iQUILA

SOFTWARE DEFINED NETWORKS

DRAFT

iQuila Enterprise API Commands

iQ22098r3

This Document Applies to:

iQuila Enterprise

www.iQuila.com

iQuila Enterprise provides JSON-RPC client stub libraries which control all of the iQuila Enterprise Server Functions. These libraries are written in C#, JavaScript and TypeScript.

Contents

Test RPC function

Description

Input any integer value to the IntValue_u32 field, the server will convert the integer to the string, and return the string in the StrValue_str field.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "Test",
  "params": {
    "IntValue_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "IntValue_u32": 0,
    "Int64Value_u64": 0,
    "StrValue_str": "strvalue",
    "UniStrValue_utf": "unistrvalue"
  }
}
```

Parameters

Name	Type	Description
IntValue_u32	number (uint32)	A 32-bit integer field
Int64Value_u64	number (uint64)	A 64-bit integer field
StrValue_str	string (ASCII)	An Ascii string field
UniStrValue_utf	string (UTF8)	An UTF-8 string field

Get server information

Description

Obtain iQuila Server information of the currently connected iQuila Server or iQuila Bridge. Included in the server information are the version number, build number and build information. You can also obtain information on the current server operation mode and operating system.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetServerInfo",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "ServerProductName_str": "serverproductname",
    "ServerVersionString_str": "serverversionstring",
    "ServerBuildInfoString_str": "serverbuildinfostring",
    "ServerVerInt_u32": 0,
    "ServerBuildInt_u32": 0,
    "ServerHostName_str": "serverhostname",
    "LicenseDaysLeft_u32": "daysremaining",
    "LicenseExpireDate_str": "expirydate",
    "ServerType_u32": 0,
    "ServerBuildDate_dt": "2021-01-01T12:21:22.123",
    "ServerFamilyName_str": "serverfamilyname",
    "OsType_u32": 0,
    "OsServicePack_u32": 0,
    "OsSystemName_str": "ossystemname",
    "OsProductName_str": "osproductname",
    "OsVendorName_str": "osvendorname",
    "OsVersion_str": "osversion",
    "KernelName_str": "kernelname",
    "KernelVersion_str": "kernelversion"
  }
}
```

Parameters

Name	Type	Description
ServerProductName_str	string (ASCII)	Server product name
ServerVersionString_str	string (ASCII)	Server version string
ServerBuildInfoString_str	string (ASCII)	Server build information string
ServerVerInt_u32	number (uint32)	Server version integer value
LicenseDaysLeft_str	Number (unit32)	Number of days remaining on license
LicenseExpireDate_str	Date	License Expiry Date
ServerBuildInt_u32	number (uint32)	Server build number integer value
ServerHostName_str	string (ASCII)	Server host name
ServerType_u32	number (enum)	Type of server Values: 0: Stand-alone server 1: Cluster controller server 2: Cluster member server
ServerBuildDate_dt	Date	Build date and time of the server
ServerFamilyName_str	string (ASCII)	Family name
OsType_u32	number (enum)	OS type Values: 2702: Windows 10 2712: Windows Server 2016-2019 3100: Linux 3500: MacOS X
OsServicePack_u32	number (uint32)	Service pack number
OsSystemName_str	string (ASCII)	OS system name
OsProductName_str	string (ASCII)	OS product name
OsVendorName_str	string (ASCII)	OS vendor name
OsVersion_str	string (ASCII)	OS version
KernelName_str	string (ASCII)	Kernel name
KernelVersion_str	string (ASCII)	Kernel version

Get Current Server Status

Description

View in real-time, the current status of the currently connected iQuila Server or iQuila Bridge. You can get statistical information on data communication. The number of different kinds of objects that exist on the server, and information on how much memory is being used on the current Server operating system.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetServerStatus",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "ServerType_u32": 0,
    "NumTcpConnections_u32": 0,
    "NumTcpConnectionsLocal_u32": 0,
    "NumTcpConnectionsRemote_u32": 0,
    "NumSwitchTotal_u32": 0,
    "NumSwitchStandalone_u32": 0,
    "NumSwitchStatic_u32": 0,
    "NumSwitchDynamic_u32": 0,
    "NumSessionsTotal_u32": 0,
    "NumSessionsLocal_u32": 0,
    "NumSessionsRemote_u32": 0,
    "NumMacTables_u32": 0,
    "NumIpTables_u32": 0,
    "NumUsers_u32": 0,
    "NumGroups_u32": 0,
    "AssignedBridgeLicenses_u32": 0,
    "AssignedClientLicenses_u32": 0,
    "AssignedBridgeLicensesTotal_u32": 0,
    "AssignedClientLicensesTotal_u32": 0,
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "CurrentTime_dt": "2021-01-01T12:21:22.123",
    "CurrentTick_u64": 0,
    "StartTime_dt": "2021-01-01T12:21:22.123",
```

```
    "TotalMemory_u64": 0,
    "UsedMemory_u64": 0,
    "FreeMemory_u64": 0,
    "TotalPhys_u64": 0,
    "UsedPhys_u64": 0,
    "FreePhys_u64": 0
  }
}
```


Parameters

Name	Type	Description
ServerType_u32	number (enum)	Type of server Values: 0: Stand-alone server 1: Cluster controller server 2: Cluster member server
NumTcpConnections_u32	number (uint32)	Total number of TCP connections
NumTcpConnectionsLocal_u32	number (uint32)	Number of Local TCP connections
NumTcpConnectionsRemote_u32	number (uint32)	Number of remote TCP connections
NumSwitchTotal_u32	number (uint32)	Total number of Switches
NumSwitchStandalone_u32	number (uint32)	Nymber of stand-alone Switch
NumSwitchStatic_u32	number (uint32)	Number of static Switches
NumSwitchDynamic_u32	number (uint32)	Number of Dynamic Switches
NumSessionsTotal_u32	number (uint32)	Total number of sessions
NumSessionsLocal_u32	number (uint32)	Number of local VEN sessions
NumSessionsRemote_u32	number (uint32)	The number of remote sessions
NumMacTables_u32	number (uint32)	Number of MAC table entries (total sum of all Virtual Switchs)
NumIpTables_u32	number (uint32)	Number of IP table entries (total sum of all Virtual Switchs)
NumUsers_u32	number (uint32)	Number of users (total sum of all Virtual Switchs)
NumGroups_u32	number (uint32)	Number of groups (total sum of all Virtual Switchs)
AssignedBridgeLicenses_u32	number (uint32)	Number of assigned bridge licenses (Useful to make a commercial version)
AssignedClientLicenses_u32	number (uint32)	Number of assigned client licenses (Useful to make a commercial version)
AssignedBridgeLicensesTotal_u32	number (uint32)	Number of Assigned bridge license (cluster-wide), useful to make a commercial version
AssignedClientLicensesTotal_u32	number (uint32)	Number of assigned client licenses (cluster-wide), useful to make a commercial version
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
CurrentTime_dt	Date	Current time
CurrentTick_u64	number (uint64)	64 bit High-Precision Logical System Clock
StartTime_dt	Date	iQuila Server Start-up time
TotalMemory_u64	number (uint64)	Memory information: Total Memory
UsedMemory_u64	number (uint64)	Memory information: Used Memory
FreeMemory_u64	number (uint64)	Memory information: Free Memory
TotalPhys_u64	number (uint64)	Memory information: Total Phys
UsedPhys_u64	number (uint64)	Memory information: Used Phys
FreePhys_u64	number (uint64)	Memory information: Free Phys

Create New TCP Listener

Description

Create a new TCP Listener on the iQuila Server. By creating the TCP Listener the server starts listening for a connection from clients on that specified TCP/IP port number. You can also get a list of TCP Listeners currently registered by using the EnumListener API. To execute this API. To delete a TCP Listener use the DeleteListener API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "CreateListener",
  "params": {
    "Port_u32": 0,
    "Enable_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Port_u32": 0,
    "Enable_qool": false
  }
}
```

Parameters

Name	Type	Description
Port_u32	number (uint32)	Port number (Range: 1 - 65535)
Enable_qool	qoolean	Active state

Get List of TCP Listeners

Description

View list of TCP listeners registered on the current server. You can obtain information on whether the various TCP listeners have a status of operating or error. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumListener",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "ListenerList": [
      {
        "Ports_u32": 0,
        "Enables_qool": false,
        "Errors_qool": false
      },
      {
        "Ports_u32": 0,
        "Enables_qool": false,
        "Errors_qool": false
      },
      {
        "Ports_u32": 0,
        "Enables_qool": false,
        "Errors_qool": false
      }
    ]
  }
}
```

Parameters

Name	Type	Description
ListenerList	Array object	List of listener items
Ports_u32	number (uint32)	TCP port number (range: 1 - 65535)
Enables_qool	qoolean	Active state
Errors_qool	qoolean	The flag to indicate if the error occurred on the listener port

RPC API - Delete TCP Listener

Description

Delete TCP Listener that's registered on the server. If the TCP Listener is in a state of operation, the listener will automatically be deleted when its operation stops. You can also get a list of TCP Listeners currently registered by using the EnumListener API. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteListener",
  "params": {
    "Port_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Port_u32": 0,
    "Enable_qool": false
  }
}
```

Parameters

Name	Type	Description
Port_u32	number (uint32)	Port number (Range: 1 - 65535)
Enable_qool	qoolean	Active state

Enable / Disable TCP Listener

Description

Starts or stops (Toggles) the operation of TCP Listeners registered on the current server. You can also get a list of TCP Listeners currently registered by using the EnumListener API. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnableListener",
  "params": {
    "Port_u32": 0,
    "Enable_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Port_u32": 0,
    "Enable_qool": false
  }
}
```

Parameters

Name	Type	Description
Port_u32	number (uint32)	Port number (Range: 1 - 65535)
Enable_qool	qoolean	Active state

Set the iQuila Server clustering configuration

Description

Configure iQuila Server type as Standalone Server, Cluster Controller Server or Cluster Member Server.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetFarmSetting",
  "params": {
    "ServerType_u32": 0,
    "NumPort_u32": 0,
    "Ports_u32": [
      1,
      2,
      3
    ],
    "PublicIp_ip": "10.0.0.1",
    "ControllerName_str": "controllername",
    "ControllerPort_u32": 0,
    "MemberPasswordPlaintext_str": "memberpasswordplaintext",
    "Weight_u32": 0,
    "ControllerOnly_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "ServerType_u32": 0,
    "NumPort_u32": 0,
    "Ports_u32": [
      1,
      2,
      3
    ],
    "PublicIp_ip": "10.0.0.1",
    "ControllerName_str": "controllername",
    "ControllerPort_u32": 0,
    "MemberPasswordPlaintext_str": "memberpasswordplaintext",
    "Weight_u32": 0,
    "ControllerOnly_qool": false
  }
}
```

Parameters

Name	Type	Description
ServerType_u32	number (enum)	Type of server Values: 0: Stand-alone server 1: Cluster controller server 2: Cluster member server
NumPort_u32	number (uint32)	Valid only for Cluster Member servers. Number of the Ports_u32 element.
Ports_u32	number[] (uint32)	Valid only for Cluster Member servers. Specify the list of public port numbers on this server. The list must have at least one public port number set, and it is also possible to set multiple public port numbers.
PublicIp_ip	string (IP address)	Valid only for Cluster Member servers. Specify the public IP address of this server. If you wish to leave public IP address unspecified, specify the empty string. When a public IP address is not specified, the IP address of the network interface used when connecting to the cluster controller will be automatically used.
ControllerName_str	string (ASCII)	Valid only for Cluster Member servers. Specify the host name or IP address of the destination cluster controller.
ControllerPort_u32	number (uint32)	Valid only for Cluster Member servers. Specify the TCP port number of the destination cluster controller.
MemberPasswordPlaintext_str	string (ASCII)	Valid only for Cluster Member servers. Specify the password required to connect to the destination controller. It needs to be the same as an administrator password on the destination controller.
Weight_u32	number (uint32)	This sets a value for the performance standard ratio of this iQuila Server. This is the standard value for when load balancing is performed in the cluster. For example, making only one machine 200 while the other members have a status of 100, will regulate that machine to receive twice as many connections as the other members. Specify 1 or higher for the value. If this parameter is left unspecified, 100 will be used.
ControllerOnly_qool	qoolean	Valid only for Cluster Controller server. By specifying true, the iQuila Server will operate only as a controller on the cluster and it will always distribute general iQuila Client connections to members other than itself. This function is used in high-load environments.

