or a non-OpenStack vSite). "00000" if not an openstack vSite. eg. customer actual vSite.
subnetB	<a href="#">xsd:string</a>	The cidr of the second subnet to be connected.

```
{
  "connections": [
    {
      "vSiteIdA": <id:string>,
      "subnetA": <cidr:string>,
      "vSiteIdB": <id:string>,
      "subnetB": <cidr:string>
    }
  ]
}
```

## Response codes

```
200 - OK
400 - Bad Request
{
  "message" : <string>
}
```

## EXAMPLE

```
msm/v1.0/customer/coke/disassociateRoute
```

```
{
  "connections": [
    {
      "vSiteIdA": "reston",
      "subnetA": "10.0.0.0/24",
      "vSiteIdB": "hongkong",
      "subnetB": "10.20.0.0/24"
    }
  ]
}
```

```
msm/v1.0/customer/coke/disassociateRoute
```

```
{
  "connections": [
    {
      "vSiteIdA": "reston",
      "subnetA": "10.0.0.0/24",
      "vSiteIdB": "00000",
      "subnetB": "192.168.0.0/24"
    }
  ]
}
```

## CreateVM

Create and launch and VM in OpenStack vSite

## PUT

```
msm/v1.0/customer/<custId>/createVM/<vSiteId>
```

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The id retrieved via get operation
imageRef	<a href="#">xsd:string</a>	Predefined list images installed on Openstack vSites [ Ubuntu14.04, Centos6.6, ... ]
flavorRef	<a href="#">xsd:string</a>	Predefined flavor [ <a href="#">small</a> , <a href="#">medium</a> , <a href="#">large</a> , <a href="#">xlarge</a> ] which will include <ul style="list-style-type: none"><li>• vcpu #</li><li>• ram size</li><li>• disk size</li></ul>
fixedIP	<a href="#">&lt;xsd:string&gt;</a>	The internal IP of the VM (optional - ip will be generated)
floatingIP	<a href="#">&lt;xsd:boolean&gt;</a>	1/0 if this VM is to be allocated an external IP. The allocated IP will be return in the response
srId	<a href="#">&lt;xsd:string&gt;</a>	meta-data reference for SR ID
name	<a href="#">&lt;xsd:string&gt;</a>	name of the vm

## REQUEST

```
{
  "name": <name: string>,
  "imageRef": <name: string>,
  "flavorRef": <name: string>,
  "fixedIP": <ip: string>,
  "floatingIP": <1/0: boolean>,
  "srId": <srId: string>
}
```

## RESPONSE

```
{
  "custId": <id:string>,
  "vSiteId": <id:string>,
  "name": <name:string>,
  "uuid": <uuid:string>,
  "fixedIP": <ip:string>,
  "floatIP": <ip:string>,
  "srId": <srId:string>,
  "privateKey": <key:string>
}
```

## Response codes

200 - OK  
400 - Bad Request  
409 - Conflict

## EXAMPLE



```
msm/v1.0/customer/coke/createVM/reston
```

```
{
  "name": "firstVM",
  "imageRef": "Ubuntu14.04",
  "flavorRef": "m1.tiny",
  "fixedIP": "10.0.0.2",
  "floatingIP": 0,
  "srId": "2004402"
}
```

Response:

```
{
  "custId": "coke",
  "vSiteId": "reston",
  "name": "firstVM",
  "uuid": "a0f8ded9-93c6-4ee6-bf75-8154b50e7014",
  "fixedIP": "10.0.0.2",
  "floatIP": "",
  "srId": "2004402"
}
```

## DeleteVM

Remove a VM from OpenStack vSite

### DELETE

```
msm/v1.0/customer/<custId>/deleteVM/<uuid>/vSite/<vSiteId>
```

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
siteId	<a href="#">xsd:string</a>	The id retrieved via get operation
uuid	<a href="#">xsd:string</a>	The vm uuid from the create or get operation

## Response codes

200 - OK  
400 - Bad Request  
409 - Conflict

## EXAMPLE

```
msm/v1.0/customer/coke/deleteVM/a0f8ded9-93c6-4ee6-bf75-8154b50e7014/vSite/reston
```

# ShutdownVM

Turns off VM

## POST

msm/v1.0/customer/<id>/shutdownVM/<uuid>/vSite/<siteid>

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The id retrieved via get operation
uuid	<a href="#">xsd:string</a>	The vm uuid from the create or get operation

## Response codes

200 - OK  
400 - Bad Request  
409 - Conflict

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X POST  
http://localhost:8080/msm/v1.0/customer/testCustomerId/shutdownVM/testUuid/vSite/reston
```

200

# BootVM

Boot VM

## POST

msm/v1.0/customer/<id>/bootVM/<uuid>/vSite/<vSiteid>

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The id retrieved via get operation
uuid	<a href="#">xsd:string</a>	The vm uuid from the create or get operation

## Response codes

200 - OK  
400 - Bad Request  
409 - Conflict

## Testing

```
time curl -w "\n%{http_code}\n" -X POST
http://localhost:8080/msm/v1.0/customer/{customer}/bootVM/{uuid}/vSite/{vSite}

200
```

## RebootVM

Boot VM

## POST

msm/v1.0/customer/<id>/rebootVM/<uuid>/vSite/<vSiteid>

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The id retrieved via get operation
uuid	<a href="#">xsd:string</a>	The vm uuid from the create or get operation

## Response codes

200 - OK  
400 - Bad Request  
409 - Conflict

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X POST
http://localhost:8080/msm/v1.0/customer/testCustomerId/rebootVM/testUuid/vSite/reston

200
```

## syncVM

Synchronize with cache with OpenStack. (in case any VMs were created manually outside of portal).

