ONOSFW API Specification for Brahmaputra

# Overview

This document describes the API specification supported by the first phase of ONOSFW implementation intended for the OPNFV Brahmaputra release. ONOSFW provides a framework to allow ONOS to be seamlessly integrated into the OPNFV defined VIM structure. Although ONOSFW will enable many ONOS capabilities to support many NFV operations, for the Brahmaputra release, its scope is limited to provide two sets of functionalities/APIs: integrating Openstack through Neutron plug-in, and using OVSDB protocol for southbound communications. Users can use these APIs to create/modify/delete virtual tenant networks (VTNs) in OPNFV environment for service integration. In the future releases of OPNFV, more functions will be added to ONOSFW to extend its capabilities more networking and service applications.

# Description

ONOSFW/ONOS interfaces with Openstack Neutron through the onos-app-vtn APIs. These APIs is responsible to provide VTN management, to map virtual and physical, and to manage communications with Neutron. onos-app-vtn has 3 subgroups, as described in section 3:

onos-app-vtn-mgr

VtnService (class)

onos-app-vtn-rsc

FloatingIpService (class)

RouterService (class)

RouterInterfaceService (class)

SubnetService (class)

TenantNetworkService (class)

VirtualPortService (class)

onos-app-vtn-web

FloatingIpWebResource (class)

RouterWebResource (class)

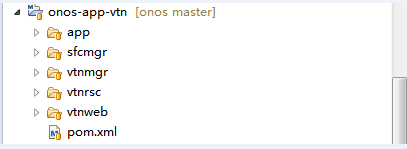
SubnetWebResource (class)

TenantNetworkResource (class)

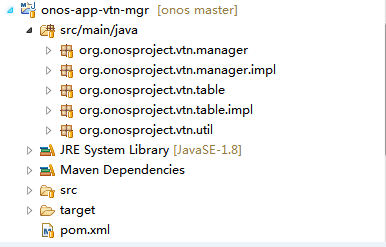
VirtualPortWebResource (class)

The source code tree is arranged as :

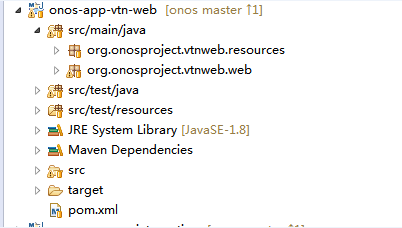
onos-app-vtn



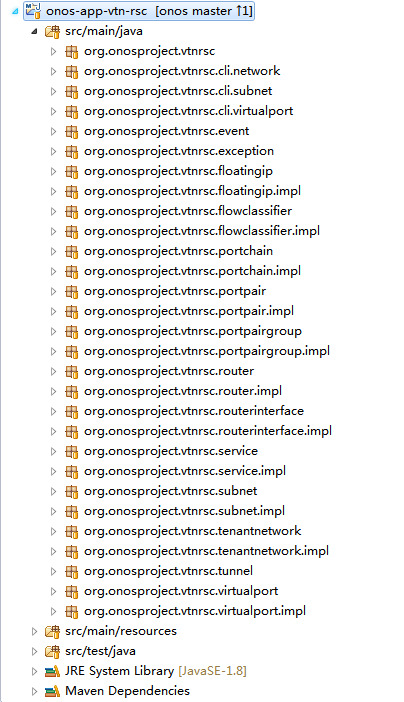
onos-app-vtn-mgr



onos-app-vtn-web



onos-app-vtn-web



The support for communications between ONOS and Open vSwitch is through OVSBD protocol, which is implemented in onos-ovsdb-api APIs, as described in section 4. The APIs has 3 subgroups:

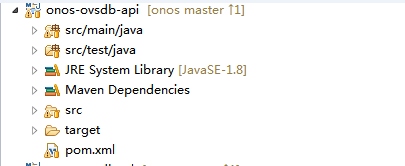
OvsdbController (class)

OvsdbClientService (class)

OvsdbProviderService (class)

The source code tree is arranged as :

onos-ovsdb-api



# onos-app-vtn APIs

These APIs provides VTN management, binds between virtual and physical element, and interfaces to communicate with Neutron.

## onos-app-vtn-mgr APIs

These functions provide interfaces to report information and messages of virtual devices, to notify VTN services of device creation and deletion, and to apply proper interface flow configuration based on the message type.