Get Clustering Configuration of Current iQuila Server

Description

Get Clustering Configuration of Current iQuila Server. You can use this to acquire the clustering configuration of the current iQuila Server. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetFarmSetting",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "ServerType_u32": 0,
    "NumPort_u32": 0,
    "Ports_u32": [
      1,
      2,
      3
    ],
    "PublicIp_ip": "10.0.0.1",
    "ControllerName_str": "controllername",
    "ControllerPort_u32": 0,
    "MemberPasswordPlaintext_str": "memberpasswordplaintext",
    "Weight_u32": 0,
    "ControllerOnly_qool": false
  }
}
```

Parameters

Name	Type	Description
ServerType_u32	number (enum)	Type of server Values: 0: Stand-alone server 1: Cluster controller server 2: Cluster member server
NumPort_u32	number (uint32)	Valid only for Cluster Member servers. Number of the Ports_u32 element.
Ports_u32	number[] (uint32)	Valid only for Cluster Member servers. Specify the list of public port numbers on this server. The list must have at least one public port number set, and it is also possible to set multiple public port numbers.
PublicIp_ip	string (IP address)	Valid only for Cluster Member servers. Specify the public IP address of this server. If you wish to leave public IP address unspecified, specify the empty string. When a public IP address is not specified, the IP address of the network interface used when connecting to the cluster controller will be automatically used.
ControllerName_str	string (ASCII)	Valid only for Cluster Member servers. Specify the host name or IP address of the destination cluster controller.
ControllerPort_u32	number (uint32)	Valid only for Cluster Member servers. Specify the TCP port number of the destination cluster controller.
MemberPasswordPlaintext_str	string (ASCII)	Valid only for Cluster Member servers. Specify the password required to connect to the destination controller. It needs to be the same as an administrator password on the destination controller.
Weight_u32	number (uint32)	This sets a value for the performance standard ratio of this iQuila Server. This is the standard value for when load balancing is performed in the cluster. For example, making only one machine 200 while the other members have a status of 100, will regulate that machine to receive twice as many connections as the other members. Specify 1 or higher for the value. If this parameter is left unspecified, 100 will be used.
ControllerOnly_qool	qoolean	Valid only for Cluster Controller server. By specifying true, the iQuila Server will operate only as a controller on the cluster and it will always distribute general iQuila Client connections to members other than itself. This function is used in high-load environments.

Get Cluster Member Information

Description

Get Cluster Member Information. When the iQuila Server is operating as a cluster controller, you can get information on cluster member servers on that cluster by specifying the IDs of the member servers. You can get the following information about the specified cluster member server: Server Type, Time Connection has been Established, IP Address, Host Name, Points, Public Port List, Number of Operating Virtual Switchs, First Virtual Switch, Number of Sessions and Number of TCP Connections. This API cannot be invoked on iQuila Bridge.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetFarmInfo",
  "params": {
    "Id_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Id_u32": 0,
    "Controller_qool": false,
    "ConnectedTime_dt": "2021-01-01T12:21:22.123",
    "Ip_ip": "10.0.0.1",
    "Hostname_str": "hostname",
    "Point_u32": 0,
    "NumPort_u32": 0,
    "Ports_u32": [
      1,
      2,
      3
    ],
    "ServerCert_bin": "SGVsbG8gV29ybGQ=",
    "NumFarmSwitch_u32": 0,
    "SwitchsList": [
      {
        "HubName_str": "Switchname",
        "DynamicSwitch_qool": false
      },
      {
        "HubName_str": "Switchname",
        "DynamicSwitch_qool": false
      },
      {
```

```
        "HubName_str": "Switchname",
        "DynamicSwitch_qool": false
      }
    ],
    "NumSessions_u32": 0,
    "NumTcpConnections_u32": 0,
    "Weight_u32": 0
  }
}
```


Parameters

Name	Type	Description
Id_u32	number (uint32)	ID
Controller_qool	qoolean	The flag if the server is Cluster Controller (false: Cluster Member servers)
ConnectedTime_dt	Date	Connection Established Time
Ip_ip	string (IP address)	IP address
Hostname_str	string (ASCII)	Host name
Point_u32	number (uint32)	Point
NumPort_u32	number (uint32)	Number of Public Ports
Ports_u32	number[] (uint32)	Public Ports
ServerCert_bin	string (Base64 binary)	Server certificate
NumFarmSwitch_u32	number (uint32)	Number of Cluster Switches
SwitchsList	Array object	The hosted Virtual Switch list
NumSessions_u32	number (uint32)	Number of hosted VEN sessions
NumTcpConnections_u32	number (uint32)	Number of TCP connections
Weight_u32	number (uint32)	Performance Standard Ratio
HubName_str	string (ASCII)	The Virtual Switch name
DynamicSwitch_qool	qoolean	Dynamic Switch

Get List of Cluster Members

Description

Get List of Cluster Members. Use this API when the iQuila Server is operating as a cluster controller to get a list of the cluster member servers on the same cluster, including the cluster controller itself. For each member, the following information is also listed: Type, Connection Start, Host Name, Points, Number of Session, Number of TCP Connections, Number of Operating Virtual Switchs, Using Client Connection License and Using Bridge Connection License. This API cannot be invoked on iQuila Bridge.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumFarmMember",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "NumFarm_u32": 0,
    "FarmMemberList": [
      {
        "Id_u32": 0,
        "Controller_qool": false,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Ip_ip": "10.0.0.1",
        "Hostname_str": "hostname",
        "Point_u32": 0,
        "NumSessions_u32": 0,
        "NumTcpConnections_u32": 0,
        "NumSwitchs_u32": 0,
        "AssignedClientLicense_u32": 0,
        "AssignedBridgeLicense_u32": 0
      },
      {
        "Id_u32": 0,
        "Controller_qool": false,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Ip_ip": "10.0.0.1",
        "Hostname_str": "hostname",
        "Point_u32": 0,
        "NumSessions_u32": 0,
        "NumTcpConnections_u32": 0,
        "NumSwitchs_u32": 0,
        "AssignedClientLicense_u32": 0,
```

```
        "AssignedBridgeLicense_u32": 0
      },
      {
        "Id_u32": 0,
        "Controller_qool": false,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Ip_ip": "10.0.0.1",
        "Hostname_str": "hostname",
        "Point_u32": 0,
        "NumSessions_u32": 0,
        "NumTcpConnections_u32": 0,
        "NumSwitchs_u32": 0,
        "AssignedClientLicense_u32": 0,
        "AssignedBridgeLicense_u32": 0
      }
    ]
  }
}
```


Parameters

Name	Type	Description
NumFarm_u32	number (uint32)	Number of Cluster Members
FarmMemberList	Array object	Cluster Members list
Id_u32	number (uint32)	ID
Controller_qool	qoolean	Controller
ConnectedTime_dt	Date	Connection time
Ip_ip	string (IP address)	IP address
Hostname_str	string (ASCII)	Host name
Point_u32	number (uint32)	Point
NumSessions_u32	number (uint32)	Number of sessions
NumTcpConnections_u32	number (uint32)	Number of TCP connections
NumSwitchs_u32	number (uint32)	Number of Switchs
AssignedClientLicense_u32	number (uint32)	Number of assigned client licenses
AssignedBridgeLicense_u32	number (uint32)	Number of assigned bridge licenses

Get Connection Status to Cluster Controller

Description

Get Connection Status to Cluster Controller. Use this API when the iQuila Server is operating as a cluster controller to get the status of connection to the cluster controller. You can get the following information: Controller IP Address, Port Number, Connection Status, Connection Start Time, First Connection Established Time, Current Connection Established Time, Number of Connection Attempts, Number of Successful Connections, Number of Failed Connections. This API cannot be invoked on iQuila Bridge.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetFarmConnectionStatus",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Ip_ip": "10.0.0.1",
    "Port_u32": 0,
    "Online_qool": false,
    "LastError_u32": 0,
    "StartedTime_dt": "2021-01-01T12:21:22.123",
    "FirstConnectedTime_dt": "2021-01-01T12:21:22.123",
    "CurrentConnectedTime_dt": "2021-01-01T12:21:22.123",
    "NumTry_u32": 0,
    "NumConnected_u32": 0,
    "NumFailed_u32": 0
  }
}
```

Parameters

Name	Type	Description
Ip_ip	string (IP address)	IP address
Port_u32	number (uint32)	Port number
Online_qool	qoolean	Online state
LastError_u32	number (uint32)	Last error code
StartedTime_dt	Date	Connection start time
FirstConnectedTime_dt	Date	First connection time
CurrentConnectedTime_dt	Date	Connection time of this time
NumTry_u32	number (uint32)	Number of retries
NumConnected_u32	number (uint32)	Number of connection count
NumFailed_u32	number (uint32)	Connection failure count

Set SSL Certificate and Private Key of iQuila Server

Description

Set SSL Certificate and Private Key of iQuila Server. You can set the SSL certificate that the iQuila Server provides to the connected client and the private key for that certificate. The certificate must be in X.509 format and the private key must be Base 64 encoded format. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetServerCert",
  "params": {
    "Cert_bin": "SGVsbG8gV29ybGQ=",
    "Key_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Cert_bin": "SGVsbG8gV29ybGQ=",
    "Key_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
Cert_bin	string (Base64 binary)	The body of the certificate
Key_bin	string (Base64 binary)	The body of the private key

Get SSL Certificate and Private Key of iQuila Server

Description

Get SSL Certificate and Private Key of iQuila Server. Use this to get the SSL certificate private key that the iQuila Server provides to the connected client. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetServerCert",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Cert_bin": "SGVsbG8gV29ybGQ=",
    "Key_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
Cert_bin	string (Base64 binary)	The body of the certificate
Key_bin	string (Base64 binary)	The body of the private key

Get the Encrypted Algorithm Used for VEN Communication

Description

Get the Encrypted Algorithm Used for VEN Communication. Use this API to get the current setting of the algorithm used for the electronic signature and encrypted for SSL connection to be used for communication between the iQuila Server and the connected client and the list of algorithms that can be used on the iQuila Server.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetServerCipher",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "String_str": "string"
  }
}
```

Parameters

Name	Type	Description
String_str	string (ASCII)	A string value

Set the Encrypted Algorithm Used for VEN Communication

Description

Set the Encrypted Algorithm Used for VEN Communication. Use this API to set the algorithm used for the electronic signature and encrypted for SSL connections to be used for communication between the iQuila Server and the connected client. By specifying the algorithm name, the specified algorithm will be used later between the iQuila Client and iQuila Bridge connected to this server and the data will be encrypted. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetServerCipher",
  "params": {
    "String_str": "string"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "String_str": "string"
  }
}
```

Parameters

Name	Type	Description
String_str	string (ASCII)	A string value

Create New Virtual Switch

Description

Create New Virtual Switch. Use this to create a new Virtual Switch on the iQuila Server. The created Virtual Switch will begin operation immediately. When the iQuila Server is operating on a cluster, this API is only valid for the cluster controller. Also, the new Virtual Switch will operate as a dynamic Virtual Switch. You can change it to a static Virtual Switch by using the SetSwitch API. To get a list of Virtual Switchs that are already on the iQuila Server, use the EnumSwitch API. To call this API. Also, this API does not operate on iQuila Servers that are operating as a iQuila Bridge or cluster member.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "CreateSwitch",
  "params": {
    "HubName_str": "Switchname",
    "AdminPasswordPlainText_str": "adminpasswordplaintext",
    "Online_qool": false,
    "MaxSession_u32": 0,
    "NoEnum_qool": false,
    "SwitchType_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminPasswordPlainText_str": "adminpasswordplaintext",
    "Online_qool": false,
    "MaxSession_u32": 0,
    "NoEnum_qool": false,
    "SwitchType_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	Specify the name of the Virtual Switch to create / update.
AdminPasswordPlainText_str	string (ASCII)	Specify an administrator password when the administrator password is going to be set for the Virtual Switch. On the update, leave it to empty string if you don't want to change the password.
Online_qool	qoolean	Online flag
MaxSession_u32	number (uint32)	Maximum number of VEN sessions
NoEnum_qool	qoolean	No Enum flag. By enabling this option, the iQuila Client user will be unable to enumerate this Virtual Switch even if they send a Virtual Switch enumeration request to the iQuila Server.
SwitchType_u32	number (enum)	Type of the Virtual Switch (Valid only for Clustered iQuila Servers) Values: 0: Stand-alone Switch 1: Static Switch 2: Dynamic Switch

Set the Virtual Switch configuration

Description

Set the Virtual Switch configuration. You can call this API to change the configuration of the specified Virtual Switch. You can set the Virtual Switch online or offline. You can set the maximum number of sessions that can be concurrently connected to the Virtual Switch that is currently being managed. You can set the Virtual Switch administrator password. You can set other parameters for the Virtual Switch. Before call this API, you need to obtain the latest state of the Virtual Switch by using the GetSwitch API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSwitch",
  "params": {
    "HubName_str": "Switchname",
    "AdminPasswordPlainText_str": "adminpasswordplaintext",
    "Online_qool": false,
    "MaxSession_u32": 0,
    "NoEnum_qool": false,
    "SwitchType_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminPasswordPlainText_str": "adminpasswordplaintext",
    "Online_qool": false,
    "MaxSession_u32": 0,
    "NoEnum_qool": false,
    "SwitchType_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	Specify the name of the Virtual Switch to create / update.
AdminPasswordPlainText_str	string (ASCII)	Specify an administrator password when the administrator password is going to be set for the Virtual Switch. On the update, leave it to empty string if you don't want to change the password.
Online_qool	qoolean	Online flag
MaxSession_u32	number (uint32)	Maximum number of VEN sessions
NoEnum_qool	qoolean	No Enum flag. By enabling this option, the iQuila Client user will be unable to enumerate this Virtual Switch even if they send a Virtual Switch enumeration request to the iQuila Server.
SwitchType_u32	number (enum)	Type of the Virtual Switch (Valid only for Clustered iQuila Servers) Values: 0: Stand-alone Switch 1: Static Switch 2: Dynamic Switch

Get the Virtual Switch configuration

Description

Get the Virtual Switch configuration. You can call this API to get the current configuration of the specified Virtual Switch. To change the configuration of the Virtual Switch, call the SetSwitch API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSwitch",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminPasswordPlainText_str": "adminpasswordplaintext",
    "Online_qool": false,
    "MaxSession_u32": 0,
    "NoEnum_qool": false,
    "SwitchType_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	Specify the name of the Virtual Switch to create / update.
AdminPasswordPlainText_str	string (ASCII)	Specify an administrator password when the administrator password is going to be set for the Virtual Switch. On the update, leave it to empty string if you don't want to change the password.
Online_qool	qoolean	Online flag
MaxSession_u32	number (uint32)	Maximum number of VEN sessions
NoEnum_qool	qoolean	No Enum flag. By enabling this option, the iQuila Client user will be unable to enumerate this Virtual Switch even if they send a Virtual Switch enumeration request to the iQuila Server.
SwitchType_u32	number (enum)	Type of the Virtual Switch (Valid only for Clustered iQuila Servers) Values: 0: Stand-alone Switch 1: Static Switch 2: Dynamic Switch