## POST

msm/v1.0/customer/<id>/syncVMs/vSite/<vSiteid>

Parameter	Type	Description
-----------	------	-------------

custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The id retrieved via get operation

## Response codes

200 - OK  
400 - Bad Request  
409 - Conflict

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X POST -d '{}'  

http://localhost:8080/msm/v1.0/customer/testCustomerId/syncVMs/vSite/reston
```

200

## GetVM

Create and launch and VM in OpenStack vSite

## POST

**msm/v1.0/customer/<id>/getVM/<uuid>/vSite/<vSiteid>**

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The id retrieved via get operation
uuid	<a href="#">xsd:string</a>	The vm uuid from the create or get operation

## RESPONSE

```
{
  "name": <name:string>,
  "uuid": <uuid:string>,
  "fixedIP": <ip:string>,
  "floatIP": <ip:string>,
  "srId": <srId:string>,
  "privateKey": <key:string>,
  "imageRef": <image:string>,
  "flavorRef": <flavor:string>,
  "status": <status:string>
}
```

## Response codes

200 - OK  
400 - Bad Request

## EXAMPLE

```
msm/v1.0/customer/coke/getVM/a0f8ded9-93c6-4ee6-bf75-8154b50e7014/vSite/reston
{
  "name": "VM1",
  "uuid": "a0f8ded9-93c6-4ee6-bf75-8154b50e7014",
  "fixedIP": "10.0.0.2",
  "floatIP": 0,
  "srId": "2004402",
  "privateKey": "-----BEGIN RSA PRIVATE
KEY-----\nMIICXAIBAAKBgQDW7CwAxDYjmqeYJlY0Jii0UteN0gtMI4Jo/spRaft8ufiwow\nT9X81eOZxMxK/2XBAVm
ErPXfGmV0nESfyrZfm/7QxxkWnfLYRszNX83hA9IK6cIP\n2cR8ZQjk1bJEwDeBiV6MghJqgc2qBUJtbflsmHsaXWNlOU7w
xxXYmn9FpwIDAQAB\nAoGAVTwuq1MlgXNJOh+HoYDso3XORpljXSAY0mAAvkIuucMDf09+B/Dx1ach747L\nn0oZ9TVEctop
9rDBkwh1jXdfgAUroK9siJFbt9yQRzj3/wRkjktIHx0Xzz3a6viXV\nB8wtbB84dL6t4KEM+RzV94HYAY/w3K0tgB03nsvR
WMqFoBkCQQDq63KK9xsyVPPz\nnud8NQVe7GCZqzy3ilhzx+GyBWiJs+KF/bUaMeh6QLNFs0TFNPicjzE1c+8ni+Ydx\nnPus
yhiDLakEA6jVZ949i6zwLrtpK+q5ZTtOFgrcaSbrBK+i36Z9cI9tw3Ei30ahW\nnq7XN6RwF3lpAANuNF14UUnwt4Qg8910/
mwJBAI3UTphFrh7KGWrJijgHfj8Pyonr\nnwWkudhVEhcXmtCgUR/SV9eT0CaXD2UCYX+qgo3pia0a7IV0uh65AbWfYh10CQ
Biv\n3vfVSVv6TFDl+YlK72b7rMsGeeLjqJD0Ss55WcD9lHujvpx+3gPjefzkU0oIwU5V\nnlhlRdVrsc/mQYLynP5kCQGto
bqtCUC+ug/IQ6MXMQC0M9qfyF349K0LhKS38KP2B\nnsxowAObN8dhLdJzUwFCri+PemPH2EYixK9tTjF2JHHg=\n-----EN
D RSA PRIVATE KEY-----\n",
  "imageRef": "Ubuntu14.04",
  "flavorRef": "m1.tiny",
  "status": "on"
}
```

## Query route

List the router for a customer for a vSites

## GET

msm/v1.0/customer/{custId}/route/{vSiteId}

Parameter	Type	Description
custId	<a href="#">xsd:string</a>	The customer ID maintained by external systems
vSiteId	<a href="#">xsd:string</a>	The vSite ID

## Response:

```
{
  "routes": [
    {
      "vSubnet": <cidr:string>,
      "subnet": <cidr:string>
    }
  ]
}
```

## Response codes

200 - OK

## Example

```
{
  "routes": [
    {
      "vSubnet": "192.168.8.0/24",
      "subnet": "1.1.1.1/32"
    }
  ]
}
```

## Testing

```
time curl -w "\n%{http_code}\n" -H "Content-Type: application/json" -X GET
http://localhost:8080/msm/v1.0/customer/customer_1/routes/site1
{
  "routes":[
    {
      "vSubnet": "192.168.8.0/24",
      "subnet": "1.1.1.1/32"
    }
  ]
}
200
```