### Device operational status and event handlers

* + - 1. VtnService：Receive virtual device messages，send the configuration information or flow rule messages.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | VtnService | | | | | | | | | | | | |
| **Method Name** | onControllerDetected | **Argument** | Device controller device | | | | | | | | OVSDB node message | | |
| **Returns** | void | | | | | | | | | |  |
| **Functional Description** | When the OVSDB node is detected, this interface will be called to create ovs bridge, VXLAN tunnel，and set up OVSDB channel and OpenFlow channel. | | | | | | | | | | |
| onControllerVanished | **Argument** | Device controller device | | | | | | | OVSDB node message | | | |
| **Returns** | void | | | | | | | | |  | |
| **Functional Description** | When the OVSDB node is detected, this interface will be called to delete the VXLAN tunnel. | | | | | | | | | | |
| onOvsDetected | **Argument** | Device device | | | Ovs bridge message | | | | | | | |
| **Returns** | void | | | | | | | | |  | |
| **Functional Description** | When the ovs bridge is detected, this interface will be called to apply tunnel out flowrule. | | | | | | | | | | |
| onOvsVanished | **Argument** | Device device | | | | | | Ovs bridge message | | | | |
| **Returns** | void | | | | | |  | | | | |
| **Functional Description** | When the ovs bridge is detected, this interface will be called to delete tunnel out flowrule | | | | | | | | | | |
| onHostDetected | **Argument** | Host host | | | | Host message | | | | | | |
| **Returns** | void | | | | | | | | |  | |
| **Functional Description** | When the host is detected, this interface will be called to apply L2 and L3 flowrule. | | | | | | | | | | |
| onHostVanished | **Argument** | Host host | | | | Host message | | | | | | |
| **Returns** | void | | | | | | | | |  | |
| **Functional Description** | When the host is detected, this interface will be called to delete L2 and L3 flowrule. | | | | | | | | | | |
| onRouteInterfaceDetected | **Argument** | VtnRscEventFeedback l3Feedback | | | | L3 configation message from OpenStack | | | | | | |
| **Returns** | void |  | | | | | | | | | |
| **Functional Description** | When the route-interface is detected, This interface will be called to apply east-west flowrule. | | | | | | | | | | |
| onRouteInterfaceVanished | **Argument** | VtnRscEventFeedback l3Feedback | | | | | L3 configation message from OpenStack | | | | | |
| **Returns** | void | | | | |  | | | | | |
| **Functional Description** | When the route-interface is detected, this interface will be called to delete east-west flowrule. | | | | | | | | | | |
| programFloatingIpEvent | **Argument**  **Returns** | VtnRscEventFeedback l3Feedback | | | | | | | | L3 configation message from OpenStack | | |
| VtnRscEvent.Type type | | | | | | | | FloatingIp binding type | | |
| **Functional Description** | void | |  | | | | | | | | |
| **Argument** | When the FloatingiIp is detected, this interface will be called to apply or delete south-north flowrule. | | | | | | | | | | |

## onos-app-vtn-rsc APIs

These functions provide interfaces to handle messages from virtual device.