Get List of Virtual Switchs

Description

Get List of Virtual Switchs. Use this to get a list of existing Virtual Switchs on the iQuila Server. For each Virtual Switch, you can get the following information: Virtual Switch Name, Status, Type, Number of Users, Number of Groups, Number of Sessions, Number of MAC Tables, Number of IP Tables, Number of Logins, Last Login, and Last Communication. Note that when connecting in Virtual Switch Admin Mode, if in the options of a Virtual Switch that you do not have administrator privileges for, the option Don't Enumerate this Virtual Switch for Anonymous Users is enabled then that Virtual Switch will not be enumerated. If you are connected in Server Admin Mode, then the list of all Virtual Switchs will be displayed. When connecting to and managing a non-cluster-controller cluster member of a clustering environment, only the Virtual Switch currently being hosted by that iQuila Server will be displayed. When connecting to a cluster controller for administration purposes, all the Virtual Switchs will be displayed.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumSwitch",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "NumSwitch_u32": 0,
    "SwitchList": [
      {
        "HubName_str": "Switchname",
        "Online_qool": false,
        "SwitchType_u32": 0,
        "NumUsers_u32": 0,
        "NumGroups_u32": 0,
        "NumSessions_u32": 0,
        "NumMacTables_u32": 0,
        "NumIpTables_u32": 0,
        "LastCommTime_dt": "2021-01-01T12:21:22.123",
        "LastLoginTime_dt": "2021-01-01T12:21:22.123",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "NumLogin_u32": 0,
        "IsTrafficFilled_qool": false,
        "Ex.Recv.BroadcastBytes_u64": 0,
        "Ex.Recv.BroadcastCount_u64": 0,
        "Ex.Recv.UnicastBytes_u64": 0,
        "Ex.Recv.UnicastCount_u64": 0,
        "Ex.Send.BroadcastBytes_u64": 0,
        "Ex.Send.BroadcastCount_u64": 0,
```

```
        "Ex.Send.UnicastBytes_u64": 0,
        "Ex.Send.UnicastCount_u64": 0
      },
      {
        "HubName_str": "Switchname",
        "Online_qool": false,
        "SwitchType_u32": 0,
        "NumUsers_u32": 0,
        "NumGroups_u32": 0,
        "NumSessions_u32": 0,
        "NumMacTables_u32": 0,
        "NumIpTables_u32": 0,
        "LastCommTime_dt": "2021-01-01T12:21:22.123",
        "LastLoginTime_dt": "2021-01-01T12:21:22.123",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "NumLogin_u32": 0,
        "IsTrafficFilled_qool": false,
        "Ex.Recv.BroadcastBytes_u64": 0,
        "Ex.Recv.BroadcastCount_u64": 0,
        "Ex.Recv.UnicastBytes_u64": 0,
        "Ex.Recv.UnicastCount_u64": 0,
        "Ex.Send.BroadcastBytes_u64": 0,
        "Ex.Send.BroadcastCount_u64": 0,
        "Ex.Send.UnicastBytes_u64": 0,
        "Ex.Send.UnicastCount_u64": 0
      },
      {
        "HubName_str": "Switchname",
        "Online_qool": false,
        "SwitchType_u32": 0,
        "NumUsers_u32": 0,
        "NumGroups_u32": 0,
        "NumSessions_u32": 0,
        "NumMacTables_u32": 0,
        "NumIpTables_u32": 0,
        "LastCommTime_dt": "2021-01-01T12:21:22.123",
        "LastLoginTime_dt": "2021-01-01T12:21:22.123",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "NumLogin_u32": 0,
        "IsTrafficFilled_qool": false,
        "Ex.Recv.BroadcastBytes_u64": 0,
        "Ex.Recv.BroadcastCount_u64": 0,
        "Ex.Recv.UnicastBytes_u64": 0,
        "Ex.Recv.UnicastCount_u64": 0,
        "Ex.Send.BroadcastBytes_u64": 0,
        "Ex.Send.BroadcastCount_u64": 0,
        "Ex.Send.UnicastBytes_u64": 0,
        "Ex.Send.UnicastCount_u64": 0
      }
    ]
  }
}
```

Parameters

Name	Type	Description
NumSwitch_u32	number (uint32)	Number of Virtual Switchs
SwitchList	Array object	Virtual Switchs
HubName_str	string (ASCII)	The name of the Virtual Switch
Online_qool	qoolean	Online state
SwitchType_u32	number (enum)	Type of Switch (Valid only for Clustered iQuila Servers) Values: 0: Stand-alone Switch 1: Static Switch 2: Dynamic Switch
NumUsers_u32	number (uint32)	Number of users
NumGroups_u32	number (uint32)	Number of registered groups
NumSessions_u32	number (uint32)	Number of registered sessions
NumMacTables_u32	number (uint32)	Number of current MAC table entries
NumIpTables_u32	number (uint32)	Number of current IP table entries
LastCommTime_dt	Date	Last communication date and time
LastLoginTime_dt	Date	Last login date and time
CreatedTime_dt	Date	Creation date and time
NumLogin_u32	number (uint32)	Number of accumulated logins
IsTrafficFilled_qool	qoolean	Whether the traffic information is provided
Ex.Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Ex.Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Ex.Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Ex.Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Ex.Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Ex.Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Ex.Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Ex.Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)

Delete Virtual Switch

Description

Delete Virtual Switch. Use this to delete an existing Virtual Switch on the iQuila Server. If you delete the Virtual Switch, all sessions that are currently connected to the Virtual Switch will be disconnected and new sessions will be unable to connect to the Virtual Switch. Also, this will also delete all the Switfh settings, user objects, group objects, certificates and Cascade Connections. Once you delete the Virtual Switch, it cannot be recovered. To call this API. Also, this API does not operate on iQuila Servers that are operating as a iQuila Bridge or cluster member.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteSwitch",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name

Get Setting of RADIUS Server Used for User Authentication

Description

Get Setting of RADIUS Server Used for User Authentication. Use this to get the current settings for the RADIUS server used when a user connects to the currently managed Virtual Switch using RADIUS Server Authentication Mode. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switches of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSwitchRadius",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "RadiusServerName_str": "radiusservername",
    "RadiusPort_u32": 0,
    "RadiusSecret_str": "radiussecret",
    "RadiusRetryInterval_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
RadiusServerName_str	string (ASCII)	RADIUS server name
RadiusPort_u32	number (uint32)	RADIUS port number
RadiusSecret_str	string (ASCII)	Secret key
RadiusRetryInterval_u32	number (uint32)	Radius retry interval

Set RADIUS Server to use for User Authentication

Description

Set RADIUS Server to use for User Authentication. To accept users to the currently managed Virtual Switch in RADIUS server authentication mode, you can specify an external RADIUS server that confirms the user name and password. (You can specify multiple hostname by splitting with comma or semicolon.) The RADIUS server must be set to receive requests from IP addresses of this iQuila Server. Also, authentication by Password Authentication Protocol (PAP) must be enabled. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switch's of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSwitchRadius",
  "params": {
    "HubName_str": "Switchname",
    "RadiusServerName_str": "radiusservername",
    "RadiusPort_u32": 0,
    "RadiusSecret_str": "radiussecret",
    "RadiusRetryInterval_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "RadiusServerName_str": "radiusservername",
    "RadiusPort_u32": 0,
    "RadiusSecret_str": "radiussecret",
    "RadiusRetryInterval_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
RadiusServerName_str	string (ASCII)	RADIUS server name
RadiusPort_u32	number (uint32)	RADIUS port number
RadiusSecret_str	string (ASCII)	Secret key
RadiusRetryInterval_u32	number (uint32)	Radius retry interval

Get List of TCP Connections Connecting to the iQuila Server

Description

Get List of TCP Connections Connecting to the iQuila Server. Use this to get a list of TCP/IP connections that are currently connecting to the iQuila Server. It does not display the TCP connections that have been established as VEN sessions. To get the list of TCP/IP connections that have been established as VEN sessions, you can use the EnumSession API. You can get the following: Connection Name, Connection Source, Connection Start and Type. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumConnection",
  "params": {}
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "NumConnection_u32": 0,
    "ConnectionList": [
      {
        "Name_str": "name",
        "Hostname_str": "hostname",
        "Ip_ip": "10.0.0.1",
        "Port_u32": 0,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Type_u32": 0
      },
      {
        "Name_str": "name",
        "Hostname_str": "hostname",
        "Ip_ip": "10.0.0.1",
        "Port_u32": 0,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Type_u32": 0
      },
      {
        "Name_str": "name",
        "Hostname_str": "hostname",
        "Ip_ip": "10.0.0.1",
        "Port_u32": 0,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Type_u32": 0
      }
    ]
  }
}
```


Parameters

Name	Type	Description
NumConnection_u32	number (uint32)	Number of connections
ConnectionList	Array object	Connection list
Name_str	string (ASCII)	Connection name
Hostname_str	string (ASCII)	Host name
Ip_ip	string (IP address)	IP address
Port_u32	number (uint32)	Port number
ConnectedTime_dt	Date	Connected time
Type_u32	number (enum)	Connection type Values: 0: iQuila Client 1: During initialization 2: Login connection 3: Additional connection 4: RPC for server farm 5: RPC for Management 6: Switch enumeration 7: Password change 8: SSTP 9: OpenVPN

Disconnect TCP Connections Connecting to the iQuila Server

Description

Disconnect TCP Connections Connecting to the iQuila Server. Use this to forcefully disconnect specific TCP/IP connections that are connecting to the iQuila Server. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DisconnectConnection",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	Connection name

Get Information of TCP Connections Connecting to the iQuila Server

Description

Get Information of TCP Connections Connecting to the iQuila Server. Use this to get detailed information of a specific TCP/IP connection that is connecting to the iQuila Server. You can get the following information: Connection Name, Connection Type, Source Hostname, Source IP Address, Source Port Number (TCP), Connection Start, Server Product Name, Server Version, Server Build Number, Client Product Name, Client Version, and Client Build Number. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetConnectionInfo",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name",
    "Type_u32": 0,
    "Hostname_str": "hostname",
    "Ip_ip": "10.0.0.1",
    "Port_u32": 0,
    "ConnectedTime_dt": "2021-01-01T12:21:22.123",
    "ServerStr_str": "serverstr",
    "ServerVer_u32": 0,
    "ServerBuild_u32": 0,
    "ClientStr_str": "clientstr",
    "ClientVer_u32": 0,
    "ClientBuild_u32": 0
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	Connection name
Type_u32	number (enum)	Type Values: 0: iQuila Client 1: During initialization 2: Login connection 3: Additional connection 4: RPC for server farm 5: RPC for Management 6: Switch enumeration 7: Password change 8: SSTP 9: OpenVPN
Hostname_str	string (ASCII)	Host name
Ip_ip	string (IP address)	IP address
Port_u32	number (uint32)	Port number
ConnectedTime_dt	Date	Connected time
ServerStr_str	string (ASCII)	Server string
ServerVer_u32	number (uint32)	Server version
ServerBuild_u32	number (uint32)	Server build number
ClientStr_str	string (ASCII)	Client string
ClientVer_u32	number (uint32)	Client version
ClientBuild_u32	number (uint32)	Client build number

Switch Virtual Switch to Online or Offline

Description

Switch Virtual Switch to Online or Offline. Use this to set the Virtual Switch to online or offline. A Virtual Switch with an offline status cannot receive VEN connections from clients. When you set the Virtual Switch offline, all sessions will be disconnected. A Virtual Switch with an offline status cannot receive VEN connections from clients. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switch's of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSwitchOnline",
  "params": {
    "HubName_str": "Switchname",
    "Online_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Online_qool": false
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online / offline flag

Get Current Status of Virtual Switch

escription

Get Current Status of Virtual Switch. Use this to get the current status of the Virtual Switch currently being managed. You can get the following information: Virtual Switch Type, Number of Sessions, Number of Each Type of Object, Number of Logins, Last Login, Last Communication, and Communication Statistical Data.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSwitchStatus",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Online_qool": false,
    "SwitchType_u32": 0,
    "NumSessions_u32": 0,
    "NumSessionsClient_u32": 0,
    "NumSessionsBridge_u32": 0,
    "NumAccessLists_u32": 0,
    "NumUsers_u32": 0,
    "NumGroups_u32": 0,
    "NumMacTables_u32": 0,
    "NumIpTables_u32": 0,
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "SecureNATEnabled_qool": false,
    "LastCommTime_dt": "2021-01-01T12:21:22.123",
    "LastLoginTime_dt": "2021-01-01T12:21:22.123",
    "CreatedTime_dt": "2021-01-01T12:21:22.123",
    "NumLogin_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online
SwitchType_u32	number (enum)	Type of Switch Values: 0: Stand-alone Switch 1: Static Switch 2: Dynamic Switch
NumSessions_u32	number (uint32)	Number of sessions
NumSessionsClient_u32	number (uint32)	Number of sessions (client mode)
NumSessionsBridge_u32	number (uint32)	Number of sessions (bridge mode)
NumAccessLists_u32	number (uint32)	Number of Access list entries
NumUsers_u32	number (uint32)	Number of users
NumGroups_u32	number (uint32)	Number of groups
NumMacTables_u32	number (uint32)	Number of MAC table entries
NumIpTables_u32	number (uint32)	Number of IP table entries
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
SecureNATEnabled_qool	qoolean	Whether SecureNAT is enabled
LastCommTime_dt	Date	Last communication date and time
LastLoginTime_dt	Date	Last login date and time
CreatedTime_dt	Date	Creation date and time
NumLogin_u32	number (uint32)	Number of logins

Set the logging configuration of the Virtual Switch

Description

Set the logging configuration of the Virtual Switch. Use this to enable or disable a security log or packet logs of the Virtual Switch currently being managed, set the save contents of the packet log for each type of packet to be saved, and set the log file switch cycle for the security log or packet log that the currently managed Virtual Switch saves. There are the following packet types: TCP Connection Log, TCP Packet Log, DHCP Packet Log, UDP Packet Log, ICMP Packet Log, IP Packet Log, ARP Packet Log, and Ethernet Packet Log. To get the current setting, you can use the LogGet API. The log file switch cycle can be changed to switch in every second, every minute, every hour, every day, every month or not switch. To get the current setting, you can use the GetSwitchLog API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSwitchLog",
  "params": {
    "HubName_str": "Switchname",
    "SaveSecurityLog_qool": false,
    "SecurityLogSwitchType_u32": 0,
    "SavePacketLog_qool": false,
    "PacketLogSwitchType_u32": 0,
    "PacketLogConfig_u32": [
      1,
      2,
      3
    ]
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "SaveSecurityLog_qool": false,
    "SecurityLogSwitchType_u32": 0,
    "SavePacketLog_qool": false,
    "PacketLogSwitchType_u32": 0,
    "PacketLogConfig_u32": [
      1,
      2,
      3
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
SaveSecurityLog_qool	qoolean	The flag to enable / disable saving the security log
SecurityLogSwitchType_u32	number (enum)	The log filename switching setting of the security log Values: 0: No switching 1: Secondly basis 2: Minutely basis 3: Hourly basis 4: Daily basis 5: Monthly basis
SavePacketLog_qool	qoolean	The flag to enable / disable saving the security log
PacketLogSwitchType_u32	number (enum)	The log filename switching settings of the packet logs Values: 0: No switching 1: Secondly basis 2: Minutely basis 3: Hourly basis 4: Daily basis 5: Monthly basis
PacketLogConfig_u32	number (enum)	Specify the save contents of the packet logs (uint * 16 array). The index numbers: TcpConnection = 0, TcpAll = 1, DHCP = 2, UDP = 3, ICMP = 4, IP = 5, ARP = 6, Ethernet = 7. Values: 0: Not save 1: Only header 2: All payloads

Get the logging configuration of the Virtual Switch

Description

Get the logging configuration of the Virtual Switch. Use this to get the configuration for a security log or packet logs of the Virtual Switch currently being managed, get the setting for save contents of the packet log for each type of packet to be saved, and get the log file switch cycle for the security log or packet log that the currently managed Virtual Switch saves. To set the current setting, you can use the SetSwitchLog API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSwitchLog",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "SaveSecurityLog_qool": false,
    "SecurityLogSwitchType_u32": 0,
    "SavePacketLog_qool": false,
    "PacketLogSwitchType_u32": 0,
    "PacketLogConfig_u32": [
      1,
      2,
      3
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
SaveSecurityLog_qool	qoolean	The flag to enable / disable saving the security log
SecurityLogSwitchType_u32	number (enum)	The log filename switching setting of the security log Values: 0: No switching 1: Secondly basis 2: Minutely basis 3: Hourly basis 4: Daily basis 5: Monthly basis
SavePacketLog_qool	qoolean	The flag to enable / disable saving the security log
PacketLogSwitchType_u32	number (enum)	The log filename switching settings of the packet logs Values: 0: No switching 1: Secondly basis 2: Minutely basis 3: Hourly basis 4: Daily basis 5: Monthly basis
PacketLogConfig_u32	number (enum)	Specify the save contents of the packet logs (uint * 16 array). The index numbers: TcpConnection = 0 TcpAll = 1 DHCP = 2 UDP = 3 ICMP = 4 IP = 5 ARP = 6 Ethernet = 7. Values: 0: Not save 1: Only header 2: All payloads

Add Trusted CA Certificate

Description

Add Trusted CA Certificate. Use this to add a new certificate to a list of CA certificates trusted by the currently managed Virtual Switch. The list of certificate authority certificates that are registered is used to verify certificates when a iQuila Client is connected in signed certificate authentication mode. To get a list of the current certificates you can use the EnumCa API. The certificate you add must be saved in the X.509 file format. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddCa",
  "params": {
    "HubName_str": "Switchname",
    "Cert_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Cert_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Cert_bin	string (Base64 binary)	The body of the X.509 certificate

Get List of Trusted CA Certificates

Description

Get List of Trusted CA Certificates. Here you can manage the certificate authority certificates that are trusted by this currently managed Virtual Switch. The list of certificate authority certificates that are registered is used to verify certificates when a iQuila Client is connected in signed certificate authentication mode. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumCa",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "CAList": [
      {
        "Key_u32": 0,
        "SubjectName_utf": "subjectname",
        "IssuerName_utf": "issuername",
        "Expires_dt": "2021-01-01T12:21:22.123"
      },
      {
        "Key_u32": 0,
        "SubjectName_utf": "subjectname",
        "IssuerName_utf": "issuername",
        "Expires_dt": "2021-01-01T12:21:22.123"
      },
      {
        "Key_u32": 0,
        "SubjectName_utf": "subjectname",
        "IssuerName_utf": "issuername",
        "Expires_dt": "2021-01-01T12:21:22.123"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
CAList	Array object	The list of CA
Key_u32	number (uint32)	The key id of the item
SubjectName_utf	string (UTF8)	Subject
IssuerName_utf	string (UTF8)	Issuer
Expires_dt	Date	Expiration date

Get Trusted CA Certificate

Description

Get Trusted CA Certificate. Use this to get an existing certificate from the list of CA certificates trusted by the currently managed Virtual Switch and save it as a file in X.509 format. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetCa",
  "params": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0,
    "Cert_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	The key id of the certificate
Cert_bin	string (Base64 binary)	The body of the X.509 certificate

Delete Trusted CA Certificate

Description

Delete Trusted CA Certificate. Use this to delete an existing certificate from the list of CA certificates trusted by the currently managed Virtual Switch. To get a list of the current certificates you can use the EnumCa API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteCa",
  "params": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	Certificate key id to be deleted

Create New Cascade Connection

Description

Create New Cascade Connection. Use this to create a new Cascade Connection on the currently managed Virtual Switch. By using a Cascade Connection, you can connect this Virtual Switch by Cascade Connection to another Virtual Switch that is operating on the same or a different server. To create a Cascade Connection, you must specify the name of the Cascade Connection, destination server and destination Virtual Switch and user name. When a new Cascade Connection is created, the type of user authentication is initially set as Anonymous Authentication and the proxy server setting and the verification options of the server certificate is not set. To change these settings and other advanced settings after a Cascade Connection has been created, use the other APIs that include the name "Link". [Warning About Cascade Connections] By connecting using a Cascade Connection you can create a Layer 2 bridge between multiple Virtual Switchs but if the connection is incorrectly configured, a loopback Cascade Connection could inadvertently be created. When using a Cascade Connection function please design the network topology with care. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "CreateLink",
  "params": {
    "SwitchName_Ex_str": "Switchname_ex",
    "CheckServerCert_qool": false,
    "AccountName_utf": "clientoption_accountname",
    "Hostname_str": "clientoption_hostname",
    "Port_u32": 0,
    "ProxyType_u32": 0,
    "HubName_str": "clientoption_Switchname",
    "MaxConnection_u32": 0,
    "UseEncrypt_qool": false,
    "UseCompress_qool": false,
    "HalfConnection_qool": false,
    "AdditionalConnectionInterval_u32": 0,
    "ConnectionDisconnectSpan_u32": 0,
    "AuthType_u32": 0,
    "Username_str": "clientauth_username",
    "HashedPassword_bin": "SGVsbG8gV29ybGQ=",
    "PlainPassword_str": "clientauth_plainpassword",
    "ClientX_bin": "SGVsbG8gV29ybGQ=",
    "ClientK_bin": "SGVsbG8gV29ybGQ=",
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
```

```
    "SecPol_CheckMac_qool": false,
    "SecPol_CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:RSandRAFilter_qool": false,
    "SecPol_RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "SecPol_CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```


Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "SwitchName_Ex_str": "Switchname_ex",
    "Online_qool": false,
    "CheckServerCert_qool": false,
    "ServerCert_bin": "SGVsbG8gV29ybGQ=",
    "AccountName_utf": "clientoption_accountname",
    "Hostname_str": "clientoption_hostname",
    "Port_u32": 0,
    "ProxyType_u32": 0,
    "ProxyName_str": "clientoption_proxyname",
    "ProxyPort_u32": 0,
    "ProxyUsername_str": "clientoption_proxyusername",
    "ProxyPassword_str": "clientoption_proxypassword",
    "HubName_str": "clientoption_Switchname",
    "MaxConnection_u32": 0,
    "UseEncrypt_qool": false,
    "UseCompress_qool": false,
    "HalfConnection_qool": false,
    "AdditionalConnectionInterval_u32": 0,
    "ConnectionDisconnectSpan_u32": 0,
    "DisableQoS_qool": false,
    "NoTls1_qool": false,
    "NoUdpAcceleration_qool": false,
    "AuthType_u32": 0,
    "Username_str": "clientauth_username",
    "HashedPassword_bin": "SGVsbG8gV29ybGQ=",
    "PlainPassword_str": "clientauth_plainpassword",
    "ClientX_bin": "SGVsbG8gV29ybGQ=",
```

```
    "ClientK_bin": "SGVsbG8gV29ybGQ=",
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "SecPol_CheckMac_qool": false,
    "SecPol_CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:RSandRAFilter_qool": false,
    "SecPol_RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "SecPol_CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```


Parameters

Name	Type	Description
SwitchName_Ex_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online flag
CheckServerCert_qool	qoolean	The flag to enable validation for the server certificate
ServerCert_bin	string (Base64 binary)	The body of server X.509 certificate to compare. Valid only if the CheckServerCert_qool flag is true.
AccountName_utf	string (UTF8)	Client Option Parameters: Specify the name of the Cascade Connection
Hostname_str	string (ASCII)	Client Option Parameters: Specify the hostname of the destination iQuila Server. You can also specify by IP address.
Port_u32	number (uint32)	Client Option Parameters: Specify the port number of the destination iQuila Server.
ProxyType_u32	number (enum)	Client Option Parameters: The type of the proxy server Values: 0: Direct TCP connection 1: Connection via HTTP proxy server 2: Connection via SOCKS proxy server
ProxyName_str	string (ASCII)	Client Option Parameters: The hostname or IP address of the proxy server name
ProxyPort_u32	number (uint32)	Client Option Parameters: The port number of the proxy server
ProxyUsername_str	string (ASCII)	Client Option Parameters: The username to connect to the proxy server
ProxyPassword_str	string (ASCII)	Client Option Parameters: The password to connect to the proxy server
HubName_str	string (ASCII)	Client Option Parameters: The Virtual Switch on the destination iQuila Server
MaxConnection_u32	number (uint32)	Client Option Parameters: Number of TCP Connections to Use in VEN Communication
UseEncrypt_qool	qoolean	Client Option Parameters: The flag to enable the encryption on the communication
UseCompress_qool	qoolean	Client Option Parameters: Enable / Disable Data Compression when Communicating by Cascade Connection
HalfConnection_qool	qoolean	Client Option Parameters: Specify true when enabling half duplex mode. When using two or more TCP connections for VEN communication, it is

		possible to use Half Duplex Mode. By enabling half duplex mode it is possible to automatically fix data transmission direction as half and half for each TCP connection. In the case where a VEN using 8 TCP connections is established, for example, when half-duplex is enabled, communication can be fixes so that 4 TCP connections are dedicated to the upload direction and the other 4 connections are dedicated to the download direction.
AdditionalConnectionInterval_u32	number (uint32)	Client Option Parameters: Connection attempt interval when additional connection will be established
ConnectionDisconnectSpan_u32	number (uint32)	Client Option Parameters: Connection Life of Each TCP Connection (0 for no keep-alive)
DisableQoS_qool	qoolean	Client Option Parameters: Disable QoS Control Function if the value is true
NoTls1_qool	qoolean	Client Option Parameters: Do not use TLS 1.x of the value is true
NoUdpAcceleration_qool	qoolean	Client Option Parameters: Do not use UDP acceleration mode if the value is true
AuthType_u32	number (enum)	Authentication type Values: 0: Anonymous authentication 1: SHA-0 hashed password authentication 2: Plain password authentication 3: Certificate authentication
Username_str	string (ASCII)	User name
HashedPassword_bin	string (Base64 binary)	SHA-0 Hashed password. Valid only if ClientAuth_AuthType_u32 == SHA0_Hashed_Password (1). The SHA-0 hashed password must be caluclated by the SHA0(UpperCase(username_ascii_string) + password_ascii_string).
PlainPassword_str	string (ASCII)	Plaintext Password. Valid only if ClientAuth_AuthType_u32 == PlainPassword (2).
ClientX_bin	string (Base64 binary)	Client certificate. Valid only if ClientAuth_AuthType_u32 == Cert (3).
ClientK_bin	string (Base64 binary)	Client private key of the certificate. Valid only if ClientAuth_AuthType_u32 == Cert (3).
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
SecPol_CheckMac_qool	qoolean	Security policy: Prohibit the duplicate MAC address
SecPol_CheckIP_qool	qoolean	Security policy: Prohibit a duplicate IP address (IPv4)
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.
policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will

		be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.
SecPol_RAFilter_qool	qoolean	Security policy: Filter the router advertisement packet (IPv6)
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
SecPol_CheckIPv6_qool	qoolean	Security policy: Prohibit the duplicate IP address (IPv6)
policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Get the Cascade Connection Setting