### Device configuration status and handlers

* + - 1. FloatingIpService，provide management of FloatingIp information.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | FloatingIpService | | | | | | | | | | | | | |
| **Method Name** | Exists | **Argument** | FloatingIpId floatingIpId | | | | | | | | Key of FloatingIp | | | |
| **Returns** | Boolean | | | | If exist return true otherwise return false. | | | | | | | |
| **Functional Description** | Check whether the special FloatingIp exist. | | | | | | | | | | | |
| floatingIpIsUsed | **Argument**  **Returns** | IpAddress floatingIpAddr | | | | | | | | Address of FloatingIp | | | |
| FloatingIpId floatingIpId | | | | | | | | Key of FloatingIp | | | |
| **Functional Description** | Boolean | | If the FloatingIp have already been used return true otherwise return false. | | | | | | | | | |
| **Argument** | Check whether the special FloatingIp has been used. | | | | | | | | | | | |
| fixIpIsUsed | **Argument**  **Returns** | IpAddress fixIpAddr | | | | | | | | | | | Internal network IP address |
| TenantId tenantId | | | | | | | | | | | Key of tenant |
| FloatingIpId floatingIpId | | | | | | | | | | | Key of FloatingIp |
| **Argument** | Boolean | | | If the FixedIp have already been used return true otherwise return false. | | | | | | | | |
| **Returns** | Check whether the special FixedIp has been used. | | | | | | | | | | | |
| getFloatingIps | **Argument** |  | | | | | |  | | | | | |
| **Returns** | Iterable<FloatingIp> | | | | | | Set of FloatingIp | | | | | |
| **Functional Description** | Get all the FloatingIp which can connect to external network. | | | | | | | | | | | |
| getFloatingIp | **Argument** | FloatingIpId floatingIpId | | | | | | | | | | Key of FloatingIp | |
| **Returns** | FloatingIp | | | | | | | | | Object of FloatingIp | | |
| **Functional Description** | Get the special FloatingIp. | | | | | | | | | | | |
| createFloatingIps | **Argument** | Iterable<FloatingIp> floatingIps | | | | | | | Set of the FloatingIp | | | | |
| **Returns** | Boolean | If create success return true otherwise return false. | | | | | | | | | | |
| **Functional Description** | Create a new FloatingIp. | | | | | | | | | | | |
| updateFloatingIps | **Argument** | Iterable<FloatingIp> floatingIps | | | | | | Set of the FloatingIp | | | | | |
| **Returns** | Boolean | If update success return true otherwise return false. | | | | | | | | | | |
| **Functional Description** | Update existing FloatingIp. | | | | | | | | | | | |
| removeFloatingIps | **Argument** | Iterable<FloatingIp> floatingIps | | | | | | | Set of the FloatingIp | | | | |
| **Returns** | boolean | If remove success return true otherwise return false. | | | | | | | | | | |
| **Functional Description** | Remove existing FloatingIp. | | | | | | | | | | | |
| addListener | **Argument** | EventuallyConsistenMapListener listener | | | | | | | | | | Map of FloatingIpId and FloatingIp | |
| **Returns** | void |  | | | | | | | | | | |
| **Functional Description** | Add FloatingIp event listener. | | | | | | | | | | | |
| removeListener | **Argument** | EventuallyConsistenMapListener listener | | | | | | | | | | Map of FloatingIpId and FloatingIp | |
| **Returns** | void | | | | |  | | | | | | |
| **Functional Description** | Remove FloatingIp event listener. | | | | | | | | | | | |

* + - 1. RouterService，provide router management service.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | RouterService | | | | | | | | | | | | | |
| **Method Name** | exists | **Argument** | RouterId routerId | | | | | | | | | | | Key of router |
| **Returns** | boolean | | | | | If exist return true otherwise return false. | | | | | | |
| **Functional Description** | Check whether the special router exist. | | | | | | | | | | | |
| getRouters | **Argument** |  | | | | | | | | | | | |
| **Returns** | Iterable<Router> | | | | Set of router. | | | | | | | |
| **Functional Description** | Get all the router. | | | | | | | | | | | |
| getRouter | **Argument** | RouterId routerId | | | | | | | Key of router | | | | |
| **Returns** | Router | | | The special router | | | | | | | | |
| **Functional Description** | Get the special router. | | | | | | | | | | | |
| createRouters | **Argument** | Iterable<Router> routers | | | | | | | | | | Set of router | |
| **Returns** | boolean | | If create success return true otherwise return false. | | | | | | | | | |
| **Functional Description** | Create a new router. | | | | | | | | | | | |
| updateRouters | **Argument** | Iterable<Router> routers | | | | | | | | | Set of router | | |
| **Returns** | boolean | | If update success return true otherwise return false. | | | | | | | | | |
| **Functional Description** | Update a new router. | | | | | | | | | | | |
| removeRouters | **Argument** | Iterable<Router> routers | | | | | | | | | | Set of router | |
| **Returns** | boolean | | If remove success return true otherwise return false. | | | | | | | | | |
| **Functional Description** | Delete a new router. | | | | | | | | | | | |
| addListener | **Argument** | EventuallyConsistenMapListener listener | | | | | | | | | | | Map of RouterId and Router |
| **Returns** | void |  | | | | | | | | | | |
| **Functional Description** | Add router event listener. | | | | | | | | | | | |
| removeListener | **Argument** | EventuallyConsistenMapListener listener | | | | | | | | Map of RouterId and Router | | | |
| **Returns** | void | | | | | |  | | | | | |
| **Functional Description** | Remove router event listener. | | | | | | | | | | | |