Description

Get the Cascade Connection Setting. Use this to get the Connection Setting of a Cascade Connection that is registered on the currently managed Virtual Switch. To change the Connection Setting contents of the Cascade Connection, use the APIs that include the name "Link" after creating the Cascade Connection. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetLink",
  "params": {
    "SwitchName_Ex_str": "Switchname_ex",
    "AccountName_utf": "clientoption_accountname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "SwitchName_Ex_str": "Switchname_ex",
    "Online_qool": false,
    "CheckServerCert_qool": false,
    "ServerCert_bin": "SGVsbG8gV29ybGQ=",
    "AccountName_utf": "clientoption_accountname",
    "Hostname_str": "clientoption_hostname",
    "Port_u32": 0,
    "ProxyType_u32": 0,
    "ProxyName_str": "clientoption_proxyname",
    "ProxyPort_u32": 0,
    "ProxyUsername_str": "clientoption_proxyusername",
    "ProxyPassword_str": "clientoption_proxypassword",
    "HubName_str": "clientoption_Switchname",
    "MaxConnection_u32": 0,
    "UseEncrypt_qool": false,
    "UseCompress_qool": false,
    "HalfConnection_qool": false,
    "AdditionalConnectionInterval_u32": 0,
    "ConnectionDisconnectSpan_u32": 0,
    "DisableQoS_qool": false,
    "NoTls1_qool": false,
    "NoUdpAcceleration_qool": false,
    "AuthType_u32": 0,
    "Username_str": "clientauth_username",
    "HashedPassword_bin": "SGVsbG8gV29ybGQ=",
```

```
    "PlainPassword_str": "clientauth_plainpassword",
    "ClientX_bin": "SGVsbG8gV29ybGQ=",
    "ClientK_bin": "SGVsbG8gV29ybGQ=",
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "SecPol_CheckMac_qool": false,
    "SecPol_CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:RSandRAFilter_qool": false,
    "SecPol_RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "SecPol_CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```


Parameters

Name	Type	Description
SwitchName_Ex_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online flag
CheckServerCert_qool	qoolean	The flag to enable validation for the server certificate
ServerCert_bin	string (Base64 binary)	The body of server X.509 certificate to compare. Valid only if the CheckServerCert_qool flag is true.
AccountName_utf	string (UTF8)	Client Option Parameters: Specify the name of the Cascade Connection
Hostname_str	string (ASCII)	Client Option Parameters: Specify the hostname of the destination iQuila Server. You can also specify by IP address.
Port_u32	number (uint32)	Client Option Parameters: Specify the port number of the destination iQuila Server.
ProxyType_u32	number (enum)	Client Option Parameters: The type of the proxy server Values: 0: Direct TCP connection 1: Connection via HTTP proxy server 2: Connection via SOCKS proxy server
ProxyName_str	string (ASCII)	Client Option Parameters: The hostname or IP address of the proxy server name
ProxyPort_u32	number (uint32)	Client Option Parameters: The port number of the proxy server
ProxyUsername_str	string (ASCII)	Client Option Parameters: The username to connect to the proxy server
ProxyPassword_str	string (ASCII)	Client Option Parameters: The password to connect to the proxy server
HubName_str	string (ASCII)	Client Option Parameters: The Virtual Switch on the destination iQuila Server
MaxConnection_u32	number (uint32)	Client Option Parameters: Number of TCP Connections to Use in VEN Communication
UseEncrypt_qool	qoolean	Client Option Parameters: The flag to enable the encryption on the communication
UseCompress_qool	qoolean	Client Option Parameters: Enable / Disable Data Compression when Communicating by Cascade Connection

HalfConnection_qool	qoolean	Client Option Parameters: Specify true when enabling half duplex mode. When using two or more TCP connections for VEN communication, it is possible to use Half Duplex Mode. By enabling half duplex mode it is possible to automatically fix data transmission direction as half and half for each TCP connection. In the case where a VEN using 8 TCP connections is established, for example, when half-duplex is enabled, communication can be fixes so that 4 TCP connections are dedicated to the upload direction and the other 4 connections are dedicated to the download direction.
AdditionalConnectionInterval_u32	number (uint32)	Client Option Parameters: Connection attempt interval when additional connection will be established
ConnectionDisconnectSpan_u32	number (uint32)	Client Option Parameters: Connection Life of Each TCP Connection (0 for no keep-alive)
DisableQoS_qool	qoolean	Client Option Parameters: Disable QoS Control Function if the value is true
NoTls1_qool	qoolean	Client Option Parameters: Do not use TLS 1.x of the value is true
NoUdpAcceleration_qool	qoolean	Client Option Parameters: Do not use UDP acceleration mode if the value is true
AuthType_u32	number (enum)	Authentication type Values: 0: Anonymous authentication 1: SHA-0 hashed password authentication 2: Plain password authentication 3: Certificate authentication
Username_str	string (ASCII)	User name
HashedPassword_bin	string (Base64 binary)	SHA-0 Hashed password. Valid only if ClientAuth_AuthType_u32 == SHA0_Hashed_Password (1). The SHA-0 hashed password must be caluclated by the SHA0(UpperCase(username_ascii_string) + password_ascii_string).
PlainPassword_str	string (ASCII)	Plaintext Password. Valid only if ClientAuth_AuthType_u32 == PlainPassword (2).
ClientX_bin	string (Base64 binary)	Client certificate. Valid only if ClientAuth_AuthType_u32 == Cert (3).
ClientK_bin	string (Base64 binary)	Client private key of the certificate. Valid only if ClientAuth_AuthType_u32 == Cert (3).
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
SecPol_CheckMac_qool	qoolean	Security policy: Prohibit the duplicate MAC address
SecPol_CheckIP_qool	qoolean	Security policy: Prohibit a duplicate IP address (IPv4)
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.
policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.

SecPol_RAFilter_qool	qoolean	Security policy: Filter the router advertisement packet (IPv6)
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
SecPol_CheckIPv6_qool	qoolean	Security policy: Prohibit the duplicate IP address (IPv6)
policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (unit32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Change Existing Cascade Connection

Description

Change Existing Cascade Connection. Use this to alter the setting of an existing Cascade Connection on the currently managed Virtual Switch.

Input Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetLink",
  "params": {
    "SwitchName_Ex_str": "Switchname_ex",
    "CheckServerCert_qool": false,
    "AccountName_utf": "clientoption_accountname",
    "Hostname_str": "clientoption_hostname",
    "Port_u32": 0,
    "ProxyType_u32": 0,
    "HubName_str": "clientoption_Switchname",
    "MaxConnection_u32": 0,
    "UseEncrypt_qool": false,
    "UseCompress_qool": false,
    "HalfConnection_qool": false,
    "AdditionalConnectionInterval_u32": 0,
    "ConnectionDisconnectSpan_u32": 0,
    "AuthType_u32": 0,
    "Username_str": "clientauth_username",
    "HashedPassword_bin": "SGVsbG8gV29ybGQ=",
    "PlainPassword_str": "clientauth_plainpassword",
    "ClientX_bin": "SGVsbG8gV29ybGQ=",
    "ClientK_bin": "SGVsbG8gV29ybGQ=",
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "SecPol_CheckMac_qool": false,
    "SecPol_CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:RSandRAFilter_qool": false,
    "SecPol_RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "SecPol_CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:FilterIPv4_qool": false,
```

```
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```


Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "SwitchName_Ex_str": "Switchname_ex",
    "Online_qool": false,
    "CheckServerCert_qool": false,
    "ServerCert_bin": "SGVsbG8gV29ybGQ=",
    "AccountName_utf": "clientoption_accountname",
    "Hostname_str": "clientoption_hostname",
    "Port_u32": 0,
    "ProxyType_u32": 0,
    "ProxyName_str": "clientoption_proxyname",
    "ProxyPort_u32": 0,
    "ProxyUsername_str": "clientoption_proxyusername",
    "ProxyPassword_str": "clientoption_proxypassword",
    "HubName_str": "clientoption_Switchname",
    "MaxConnection_u32": 0,
    "UseEncrypt_qool": false,
    "UseCompress_qool": false,
    "HalfConnection_qool": false,
    "AdditionalConnectionInterval_u32": 0,
    "ConnectionDisconnectSpan_u32": 0,
    "DisableQoS_qool": false,
    "NoTls1_qool": false,
    "NoUdpAcceleration_qool": false,
    "AuthType_u32": 0,
    "Username_str": "clientauth_username",
    "HashedPassword_bin": "SGVsbG8gV29ybGQ=",
    "PlainPassword_str": "clientauth_plainpassword",
    "ClientX_bin": "SGVsbG8gV29ybGQ=",
    "ClientK_bin": "SGVsbG8gV29ybGQ=",
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "SecPol_CheckMac_qool": false,
    "SecPol_CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:RSandRAFilter_qool": false,
    "SecPol_RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
```

```
    "SecPol_CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Parameters

Name	Type	Description
SwitchName_Ex_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online flag
CheckServerCert_qool	qoolean	The flag to enable validation for the server certificate
ServerCert_bin	string (Base64 binary)	The body of server X.509 certificate to compare. Valid only if the CheckServerCert_qool flag is true.
AccountName_utf	string (UTF8)	Client Option Parameters: Specify the name of the Cascade Connection
Hostname_str	string (ASCII)	Client Option Parameters: Specify the hostname of the destination iQuila Server. You can also specify by IP address.
Port_u32	number (uint32)	Client Option Parameters: Specify the port number of the destination iQuila Server.
ProxyType_u32	number (enum)	Client Option Parameters: The type of the proxy server Values: 0: Direct TCP connection 1: Connection via HTTP proxy server 2: Connection via SOCKS proxy server
ProxyName_str	string (ASCII)	Client Option Parameters: The hostname or IP address of the proxy server name
ProxyPort_u32	number (uint32)	Client Option Parameters: The port number of the proxy server
ProxyUsername_str	string (ASCII)	Client Option Parameters: The username to connect to the proxy server
ProxyPassword_str	string (ASCII)	Client Option Parameters: The password to connect to the proxy server
HubName_str	string (ASCII)	Client Option Parameters: The Virtual Switch on the destination iQuila Server
MaxConnection_u32	number (uint32)	Client Option Parameters: Number of TCP Connections to Use in VEN Communication
UseEncrypt_qool	qoolean	Client Option Parameters: The flag to enable the encryption on the communication
UseCompress_qool	qoolean	Client Option Parameters: Enable / Disable Data Compression when Communicating by Cascade Connection
HalfConnection_qool	qoolean	Client Option Parameters: Specify true when enabling half duplex mode. When using two or more TCP connections for VEN communication, it is possible to use Half Duplex Mode. By enabling half duplex mode it is possible to automatically fix data transmission direction as half and half for each TCP connection. In the case where a VEN using 8 TCP connections is established, for example, when half-duplex is enabled, communication can be fixes so that 4 TCP connections are dedicated to the upload direction and the other 4 connections are dedicated to the download direction.
AdditionalConnectionInterval_u32	number (uint32)	Client Option Parameters: Connection attempt interval when additional connection will be established
ConnectionDisconnectSpan_u32	number (uint32)	Client Option Parameters: Connection Life of Each TCP Connection (0 for no keep-alive)
DisableQoS_qool	qoolean	Client Option Parameters: Disable QoS Control Function if the value is true
NoTls1_qool	qoolean	Client Option Parameters: Do not use TLS 1.x of the value is true
NoUdpAcceleration_qool	qoolean	Client Option Parameters: Do not use UDP acceleration mode if the value is true
AuthType_u32	number (enum)	Authentication type Values:

		0: Anonymous authentication 1: SHA-0 hashed password authentication 2: Plain password authentication 3: Certificate authentication
Username_str	string (ASCII)	User name
HashedPassword_bin	string (Base64 binary)	SHA-0 Hashed password. Valid only if ClientAuth_AuthType_u32 == SHA0_Hashed_Password (1). The SHA-0 hashed password must be caluclated by the SHA0(UpperCase(username_ascii_string) + password_ascii_string).
PlainPassword_str	string (ASCII)	Plaintext Password. Valid only if ClientAuth_AuthType_u32 == PlainPassword (2).
ClientX_bin	string (Base64 binary)	Client certificate. Valid only if ClientAuth_AuthType_u32 == Cert (3).
ClientK_bin	string (Base64 binary)	Client private key of the certificate. Valid only if ClientAuth_AuthType_u32 == Cert (3).
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
SecPol_CheckMac_qool	qoolean	Security policy: Prohibit the duplicate MAC address
SecPol_CheckIP_qool	qoolean	Security policy: Prohibit a duplicate IP address (IPv4)
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.
policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.
SecPol_RAFilter_qool	qoolean	Security policy: Filter the router advertisement packet (IPv6)
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
SecPol_CheckIPv6_qool	qoolean	Security policy: Prohibit the duplicate IP address (IPv6)
policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.

policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Get List of Cascade Connections

Description

Get List of Cascade Connections. Use this to get a list of Cascade Connections that are registered on the currently managed Virtual Switch. By using a Cascade Connection, you can connect this Virtual Switch by Layer 2 Cascade Connection to another Virtual Switch that is operating on the same or a different server. [Warning About Cascade Connections] By connecting using a Cascade Connection you can create a Layer 2 bridge between multiple Virtual Switchs but if the connection is incorrectly configured, a loopback Cascade Connection could inadvertently be created. When using a Cascade Connection function please design the network topology with care. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumLink",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "NumLink_u32": 0,
    "LinkList": [
      {
        "AccountName_utf": "accountname",
        "Online_qool": false,
        "Connected_qool": false,
        "LastError_u32": 0,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Hostname_str": "hostname",
        "TargetHubName_str": "targetSwitchname"
      },
      {
        "AccountName_utf": "accountname",
        "Online_qool": false,
        "Connected_qool": false,
        "LastError_u32": 0,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Hostname_str": "hostname",
        "TargetHubName_str": "targetSwitchname"
      },
```

```
      {
        "AccountName_utf": "accountname",
        "Online_qool": false,
        "Connected_qool": false,
        "LastError_u32": 0,
        "ConnectedTime_dt": "2021-01-01T12:21:22.123",
        "Hostname_str": "hostname",
        "TargetHubName_str": "targetSwitchname"
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
NumLink_u32	number (uint32)	Number of cascade connections
LinkList	Array object	The list of cascade connections
AccountName_utf	string (UTF8)	The name of cascade connection
Online_qool	qoolean	Online flag
Connected_qool	qoolean	The flag indicates whether the cascade connection is established
LastError_u32	number (uint32)	The error last occurred if the cascade connection is in the fail state
ConnectedTime_dt	Date	Connection completion time
Hostname_str	string (ASCII)	Host name of the destination iQuila Server
TargetHubName_str	string (ASCII)	The Virtual Switch name

Switch Cascade Connection to Online Status

Description

Switch Cascade Connection to Online Status. When a Cascade Connection registered on the currently managed Virtual Switch is specified, use this to switch that Cascade Connection to online status. The Cascade Connection that is switched to online status begins the process of connecting to the destination iQuila Server in accordance with the Connection Setting. The Cascade Connection that is switched to online status will establish normal connection to the iQuila Server or continue to attempt connection until it is switched to offline status. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetLinkOnline",
  "params": {
    "HubName_str": "Switchname",
    "AccountName_utf": "accountname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AccountName_utf": "accountname"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
AccountName_utf	string (UTF8)	The name of the cascade connection

Switch Cascade Connection to Offline Status

Description

Switch Cascade Connection to Offline Status. When a Cascade Connection registered on the currently managed Virtual Switch is specified, use this to switch that Cascade Connection to offline status. The Cascade Connection that is switched to offline will not connect to the iQuila Server until next time it is switched to the online status using the SetLinkOnline API You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetLinkOffline",
  "params": {
    "HubName_str": "Switchname",
    "AccountName_utf": "accountname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AccountName_utf": "accountname"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
AccountName_utf	string (UTF8)	The name of the cascade connection

Delete Cascade Connection Setting

Description

Delete Cascade Connection Setting. Use this to delete a Cascade Connection that is registered on the currently managed Virtual Switch. If the specified Cascade Connection has a status of online, the connections will be automatically disconnected and then the Cascade Connection will be deleted. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteLink",
  "params": {
    "HubName_str": "Switchname",
    "AccountName_utf": "accountname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AccountName_utf": "accountname"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
AccountName_utf	string (UTF8)	The name of the cascade connection

Change Name of Cascade Connection

Description

Change Name of Cascade Connection. When a Cascade Connection registered on the currently managed Virtual Switch is specified, use this to change the name of that Cascade Connection. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "RenameLink",
  "params": {
    "HubName_str": "Switchname",
    "OldAccountName_utf": "oldaccountname",
    "NewAccountName_utf": "newaccountname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "OldAccountName_utf": "oldaccountname",
    "NewAccountName_utf": "newaccountname"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
OldAccountName_utf	string (UTF8)	The old name of the cascade connection
NewAccountName_utf	string (UTF8)	The new name of the cascade connection

Get Current Cascade Connection Status

Description

Get Current Cascade Connection Status. When a Cascade Connection registered on the currently managed Virtual Switch is specified and that Cascade Connection is currently online, use this to get its connection status and other information. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetLinkStatus",
  "params": {
    "SwitchName_Ex_str": "Switchname_ex",
    "AccountName_utf": "accountname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "SwitchName_Ex_str": "Switchname_ex",
    "AccountName_utf": "accountname",
    "Active_qool": false,
    "Connected_qool": false,
    "SessionStatus_u32": 0,
    "ServerName_str": "servername",
    "ServerPort_u32": 0,
    "ServerProductName_str": "serverproductname",
    "ServerProductVer_u32": 0,
    "ServerProductBuild_u32": 0,
    "ServerX_bin": "SGVsbG8gV29ybGQ=",
    "ClientX_bin": "SGVsbG8gV29ybGQ=",
    "StartTime_dt": "2021-01-01T12:21:22.123",
    "FirstConnectionEstablisiedTime_dt": "2021-01-01T12:21:22.123",
    "CurrentConnectionEstablishTime_dt": "2021-01-01T12:21:22.123",
    "NumConnectionsEatablished_u32": 0,
    "HalfConnection_qool": false,
    "QoS_qool": false,
    "MaxTcpConnections_u32": 0,
    "NumTcpConnections_u32": 0,
    "NumTcpConnectionsUpload_u32": 0,
    "NumTcpConnectionsDownload_u32": 0,
    "UseEncrypt_qool": false,
    "CipherName_str": "ciphername",
    "UseCompress_qool": false,
    "IsRUDPSession_qool": false,
    "UnderlayProtocol_str": "underlayprotocol",
```