* + - 1. RouterInterfaceService，provide router-interface management service.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | RouterInterfaceService | | | | | | | | | | | | |
| **Method Name** | exists | **Argument** | RouterId routerId | | | | | | | | | Key of router | |
| **Returns** | boolean | | | If exist return true otherwise return false. | | | | | | | |
| **Functional Description** | Check whether the special route-interface exist. | | | | | | | | | | |
| getRouterInterfaces | **Argument** |  | | | | | | | | | | |
| **Returns** | Iterable<RouterInterface> | | | | Set of RouterInterface | | | | | | |
| **Functional Description** | Get all the route-interface. | | | | | | | | | | |
| getRouterInterfaces | **Argument** | RouterId routerId | | | | | | | | Key of router | | |
| **Returns** | RouterInterfaceStore | | | | | | | | The special RouterInterfaceStore | | |
| **Functional Description** | Get the special RouterInterfaceStore | | | | | | | | | | |
| addRouterInterface | **Argument** | RouterInterface routerInterface | | | | | | | | Object of RouterInterface | | |
| **Returns** | boolean | | If create success return true otherwise return false. | | | | | | | | |
| **Functional Description** | Create new router-interface | | | | | | | | | | |
| removeRouterInterface | **Argument** | RouterInterface routerInterface | | | | | | | Object of RouterInterface | | | |
| **Returns** | boolean | | If remove success return true otherwise return false. | | | | | | | | |
| **Functional Description** | Remove existing router-interface | | | | | | | | | | |
| addListener | **Argument** | EventuallyConsistenMapListener listener | | | | | | | | | | SubnetId与RouterInterface的map |
| **Returns** | void |  | | | | | | | | | |
| **Functional Description** | Add RouterInterface event listener. | | | | | | | | | | |
| removeListener | **Argument** | EventuallyConsistenMapListener listener | | | | | | SubnetId与RouterInterface的map | | | | |
| **Returns** | void | | | | |  | | | | | |
| **Functional Description** | Remove RouterInterface event listener. | | | | | | | | | | |

* + - 1. SubnetService，provider subnet management service.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | SubnetService | | | | | | | | | | |
| **Method Name** | exists | **Argument** | SubnetId subnetId | | | | | | | | Key of subnet |
| **Returns** | boolean | | | If exist return true otherwise return false. | | | | | |
| **Functional Description** | Check whether the special subnet exist. | | | | | | | | |
| getSubnets | **Argument** |  | | | | | | | | |
| **Returns** | Iterable<Subnet> | | | | Set of subnet | | | | |
| **Functional Description** | Get all the subnet | | | | | | | | |
| getSubnet | **Argument** | SubnetId subnetId | | | | | Key of subnet | | | |
| **Returns** | Subnet | | The special subnet | | | | | | |
| **Functional Description** | Get the special subnet. | | | | | | | | |
| createSubnets | **Argument** | Iterable<Subnet> subnets | | | | | | | Set of subnet. | |
| **Returns** | boolean | If create success return true otherwise return false. | | | | | | | |
| **Functional Description** | Create net subnet. | | | | | | | | |
| updateSubnets | **Argument** | Iterable<Subnet> subnets | | | | | | Set of subnet. | | |
| **Returns** | boolean | If update success return true otherwise return false. | | | | | | | |
| **Functional Description** | Update the existing subnet. | | | | | | | | |
| removeSubnets | **Argument** | Iterable<Subnet> subnets | | | | | | | Set of subnet | |
| **Returns** | boolean | If remove success return true otherwise return false. | | | | | | | |
| **Functional Description** | Remove the existing subnet. | | | | | | | | |

* + - 1. TenantNetworkService, provide network management service.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | TenantNetworkService | | | | | | | | | | |
| **Method Name** | exists | **Argument** | TenantNetworkId networkId | | | | | | | 租户网络network的id | |
| **Returns** | boolean | | | If exist return true otherwise return false. | | | | | |
| **Functional Description** | Check whether the special network exist. | | | | | | | | |
| getNetworkCount | **Argument** |  | | |  | | | | | |
| **Returns** | int | The number of existing TenantNetwork. | | | | | | | |
| **Functional Description** | Get the total number of existing TenantNetwork. | | | | | | | | |
| getNetworks | **Argument** |  | | | | | | | | |
| **Returns** | Iterable<TenantNetwork> | | | | | | | Set of TenantNetwork | |
| **Functional Description** | Get all the existing TenantNetwork | | | | | | | | |
| getNetwork | **Argument** | TenantNetworkId networkId | | | | | Key of TenantNetworkId | | | |
| **Returns** | TenantNetwork | | | | The Special TenantNetwork | | | | |
| **Functional Description** | Get the special TenantNetwork. | | | | | | | | |
| createNetworks | **Argument** | Iterable<TenantNetwork> networks | | | | | | | | Set of TenantNetwork |
| **Returns** | boolean | | If create success return true otherwise return false. | | | | | | |
| **Functional Description** | Create new TenantNetwork. | | | | | | | | |
| updateNetworks | **Argument** | Iterable< TenantNetwork > networks | | | | | | | Set of TenantNetwork | |
| **Returns** | boolean | | If update success return true otherwise return false. | | | | | | |
| **Functional Description** | Update existing TenantNetwork | | | | | | | | |
| removeNetworks | **Argument** | Iterable< TenantNetwork > networks | | | | | | Set of TenantNetwork | | |
| **Returns** | boolean | | If update success return true otherwise return false. | | | | | | |
| **Functional Description** | Remove existing TenantNetwork | | | | | | | | |

* + - 1. VirtualPortService, provide VirtualPortmanagement service.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | VirtualPortService | | | | | | | | | | | | | | |
| **Method Name** | exists | **Argument** | VirtualPortId virtualPortId | | | | | | | | | | | Key of VirtualPort | |
| **Returns** | boolean | | | | If exist return true otherwise return false. | | | | | | | | |
| **Functional Description** | Check whether the special virtual port exist. | | | | | | | | | | | | |
| getPort | **Argument** | VirtualPortId virtualPortId | | | | | | | | Key of VirtualPort | | | | |
| **Returns** | VirtualPort | | | The special VirtualPort | | | | | | | | | |
| **Functional Description** | Get special VirtualPort. | | | | | | | | | | | | |
| getPorts | **Argument** |  | | | | | | | | | | | | |
| **Returns** | Collection<VirtualPort> | | | | | | | | | Set of VirtualPort | | | |
| **Functional Description** | Get the existing VirtualPorts. | | | | | | | | | | | | |
| getPort | **Argument** | FixedIp fixedIP | | | | | | | | | | The special fixedIP | | |
| **Returns** | VirtualPort | The special VirtualPort. | | | | | | | | | | | |
| **Functional Description** | Get special VirtualPort. | | | | | | | | | | | | |
| getPort | **Argument**  **Returns** | TenantId tenanteId | | | | | | | | | Key of Tenant | | | |
| IpAddress ip | | | | | | | | | Host IP address | | | |
| **Functional Description** | VirtualPort | The special VirtualPort. | | | | | | | | | | | |
| **Argument** | Get special VirtualPort. | | | | | | | | | | | | |
| getPort | **Argument**  **Returns** | TenantNetworkId tenantNetworkId | | | | | | | | Key of TenantNetwork | | | | |
| IpAddress ip | | | | | | | | Host IP address | | | | |
| **Functional Description** | VirtualPort | The special VirtualPort. | | | | | | | | | | | |
| **Argument** | Get special VirtualPort. | | | | | | | | | | | | |
| getPorts | **Argument** | TenantNetworkId tenantNetworkId | | | | | | | | | | | | Key of TenantNetwork |
| **Returns** | Collection<VirtualPort> | | | | | | Set of VirtualPort | | | | | | |
| **Functional Description** | Get special VirtualPorts. | | | | | | | | | | | | |
| getPorts | **Argument** | DeviceId deviceId | | | | | | | Key of device | | | | | |
| **Returns** | Collection<VirtualPort> | | | | | | Set of VirtualPort | | | | | | |
| **Functional Description** | Get special VirtualPorts. | | | | | | | | | | | | |
| createPorts | **Argument** | Iterable<VirtualPort> virtualPorts | | | | | Set of VirtualPort | | | | | | | |
| **Returns** | boolean | | If create success return true otherwise return false. | | | | | | | | | | |
| **Functional Description** | Create a new VirtualPort. | | | | | | | | | | | | |
| updatePorts | **Argument** | Iterable<VirtualPort> virtualPorts | | | | | Set of VirtualPort | | | | | | | |
| **Returns** | boolean | | If update success return true otherwise return false. | | | | | | | | | | |
| **Functional Description** | Update the existing VirtualPort | | | | | | | | | | | | |
| removePorts | **Argument** | Iterable<VirtualPort> virtualPorts | | | | | Set of VirtualPort | | | | | | | |
| **Returns** | boolean | | If remove success return true otherwise return false. | | | | | | | | | | |
| **Functional Deing scription** | Remove the existing VirtualPort | | | | | | | | | | | | |

## onos-app-vtn-web APIs

These functions provide interface to facilitate communications between Openstack Neutron and ONOS through REST.