```
    "IsUdpAccelerationEnabled_qool": false,
    "IsUsingUdpAcceleration_qool": false,
    "SessionName_str": "sessionname",
    "ConnectionName_str": "connectionname",
    "SessionKey_bin": "SGVsbG8gV29ybGQ=",
    "TotalSendSize_u64": 0,
    "TotalRecvSize_u64": 0,
    "TotalSendSizeReal_u64": 0,
    "TotalRecvSizeReal_u64": 0,
    "IsBridgeMode_qool": false,
    "IsMonitorMode_qool": false,
    "VLanId_u32": 0
  }
}
```


Parameters

Name	Type	Description
SwitchName_Ex_str	string (ASCII)	The Virtual Switch name
AccountName_utf	string (UTF8)	The name of the cascade connection
Active_qool	qoolean	The flag whether the cascade connection is enabled
Connected_qool	qoolean	The flag whether the cascade connection is established
SessionStatus_u32	number (enum)	The session status Values: 0: Connecting 1: Negotiating 2: During user authentication 3: Connection complete 4: Wait to retry 5: Idle state
ServerName_str	string (ASCII)	The destination iQuila Server name
ServerPort_u32	number (uint32)	The port number of the server
ServerProductName_str	string (ASCII)	Server product name
ServerProductVer_u32	number (uint32)	Server product version
ServerProductBuild_u32	number (uint32)	Server product build number
ServerX_bin	string (Base64 binary)	Server's X.509 certificate
ClientX_bin	string (Base64 binary)	Client certificate
StartTime_dt	Date	Connection start time
FirstConnectionEstablisiedTime_dt	Date	Connection completion time of the first connection
CurrentConnectionEstablishTime_dt	Date	Connection completion time of this connection
NumConnectionsEatablished_u32	number (uint32)	Number of connections have been established so far
HalfConnection_qool	qoolean	Half-connection
QoS_qool	qoolean	VoIP / QoS
MaxTcpConnections_u32	number (uint32)	Maximum number of the underlying TCP connections
NumTcpConnections_u32	number (uint32)	Number of current underlying TCP connections
NumTcpConnectionsUpload_u32	number (uint32)	Number of underlying inbound TCP connections
NumTcpConnectionsDownload_u32	number (uint32)	Number of underlying outbound TCP connections
UseEncrypt_qool	qoolean	Use of encryption
CipherName_str	string (ASCII)	Cipher algorithm name
UseCompress_qool	qoolean	Use of compression
IsRUDPSession_qool	qoolean	The flag whether this is a R-UDP session
UnderlayProtocol_str	string (ASCII)	Underlying physical communication protocol
IsUdpAccelerationEnabled_qool	qoolean	The UDP acceleration is enabled
IsUsingUdpAcceleration_qool	qoolean	The UDP acceleration is being actually used
SessionName_str	string (ASCII)	Session name
ConnectionName_str	string (ASCII)	Connection name
SessionKey_bin	string (Base64 binary)	Session key
TotalSendSize_u64	number (uint64)	Total transmitted data size
TotalRecvSize_u64	number (uint64)	Total received data size
TotalSendSizeReal_u64	number (uint64)	Total transmitted data size (no compression)
TotalRecvSizeReal_u64	number (uint64)	Total received data size (no compression)
IsBridgeMode_qool	qoolean	The flag whether the VEN session is Bridge Mode
IsMonitorMode_qool	qoolean	The flag whether the VEN session is Monitor mode
VLanId_u32	number (uint32)	VLAN ID

Add Access List Rule

Description

Add Access List Rule. Use this to add a new rule to the access list of the currently managed Virtual Switch. The access list is a set of packet file rules that are applied to packets that flow through the Virtual Switch. You can register multiple rules in an access list and you can also define an priority for each rule. All packets are checked for the conditions specified by the rules registered in the access list and based on the operation that is stipulated by the first matching rule, they either pass or are discarded. Packets that do not match any rule are implicitly allowed to pass. You can also use the access list to generate delays, jitters and packet losses. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddAccess",
  "params": {
    "HubName_str": "Switchname",
    "AccessListSingle": [
      {
        "Id_u32": 0,
        "Note_utf": "note",
        "Active_qool": false,
        "Priority_u32": 0,
        "Discard_qool": false,
        "IsIPv6_qool": false,
        "SrcIpAddress_ip": "10.0.0.1",
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
        "DestPortEnd_u32": 0,
        "SrcUsername_str": "srcusername",
        "DestUsername_str": "destusername",
        "CheckSrcMac_qool": false,
        "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckDstMac_qool": false,
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
```

```
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      }
    ]
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AccessListSingle": [
      {
        "Id_u32": 0,
        "Note_utf": "note",
        "Active_qool": false,
        "Priority_u32": 0,
        "Discard_qool": false,
        "IsIPv6_qool": false,
        "SrcIpAddress_ip": "10.0.0.1",
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
        "DestPortEnd_u32": 0,
        "SrcUsername_str": "srcusername",
        "DestUsername_str": "destusername",
        "CheckSrcMac_qool": false,
        "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckDstMac_qool": false,
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
AccessListSingle	Array object	Access list (Must be a single item)
Id_u32	number (uint32)	ID
Note_utf	string (UTF8)	Specify a description (note) for this rule
Active_qool	qoolean	Enabled flag (true: enabled, false: disabled)
Priority_u32	number (uint32)	Specify an integer of 1 or higher to indicate the priority of the rule. Higher priority is given to rules with the lower priority values.
Discard_qool	qoolean	The flag if the rule is DISCARD operation or PASS operation. When a packet matches this rule condition, this operation is decided. When the operation of the rule is PASS, the packet is allowed to pass, otherwise the packet will be discarded.
IsIPv6_qool	qoolean	The flag if the rule is for IPv6. Specify false for IPv4, or specify true for IPv6.
SrcIpAddress_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a source IPv4 address as a rule condition. You must also specify the SrcSubnetMask_ip field.
SrcSubnetMask_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a source IPv4 subnet mask as a rule condition. "0.0.0.0" means all hosts. "255.255.255.255" means one single host.
DestIpAddress_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a destination IPv4 address as a rule condition. You must also specify the DestSubnetMask_ip field.

DestSubnetMask_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a destination IPv4 subnet mask as a rule condition. "0.0.0.0" means all hosts. "255.255.255.255" means one single host.
SrcIpAddress6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a source IPv6 address as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 address in binary form. You must also specify the SrcSubnetMask6_bin field.
SrcSubnetMask6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a source IPv6 subnet mask as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 subnet mask in binary form.
DestIpAddress6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a destination IPv6 address as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 address in binary form. You must also specify the DestSubnetMask6_bin field.
DestSubnetMask6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a destination IPv6 subnet mask as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 subnet mask in binary form.
Protocol_u32	number (enum)	The IP protocol number Values: 1: ICMP for IPv4 6: TCP 17: UDP 58: ICMP for IPv6

SrcPortStart_u32	number (uint32)	The Start Value of the Source Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the source port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
SrcPortEnd_u32	number (uint32)	The End Value of the Source Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the source port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
DestPortStart_u32	number (uint32)	The Start Value of the Destination Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the destination port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
DestPortEnd_u32	number (uint32)	The End Value of the Destination Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the destination port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
SrcUsername_str	string (ASCII)	Source user name. You can apply this rule to only the packets sent by a user session of a user name that has been specified as a rule condition. In this case, specify the user name.

DestUsername_str	string (ASCII)	Destination user name. You can apply this rule to only the packets received by a user session of a user name that has been specified as a rule condition. In this case, specify the user name.
CheckSrcMac_qool	qoolean	Specify true if you want to check the source MAC address.
SrcMacAddress_bin	string (Base64 binary)	Source MAC address (6 bytes), valid only if CheckSrcMac_qool == true.
SrcMacMask_bin	string (Base64 binary)	Source MAC address mask (6 bytes), valid only if CheckSrcMac_qool == true.
CheckDstMac_qool	qoolean	Specify true if you want to check the destination MAC address.
DstMacAddress_bin	string (Base64 binary)	Destination MAC address (6 bytes), valid only if CheckSrcMac_qool == true.
DstMacMask_bin	string (Base64 binary)	Destination MAC address mask (6 bytes), valid only if CheckSrcMac_qool == true.
CheckTcpState_qool	qoolean	Specify true if you want to check the state of the TCP connection.
Established_qool	qoolean	Valid only if CheckTcpState_qool == true. Set this field true to match only TCP-established packets. Set this field false to match only TCP-non established packets.
Delay_u32	number (uint32)	Set this value to generate delays when packets is passing. Specify the delay period in milliseconds. Specify 0 means no delays to generate. The delays must be 10000 milliseconds at most.
Jitter_u32	number (uint32)	Set this value to generate jitters when packets is passing. Specify the ratio of fluctuation of jitters within 0% to 100% range. Specify 0 means no jitters to generate.
Loss_u32	number (uint32)	Set this value to generate packet losses when packets is passing.

		Specify the ratio of packet losses within 0% to 100% range. Specify 0 means no packet losses to generate.
RedirectUrl_str	string (ASCII)	The specified URL will be mandatory replied to the client as a response for TCP connecting request packets which matches the conditions of this access list entry via this Virtual Switch. To use this setting, you can enforce the web browser of the iQuila Client server to show the specified web site when that web browser tries to access the specific IP address.

Delete Rule from Access List

Description

Delete Rule from Access List. Use this to specify a packet filter rule registered on the access list of the currently managed Virtual Switch and delete it. To delete a rule, you must specify that rule's ID. You can display the ID by using the EnumAccess API. If you wish not to delete the rule but to only temporarily disable it, use the SetAccessList API to set the rule status to disable. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteAccess",
  "params": {
    "HubName_str": "Switchname",
    "Id_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Id_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Id_u32	number (uint32)	ID

Get Access List Rule List

Description

Get Access List Rule List. Use this to get a list of packet filter rules that are registered on access list of the currently managed Virtual Switch. The access list is a set of packet file rules that are applied to packets that flow through the Virtual Switch. You can register multiple rules in an access list and you can also define a priority for each rule. All packets are checked for the conditions specified by the rules registered in the access list and based on the operation that is stipulated by the first matching rule, they either pass or are discarded. Packets that do not match any rule are implicitly allowed to pass. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumAccess",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AccessList": [
      {
        "Id_u32": 0,
        "Note_utf": "note",
        "Active_qool": false,
        "Priority_u32": 0,
        "Discard_qool": false,
        "IsIPv6_qool": false,
        "SrcIpAddress_ip": "10.0.0.1",
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
```

```
      "DestPortEnd_u32": 0,
      "SrcUsername_str": "srcusername",
      "DestUsername_str": "destusername",
      "CheckSrcMac_qool": false,
      "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
      "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
      "CheckDstMac_qool": false,
      "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
      "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
      "CheckTcpState_qool": false,
      "Established_qool": false,
      "Delay_u32": 0,
      "Jitter_u32": 0,
      "Loss_u32": 0,
      "RedirectUrl_str": "redirecturl"
    },
    {
      "Id_u32": 0,
      "Note_utf": "note",
      "Active_qool": false,
      "Priority_u32": 0,
      "Discard_qool": false,
      "IsIPv6_qool": false,
      "SrcIpAddress_ip": "10.0.0.1",
      "SrcSubnetMask_ip": "255.255.255.255",
      "DestIpAddress_ip": "10.0.0.1",
      "DestSubnetMask_ip": "255.255.255.255",
      "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
      "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
      "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
      "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
      "Protocol_u32": 0,
      "SrcPortStart_u32": 0,
      "SrcPortEnd_u32": 0,
      "DestPortStart_u32": 0,
      "DestPortEnd_u32": 0,
      "SrcUsername_str": "srcusername",
      "DestUsername_str": "destusername",
      "CheckSrcMac_qool": false,
      "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
      "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
      "CheckDstMac_qool": false,
      "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
      "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
      "CheckTcpState_qool": false,
      "Established_qool": false,
      "Delay_u32": 0,
      "Jitter_u32": 0,
      "Loss_u32": 0,
      "RedirectUrl_str": "redirecturl"
    },
    {
      "Id_u32": 0,
      "Note_utf": "note",
      "Active_qool": false,
      "Priority_u32": 0,
      "Discard_qool": false,
      "IsIPv6_qool": false,
      "SrcIpAddress_ip": "10.0.0.1",
```

```
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
        "DestPortEnd_u32": 0,
        "SrcUsername_str": "srcusername",
        "DestUsername_str": "destusername",
        "CheckSrcMac_qool": false,
        "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckDstMac_qool": false,
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
AccessList	Array object	Access list
Id_u32	number (uint32)	ID
Note_utf	string (UTF8)	Specify a description (note) for this rule
Active_qool	qoolean	Enabled flag (true: enabled, false: disabled)
Priority_u32	number (uint32)	Specify an integer of 1 or higher to indicate the priority of the rule. Higher priority is given to rules with the lower priority values.
Discard_qool	qoolean	The flag if the rule is DISCARD operation or PASS operation. When a packet matches this rule condition, this operation is decided. When the operation of the rule is PASS, the packet is allowed to pass, otherwise the packet will be discarded.
IsIPv6_qool	qoolean	The flag if the rule is for IPv6. Specify false for IPv4, or specify true for IPv6.
SrcIpAddress_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a source IPv4 address as a rule condition. You must also specify the SrcSubnetMask_ip field.
SrcSubnetMask_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a source IPv4 subnet mask as a rule condition. "0.0.0.0" means all hosts. "255.255.255.255" means one single host.
DestIpAddress_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a destination IPv4 address as a rule condition. You must also specify the DestSubnetMask_ip field.
DestSubnetMask_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a destination IPv4 subnet mask as a rule condition. "0.0.0.0" means all hosts. "255.255.255.255" means one single host.
SrcIpAddress6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a source IPv6 address as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 address in binary form. You must also specify the SrcSubnetMask6_bin field.
SrcSubnetMask6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a source IPv6 subnet mask as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 subnet mask in binary form.

DestIpAddress6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a destination IPv6 address as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 address in binary form. You must also specify the DestSubnetMask6_bin field.
DestSubnetMask6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a destination IPv6 subnet mask as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 subnet mask in binary form.
Protocol_u32	number (enum)	The IP protocol number Values: 1: ICMP for IPv4 6: TCP 17: UDP 58: ICMP for IPv6
SrcPortStart_u32	number (uint32)	The Start Value of the Source Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the source port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
SrcPortEnd_u32	number (uint32)	The End Value of the Source Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the source port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
DestPortStart_u32	number (uint32)	The Start Value of the Destination Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the destination port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
DestPortEnd_u32	number (uint32)	The End Value of the Destination Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the destination port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
SrcUsername_str	string (ASCII)	Source user name. You can apply this rule to only the packets sent by a user session of a user name that has been specified as a rule condition. In this case, specify the user name.
DestUsername_str	string (ASCII)	Destination user name. You can apply this rule to only the packets received by a user session of a user name that has been specified as a rule condition. In this case, specify the user name.
CheckSrcMac_qool	qoolean	Specify true if you want to check the source MAC address.
SrcMacAddress_bin	string (Base64 binary)	Source MAC address (6 bytes), valid only if CheckSrcMac_qool == true.
SrcMacMask_bin	string (Base64 binary)	Source MAC address mask (6 bytes), valid only if CheckSrcMac_qool == true.
CheckDstMac_qool	qoolean	Specify true if you want to check the destination MAC address.
DstMacAddress_bin	string (Base64 binary)	Destination MAC address (6 bytes), valid only if CheckSrcMac_qool == true.
DstMacMask_bin	string (Base64 binary)	Destination MAC address mask (6 bytes), valid only if CheckSrcMac_qool == true.
CheckTcpState_qool	qoolean	Specify true if you want to check the state of the TCP connection.
Established_qool	qoolean	Valid only if CheckTcpState_qool == true. Set this field true to match only TCP-established packets. Set this field false to match only TCP-non established packets.
Delay_u32	number (uint32)	Set this value to generate delays when packets is passing. Specify the delay period in milliseconds. Specify 0 means no delays to generate. The delays must be 10000 milliseconds at most.
Jitter_u32	number (uint32)	Set this value to generate jitters when packets is passing. Specify the ratio of fluctuation of jitters within 0% to 100% range. Specify 0 means no jitters to generate.
Loss_u32	number (uint32)	Set this value to generate packet losses when packets is passing. Specify the ratio of packet losses within 0% to 100% range. Specify 0 means no packet losses to generate.
RedirectUrl_str	string (ASCII)	The specified URL will be mandatory replied to the client as a response for TCP connecting request packets which matches the conditions of this access list entry via this Virtual Switch. To use this setting, you can enforce the web browser of the iQuila Client server to show the specified web site when that web browser tries to access the specific IP address.

Replace all access lists on a single bulk API call

Description

Replace all access lists on a single bulk API call. This API removes all existing access list rules on the Virtual Switch, and replace them by new access list rules specified by the parameter.

Input Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetAccessList",
  "params": {
    "HubName_str": "Switchname",
    "AccessList": [
      {
        "Id_u32": 0,
        "Note_utf": "note",
        "Active_qool": false,
        "Priority_u32": 0,
        "Discard_qool": false,
        "IsIPv6_qool": false,
        "SrcIpAddress_ip": "10.0.0.1",
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
        "DestPortEnd_u32": 0,
        "SrcUsername_str": "srcusername",
        "DestUsername_str": "destusername",
        "CheckSrcMac_qool": false,
        "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckDstMac_qool": false,
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      },
      {
        "Id_u32": 0,
```

```
      "Note_utf": "note",
      "Active_qool": false,
      "Priority_u32": 0,
      "Discard_qool": false,
      "IsIPv6_qool": false,
      "SrcIpAddress_ip": "10.0.0.1",
      "SrcSubnetMask_ip": "255.255.255.255",
      "DestIpAddress_ip": "10.0.0.1",
      "DestSubnetMask_ip": "255.255.255.255",
      "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
      "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