* + - 1. FloatingIpWebResource：FloatingIp message API

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | FloatingIpWebResource | | | | | | | | | |
| **Method Name** | listFloatingIps | **Argument** |  | | | | |  | | |
| **Returns** | Response | | | Set of FloatingIp message | | | | |
| **Functional Description** | API for providing FloatingIp message to neutron. | | | | | | | |
| getFloatingIp | **Argument**  **Returns** | String id | | | | | | Uuid of FloatingIp | |
| List<String> fields | | | | | | The special message field of FloatingIp | |
| **Functional Description** | Response | The special FloatingIp | | | | | | |
| **Argument** | API for providing special FloatingIp message to neutron. | | | | | | | |
| createFloatingIp | **Argument** | InputStream input | | | | | | Floatingip message. | |
| **Returns** | Response | | | The status of result. | | | | |
| **Functional Description** | API for accepting neutron request to create new FloatingIp | | | | | | | |
| updateFloatingIp | **Argument**  **Returns** | String id | | | | Uuid of FloatingIp | | | |
| InputStream input | | | | FloatingIp message | | | |
| **Functional Description** | Response | | The status of result. | | | | | |
| **Argument** | API for accepting neutron request to update the special FloatingIp | | | | | | | |
| deleteFloatingIp | **Argument** | String id | | | | | | | Uuid of FloatingIp |
| **Returns** | Response | The status of result. | | | | | | |
| **Functional Description** | API for accepting neutron request to remove the special FloatingIp | | | | | | | |

* + - 1. RouterWebResource：router message API

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | RouterWebResource | | | | | | | | | |
| **Method Name** | listRouters | **Argument** |  | | | |  | | | |
| **Returns** | Response | Set of router message | | | | | | |
| **Functional Description** | API for providing router message to neutron. | | | | | | | |
| getRouter | **Argument**  **Returns** | String id | | | | | | Uuid of router | |
| List<String> fields | | | | | | The special fileds of router message | |
| **Functional Description** | Response | The special router message. | | | | | | |
| **Argument** | API for providing special Router message to neutron. | | | | | | | |
| createRouter | **Argument** | InputStream input | | | | | | Message of router | |
| **Returns** | Response | The status of result. | | | | | | |
| **Functional Description** | API for accepting neutron request to create new router | | | | | | | |
| updateRouter | **Argument** | String id | | | | | | | Uuid of router |
| InputStream input | | | | | | | Router message |
| **Returns** | Response | | | The status of result. | | | | |
| **Functional Description** | API for accepting neutron request to update the special router. | | | | | | | |
| deleteSingleRouter | **Argument** | String id | | | | | | | Uuid of router |
| **Returns** | Response | | The status of result. | | | | | |
| **Functional Description** | API for accepting neutron request to remove the special router | | | | | | | |
| addRouterInterface | **Argument** | String id | | | | | Uuid of router-interface | | |
| InputStream input | | | | | Router-interface message | | |
| **Returns** | Response | | | | The status of result. | | | |
| **Argument** | API for accepting neutron request to create new router-interface | | | | | | | |
| removeRouterInterface | **Argument** | String id | | | | | | | Router-interface的uuid |
| InputStream input | | | | | | | Router-interface message |
| **Returns** | Response | | | | | | | The status of result. |
| **Functional Description** | API for accepting neutron request to remove the special router-interface. | | | | | | | |

* + - 1. SubnetWebResource：subnet message API

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | SubnetWebResource | | | | | | | | |
| **Method Name** | listSubnets | **Argument** |  | | | |  | | |
| **Returns** | Response | | Set of subnet | | | | |
| **Functional Description** | API for providing subnet message to neutron. | | | | | | |
| getSubnet | **Argument** | String id | | | | | Uuid of subnet | |
| **Returns** | Response | | The special subnet | | | | |
| **Functional Description** | API for providing special subnet message to neutron. | | | | | | |
| createSubnet | **Argument** | InputStream input | | | | | Subnet message | |
| **Returns** | Response | | The status of result. | | | | |
| **Functional Description** | API for accepting neutron request to add new subnet.. | | | | | | |
| updateSubnet | **Argument**  **Returns** | String id | | | Uuid of subnet | | | |
| InputStream input | | | Subnet message | | | |
| **Functional Description** | Response | The status of result. | | | | | |
| **Argument** | API for accepting neutron request to update the special subnet. | | | | | | |
| deleteSingleSubnet | **Argument** | String id | | | | | | Uuid of subnet |
| **Returns** | Response | | The status of result. | | | | |
| **Functional Description** | API for accepting neutron request to remove the special subnet. | | | | | | |

* + - 1. TenantNetworkWebResource：TenantNetwork message API

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | TenantNetworkWebResource | | | | | | | | | |
| **Method Name** | getNetwork | **Argument** | String id | | | | | | Uuid of network | |
| **Returns** | Response | | The special network | | | | | |
| **Functional Description** | API for providing special network message to neutron. | | | | | | | |
| createNetworks | **Argument** | InputStream input | | | | | | Network message | |
| **Returns** | Response | | | | The status of result. | | | |
| **Functional Description** | API for accepting neutron request to add new network. | | | | | | | |
| updateNetwork | **Argument**  **Returns** | String id | | | | | Uuid of network | | |
| InputStream input | | | | | Network message | | |
| **Functional Description** | Response | | | The status of result. | | | | |
| **Argument** | API for accepting neutron request to update the special network. | | | | | | | |
| deleteSingleSubnet | **Argument** | String id | | | | | | | Uuid of network |
| **Returns** | Response | The status of result. | | | | | | |
| **Functional Description** | API for accepting neutron request to remove the special network. | | | | | | | |

* + - 1. VirtualPortWebResource：VirtualPort message API

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | VirtualPortWebResource | | | | | | | | | | |
| **Method Name** | getPorts | **Argument** |  | | | | | |  | | |
| **Returns** | Response | | | | Set of port | | | | |
| **Functional Description** | API for providing all port message to neutron. | | | | | | | | |
| getPortById | **Argument** | String id | | | | | | | Uuid of port | |
| **Returns** | Response | | The special port message | | | | | | |
| **Functional Description** | API for providing special port message to neutron. | | | | | | | | |
| createPorts | **Argument** | InputStream input | | | | | | | Port message | |
| **Returns** | Response | | | | The status of result. | | | | |
| **Functional Description** | API for accepting neutron request to add new port. | | | | | | | | |
| updatePorts | **Argument**  **Returns** | String id | | | | | Uuid of port | | | |
| InputStream input | | | | | Port message | | | |
| **Functional Description** | Response | | | The status of result. | | | | | |
| **Argument** | API for accepting neutron request to update the special port. | | | | | | | | |
| deletePorts | **Argument** | String id | | | | | | | | Uuid of port |
| **Returns** | Response | The status of result. | | | | | | | |
| **Functional Description** | API for accepting neutron request to remove the special port. | | | | | | | | |

# onos-ovsdb APIs

These functions provide interfaces to facilitate communication between Open vSwitch and ONOS through OVSDB protocol.

## onos-ovsdb-api

* + - 1. OvsdbController：Get OVSDB node message and OVSDB event.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | OvsdbController | | | | | | | | | |
| **Method Name** | addNodeListener | **Argument** | OvsdbNodeListener listener | | | | | | | Node listener |
| **Returns** | void |  | | | | | | |
| **Functional Description** | Add listener of OVSDB node event. | | | | | | | |
| removeNodeListener | **Argument** | OvsdbNodeListener listener | | Node listener | | | | | |
| **Returns** | void |  | | | | | | |
| **Functional Description** | Remove listener of OVSDB node event. | | | | | | | |
| addOvsdbEventListener | **Argument** | OvsdbEventListener listener | | | | OVSDB event listener | | | |
| **Returns** | void |  | | | | | | |
| **Functional Description** | Add listener of OVSDB event. | | | | | | | |
| removeOvsdbEventListener | **Argument** | OvsdbNodeListener listener | | | | | OVSDB event listener | | |
| **Returns** | void |  | | | | | | |
| **Functional Description** | Remove listener of OVSDB event. | | | | | | | |
| getNodeIds | **Argument** |  | | |  | | | | |
| **Returns** | List<OvsdbNodeId> | | | | | | | Set of NodeId |
| **Functional Description** | Get all the NodeId. | | | | | | | |
| getOvsdbClient | **Argument** | OvsdbNodeId nodeId | | | | | | Key of Node | |
| **Returns** | OvsdbClientService | | | | | | OVSDB client | |
| **Functional Description** | Get OVSDB client | | | | | | | |

* + - 1. OvsdbClientService：Providing API of facing north bound.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | OvsdbClientService | | | | | | | | | | |
| **Method Name** | nodeId | **Argument** |  | | | | | | | |  |
| **Returns** | OvsdbNodeId | | | | Key of NodeId | | | | |
| **Functional Description** | Get NodeId of OVSDB client. | | | | | | | | |
| createTunnel | **Argument** | IpAddress srcIp | | | | | Source IP address of tunnel point. | | | |
| IpAddress dstIp | | | | | Destination IP address of tunnel point. | | | |
| **Returns** | void | |  | | | | | | |
| **Functional Description** | Create a new tunnel. | | | | | | | | |
| dropTunnel | **Argument** | IpAddress srcIp | | | | | | | Source IP address of tunnel point. | |
| IpAddress dstIp | | | | | | | Destination IP address of tunnel point. | |
| **Returns** | void |  | | | | | | | |
| **Functional Description** | Drop the existing tunnel. | | | | | | | | |
| getTunnels | **Argument** | void | | | | | |  | | |
| **Returns** | Set<OvsdbTunnel> | | | | | | Set of OvsdbTunnel | | |
| **Functional Description** | Get all the tunnel in current node. | | | | | | | | |
| createBridge | **Argument**  **Returns** | String bridgeName | | | | | | | | Bridge name |
| String dpid | | | | | | | | Dpid of bridge |
| String exPortName | | | | | | | | External port Name |
| **Argument** | void | | | | | |  | | |
| **Returns** | Apply the configuration of bridge to current node. | | | | | | | | |
| dropBridge | **Argument** | String BridgeName | | | | | | | Bridge name | |
| **Returns** | void | | | | | | |  | |
| **Functional Description** | Drop the configuration of bridge in current node. | | | | | | | | |
| getBridges | **Argument** |  | | | | | | | | |
| **Returns** | Set<OvsdbBridge> | | | | | | | Bridge message | |
| **Functional Description** | Get the bridge message. | | | | | | | | |
| createPorts | **Argument**  **Returns** | String bridgeName | | | | | | | Bridge name | |
| String portName | | | | | | | Port name | |
| **Functional Description** | void | | | | | | |  | |
| **Argument** | Apply the configuration of port to current node. | | | | | | | | |
| dropPort | **Argument**  **Returns** | String bridgeName | | | | | | | Bridge name | |
| String portName | | | | | | | Port name | |
| **Functional Description** | void | | | | | | |  | |
| **Argument** | Drop the configuration of port in current node. | | | | | | | | |
| getPorts | **Argument** |  | | | | | | | | |
| **Returns** | Set<OvsdbPort> | | | | | | | Set of port | |
| **Functional Description** | Get all the port in current bridge.. | | | | | | | | |
| isConnected | **Argument** |  | | | | | | | | |
| **Returns** | boolean | | | If connected return true otherwise return false. | | | | | |
| **Functional Description** | Check whether the Node is connected. | | | | | | | | |

* + - 1. OvsdbProviderService：Providing API of facing south bound.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | OvsdbProviderService | | | | | | | | | |
| **Method Name** | setAgent | **Argument** | OvsdbAgent agent | | | | | | | OVSDB agent |
| **Returns** | void | | |  | | | | |
| **Functional Description** | Set OVSDB agent. | | | | | | | |
| setChannel | **Argument** | Channel channel | | | | Netty channel | | | |
| **Returns** | void | |  | | | | | |
| **Functional Description** | Set Netty channel | | | | | | | |
| nodeAdded | **Argument** |  | | | | | | | |
| **Returns** | void |  | | | | | | |
| **Functional Description** | When the node added, notice the OVSDB agent. | | | | | | | |
| nodeRemoved | **Argument** |  | | | | | | | |
| **Returns** | void | | | | |  | | |
| **Functional Description** | When the node removed, notice the OVSDB agent. | | | | | | | |
| setConnection | **Argument** | boolean connected | | | | The status of connection. | | | |
| **Returns** | void | | | |  | | | |
| **Functional Description** | Set the status of connection. | | | | | | | |
| processResult | **Argument** | JsonNode result | | | | | | Ovsdb message from south. | |
| **Returns** | void | | | | | |  | |
| **Functional Description** | Process the result of OVSDB. | | | | | | | |
| processRequest | **Argument** | JsonNode request | | | | | | Ovsdb message from south. | |
| **Returns** | void | | | | | |  | |
| **Functional Description** | Process the request of OVSDB. | | | | | | | |
| setCallback | **Argument** | Callback monitorCallback | | | | | | Object of Callback | |
| **Returns** | void | | | | | |  | |
| **Functional Description** | Set Callback method. | | | | | | | |