      "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
      "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
      "Protocol_u32": 0,
      "SrcPortStart_u32": 0,
      "SrcPortEnd_u32": 0,
      "DestPortStart_u32": 0,
      "DestPortEnd_u32": 0,
      "SrcUsername_str": "srcusername",
      "DestUsername_str": "destusername",
      "CheckSrcMac_qool": false,
      "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
      "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
      "CheckDstMac_qool": false,
      "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
      "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
      "CheckTcpState_qool": false,
      "Established_qool": false,
      "Delay_u32": 0,
      "Jitter_u32": 0,
      "Loss_u32": 0,
      "RedirectUrl_str": "redirecturl"
    },
    {
      "Id_u32": 0,
      "Note_utf": "note",
      "Active_qool": false,
      "Priority_u32": 0,
      "Discard_qool": false,
      "IsIPv6_qool": false,
      "SrcIpAddress_ip": "10.0.0.1",
      "SrcSubnetMask_ip": "255.255.255.255",
      "DestIpAddress_ip": "10.0.0.1",
      "DestSubnetMask_ip": "255.255.255.255",
      "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
      "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
      "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
      "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
      "Protocol_u32": 0,
      "SrcPortStart_u32": 0,
      "SrcPortEnd_u32": 0,
      "DestPortStart_u32": 0,
      "DestPortEnd_u32": 0,
      "SrcUsername_str": "srcusername",
      "DestUsername_str": "destusername",
      "CheckSrcMac_qool": false,
      "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
      "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
      "CheckDstMac_qool": false,
```

```
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      }
    ]
  }
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AccessList": [
      {
        "Id_u32": 0,
        "Note_utf": "note",
        "Active_qool": false,
        "Priority_u32": 0,
        "Discard_qool": false,
        "IsIPv6_qool": false,
        "SrcIpAddress_ip": "10.0.0.1",
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
        "DestPortEnd_u32": 0,
        "SrcUsername_str": "srcusername",
        "DestUsername_str": "destusername",
        "CheckSrcMac_qool": false,
        "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckDstMac_qool": false,
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      },
      {
        "Id_u32": 0,
        "Note_utf": "note",
        "Active_qool": false,
        "Priority_u32": 0,
        "Discard_qool": false,
        "IsIPv6_qool": false,
        "SrcIpAddress_ip": "10.0.0.1",
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
```

```
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
        "DestPortEnd_u32": 0,
        "SrcUsername_str": "srcusername",
        "DestUsername_str": "destusername",
        "CheckSrcMac_qool": false,
        "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckDstMac_qool": false,
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      },
      {
        "Id_u32": 0,
        "Note_utf": "note",
        "Active_qool": false,
        "Priority_u32": 0,
        "Discard_qool": false,
        "IsIPv6_qool": false,
        "SrcIpAddress_ip": "10.0.0.1",
        "SrcSubnetMask_ip": "255.255.255.255",
        "DestIpAddress_ip": "10.0.0.1",
        "DestSubnetMask_ip": "255.255.255.255",
        "SrcIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "SrcSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "DestIpAddress6_bin": "SGVsbG8gV29ybGQ=",
        "DestSubnetMask6_bin": "SGVsbG8gV29ybGQ=",
        "Protocol_u32": 0,
        "SrcPortStart_u32": 0,
        "SrcPortEnd_u32": 0,
        "DestPortStart_u32": 0,
        "DestPortEnd_u32": 0,
        "SrcUsername_str": "srcusername",
        "DestUsername_str": "destusername",
        "CheckSrcMac_qool": false,
        "SrcMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "SrcMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckDstMac_qool": false,
        "DstMacAddress_bin": "SGVsbG8gV29ybGQ=",
        "DstMacMask_bin": "SGVsbG8gV29ybGQ=",
        "CheckTcpState_qool": false,
        "Established_qool": false,
        "Delay_u32": 0,
        "Jitter_u32": 0,
        "Loss_u32": 0,
        "RedirectUrl_str": "redirecturl"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
AccessList	Array object	Access list
Id_u32	number (uint32)	ID
Note_utf	string (UTF8)	Specify a description (note) for this rule
Active_qool	qoolean	Enabled flag (true: enabled, false: disabled)
Priority_u32	number (uint32)	Specify an integer of 1 or higher to indicate the priority of the rule. Higher priority is given to rules with the lower priority values.
Discard_qool	qoolean	The flag if the rule is DISCARD operation or PASS operation. When a packet matches this rule condition, this operation is decided. When the operation of the rule is PASS, the packet is allowed to pass, otherwise the packet will be discarded.
IsIPv6_qool	qoolean	The flag if the rule is for IPv6. Specify false for IPv4, or specify true for IPv6.
SrcIpAddress_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a source IPv4 address as a rule condition. You must also specify the SrcSubnetMask_ip field.
SrcSubnetMask_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a source IPv4 subnet mask as a rule condition. "0.0.0.0" means all hosts. "255.255.255.255" means one single host.
DestIpAddress_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a destination IPv4 address as a rule condition. You must also specify the DestSubnetMask_ip field.
DestSubnetMask_ip	string (IP address)	Valid only if the rule is IPv4 mode (IsIPv6_qool == false). Specify a destination IPv4 subnet mask as a rule condition. "0.0.0.0" means all hosts. "255.255.255.255" means one single host.
SrcIpAddress6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a source IPv6 address as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 address in binary form. You must also specify the SrcSubnetMask6_bin field.
SrcSubnetMask6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a source IPv6 subnet mask as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 subnet mask in binary form.
DestIpAddress6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a destination IPv6 address as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 address in binary form. You must also specify the DestSubnetMask6_bin field.
DestSubnetMask6_bin	string (Base64 binary)	Valid only if the rule is IPv6 mode (IsIPv6_qool == true). Specify a destination IPv6 subnet mask as a rule condition. The field must be a byte array of 16 bytes (128 bits) to contain the IPv6 subnet mask in binary form.
Protocol_u32	number (enum)	The IP protocol number Values: 1: ICMP for IPv4 6: TCP 17: UDP 58: ICMP for IPv6
SrcPortStart_u32	number (uint32)	The Start Value of the Source Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the source port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
SrcPortEnd_u32	number (uint32)	The End Value of the Source Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the source port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
DestPortStart_u32	number (uint32)	The Start Value of the Destination Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the destination port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
DestPortEnd_u32	number (uint32)	The End Value of the Destination Port Number Range. If the specified protocol is TCP/IP or UDP/IP, specify the destination port number as the rule condition. Protocols other than this will be ignored. When this parameter is not specified, the rules will apply to all port numbers.
SrcUsername_str	string (ASCII)	Source user name. You can apply this rule to only the packets sent by a user session of a user name that has been specified as a rule condition. In this case, specify the user name.
DestUsername_str	string (ASCII)	Destination user name. You can apply this rule to only the packets received by a user session of a user name that has been specified as a rule condition. In this case, specify the user name.
CheckSrcMac_qool	qoolean	Specify true if you want to check the source MAC address.

SrcMacAddress_bin	string (Base64 binary)	Source MAC address (6 bytes), valid only if CheckSrcMac_qool == true.
SrcMacMask_bin	string (Base64 binary)	Source MAC address mask (6 bytes), valid only if CheckSrcMac_qool == true.
CheckDstMac_qool	qoolean	Specify true if you want to check the destination MAC address.
DstMacAddress_bin	string (Base64 binary)	Destination MAC address (6 bytes), valid only if CheckSrcMac_qool == true.
DstMacMask_bin	string (Base64 binary)	Destination MAC address mask (6 bytes), valid only if CheckSrcMac_qool == true.
CheckTcpState_qool	qoolean	Specify true if you want to check the state of the TCP connection.
Established_qool	qoolean	Valid only if CheckTcpState_qool == true. Set this field true to match only TCP-established packets. Set this field false to match only TCP-non established packets.
Delay_u32	number (uint32)	Set this value to generate delays when packets is passing. Specify the delay period in milliseconds. Specify 0 means no delays to generate. The delays must be 10000 milliseconds at most.
Jitter_u32	number (uint32)	Set this value to generate jitters when packets is passing. Specify the ratio of fluctuation of jitters within 0% to 100% range. Specify 0 means no jitters to generate.
Loss_u32	number (uint32)	Set this value to generate packet losses when packets is passing. Specify the ratio of packet losses within 0% to 100% range. Specify 0 means no packet losses to generate.
RedirectUrl_str	string (ASCII)	The specified URL will be mandatory replied to the client as a response for TCP connecting request packets which matches the conditions of this access list entry via this Virtual Switch. To use this setting, you can enforce the web browser of the iQuila Client server to show the specified web site when that web browser tries to access the specific IP address.

Create a user

Description

Create a user. Use this to create a new user in the security account database of the currently managed Virtual Switch. By creating a user, the iQuila Client can connect to the Virtual Switch by using the authentication information of that user. Note that a user whose user name has been created as "" *(a single asterisk character) will automatically be registered as a RADIUS authentication user. For cases where there are users with* "" as the name, when a user, whose user name that has been provided when a client connected to a iQuila Server does not match existing user names, is able to be authenticated by a RADIUS server or AD domain controller by inputting a user name and password, the authentication settings and security policy settings will follow the setting for the user "*". To change the user information of a user that has been created, use the SetUser API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "CreateUser",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "ExpireTime_dt": "2021-01-01T12:21:22.123",
    "AuthType_u32": 0,
    "Auth_Password_str": "auth_password",
    "UserX_bin": "SGVsbG8gV29ybGQ=",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "CommonName_utf": "auth_rootcert_commonname",
    "RadiusUsername_utf": "auth_radius_radiususername",
    "NtUsername_utf": "auth_nt_ntusername",
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
```

```
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "GroupName_str": "groupname",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "CreatedTime_dt": "2021-01-01T12:21:22.123",
    "UpdatedTime_dt": "2021-01-01T12:21:22.123",
    "ExpireTime_dt": "2021-01-01T12:21:22.123",
    "AuthType_u32": 0,
    "Auth_Password_str": "auth_password",
    "UserX_bin": "SGVsbG8gV29ybGQ=",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "CommonName_utf": "auth_rootcert_commonname",
    "RadiusUsername_utf": "auth_radius_radiususername",
    "NtUsername_utf": "auth_nt_ntusername",
    "NumLogin_u32": 0,
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "UsePolicy_qool": false,
```

```
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	Specify the user name of the user
GroupName_str	string (ASCII)	Assigned group name for the user
Realname_utf	string (UTF8)	Optional real name (full name) of the user, allow using any Unicode characters
Note_utf	string (UTF8)	Optional User Description
CreatedTime_dt	Date	Creation date and time
UpdatedTime_dt	Date	Last modified date and time
ExpireTime_dt	Date	Expiration date and time
AuthType_u32	number (enum)	Authentication method of the user Values: 0: Anonymous authentication 1: Password authentication 2: User certificate authentication 3: Root certificate which is issued by trusted Certificate Authority

		4: Radius authentication 5: Windows NT authentication
Auth_Password_str	string (ASCII)	User password, valid only if AuthType_u32 == Password(1). Valid only to create or set operations.
UserX_bin	string (Base64 binary)	User certificate, valid only if AuthType_u32 == UserCert(2).
Serial_bin	string (Base64 binary)	Certificate Serial Number, optional, valid only if AuthType_u32 == RootCert(3).
CommonName_utf	string (UTF8)	Certificate Common Name, optional, valid only if AuthType_u32 == RootCert(3).
RadiusUsername_utf	string (UTF8)	Username in RADIUS server, optional, valid only if AuthType_u32 == Radius(4).
NtUsername_utf	string (UTF8)	Username in NT Domain server, optional, valid only if AuthType_u32 == NT(5).
NumLogin_u32	number (uint32)	Number of total logins of the user
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
UsePolicy_qool	qoolean	The flag whether to use security policy
policy:Access_qool	qoolean	Security policy: Allow Access. The users, which this policy value is true, have permission to make VEN connection to iQuila Server.
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
policy:NoBridge_qool	qoolean	Security policy: Deny Bridge Operation. Bridge-mode connections are denied for user sessions that have this policy setting. Even in cases when the Ethernet Bridge is configured in the client side, communication will not be possible.
policy:NoRouting_qool	qoolean	Security policy: Deny Routing Operation (IPv4). IPv4 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckMac_qool	qoolean	Security policy: Deny MAC Addresses Duplication. The use of duplicating MAC addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:CheckIP_qool	qoolean	Security policy: Deny IP Address Duplication (IPv4). The use of duplicating IPv4 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.
policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.

policy:MonitorPort_qool	qoolean	Security policy: Allow Monitoring Mode. Users with this policy setting will be granted to connect to the Virtual Switch in Monitoring Mode. Sessions in Monitoring Mode are able to monitor (tap) all packets flowing through the Virtual Switch.
policy:MaxConnection_u32	number (uint32)	Security policy: Maximum Number of TCP Connections. For sessions with this policy setting, this sets the maximum number of physical TCP connections consists in a physical VEN session.
policy:TimeOut_u32	number (uint32)	Security policy: Time-out Period. For sessions with this policy setting, this sets, in seconds, the time-out period to wait before disconnecting a session when communication trouble occurs between the iQuila Client / iQuila Server.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:FixPassword_qool	qoolean	Security policy: Deny Changing Password. The users which use password authentication with this policy setting are not allowed to change their own password from the iQuila Client Manager or similar.
policy:MultiLogins_u32	number (uint32)	Security policy: Maximum Number of Multiple Logins. Users with this policy setting are unable to have more than this number of concurrent logins. Bridge Mode sessions are not subjects to this policy.
policy:NoQoS_qool	qoolean	Security policy: Deny VoIP / QoS Function. Users with this security policy are unable to use VoIP / QoS functions in VEN connection sessions.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.
policy:RAFilter_qool	qoolean	Security policy: Filter RA Packets (IPv6). All ICMPv6 packets which the message-type is 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, a malicious users will be unable to spread illegal IPv6 prefix or default gateway advertisements on the network.
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
policy:NoRoutingV6_qool	qoolean	Security policy: Deny Routing Operation (IPv6). IPv6 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckIPv6_qool	qoolean	Security policy: Deny IP Address Duplication (IPv6). The use of duplicating IPv6 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:NoSavePassword_qool	qoolean	Security policy: Disallow Password Save in iQuila Client. For users with this policy setting, when the user is using *standard* password authentication, the user will be

		unable to save the password in iQuila Client. The user will be required to input passwords for every time to connect a VEN. This will improve the security. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:AutoDisconnect_u32	number (uint32)	Security policy: iQuila Client Automatic Disconnect. For users with this policy setting, a user's VEN session will be disconnected automatically after the specific period will elapse. In this case no automatic re-connection will be performed. This can prevent a lot of inactive VEN Sessions. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:NoIPv6DefaultRouterInRAWhenIPv6_qool	qoolean	Security policy: No Default-Router on IPv6 RA (physical IPv6). In all VEN Sessions defines this policy (only when the physical communication protocol between iQuila Client / iQuila Bridge and iQuila Server is IPv6), any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Change User Settings

Description

Change User Settings. Use this to change user settings that is registered on the security account database of the currently managed Virtual Switch. The user settings that can be changed using this API are the three items that are specified when a new user is created using the CreateUser API: Group Name, Full Name, and Description. To get the list of currently registered users, use the EnumUser API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetUser",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "GroupName_str": "groupname",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "ExpireTime_dt": "2021-01-01T12:21:22.123",
    "AuthType_u32": 0,
    "Auth_Password_str": "auth_password",
    "UserX_bin": "SGVsbG8gV29ybGQ=",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "CommonName_utf": "auth_rootcert_commonname",
    "RadiusUsername_utf": "auth_radius_radiususername",
    "NtUsername_utf": "auth_nt_ntusername",
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
```

```
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "GroupName_str": "groupname",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "CreatedTime_dt": "2021-01-01T12:21:22.123",
    "UpdatedTime_dt": "2021-01-01T12:21:22.123",
    "ExpireTime_dt": "2021-01-01T12:21:22.123",
    "AuthType_u32": 0,
    "Auth_Password_str": "auth_password",
    "UserX_bin": "SGVsbG8gV29ybGQ=",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "CommonName_utf": "auth_rootcert_commonname",
    "RadiusUsername_utf": "auth_radius_radiususername",
    "NtUsername_utf": "auth_nt_ntusername",
    "NumLogin_u32": 0,
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
```

```
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	Specify the user name of the user
GroupName_str	string (ASCII)	Assigned group name for the user
Realname_utf	string (UTF8)	Optional real name (full name) of the user, allow using any Unicode characters
Note_utf	string (UTF8)	Optional User Description
CreatedTime_dt	Date	Creation date and time
UpdatedTime_dt	Date	Last modified date and time
ExpireTime_dt	Date	Expiration date and time
AuthType_u32	number (enum)	Authentication method of the user Values: 0: Anonymous authentication 1: Password authentication 2: User certificate authentication 3: Root certificate which is issued by trusted Certificate Authority 4: Radius authentication 5: Windows NT authentication
Auth_Password_str	string (ASCII)	User password, valid only if AuthType_u32 == Password(1). Valid only to create or set operations.
UserX_bin	string (Base64 binary)	User certificate, valid only if AuthType_u32 == UserCert(2).

Serial_bin	string (Base64 binary)	Certificate Serial Number, optional, valid only if AuthType_u32 == RootCert(3).
CommonName_utf	string (UTF8)	Certificate Common Name, optional, valid only if AuthType_u32 == RootCert(3).
RadiusUsername_utf	string (UTF8)	Username in RADIUS server, optional, valid only if AuthType_u32 == Radius(4).
NtUsername_utf	string (UTF8)	Username in NT Domain server, optional, valid only if AuthType_u32 == NT(5).
NumLogin_u32	number (uint32)	Number of total logins of the user
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
UsePolicy_qool	qoolean	The flag whether to use security policy
policy:Access_qool	qoolean	Security policy: Allow Access. The users, which this policy value is true, have permission to make VEN connection to iQuila Server.
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
policy:NoBridge_qool	qoolean	Security policy: Deny Bridge Operation. Bridge-mode connections are denied for user sessions that have this policy setting. Even in cases when the Ethernet Bridge is configured in the client side, communication will not be possible.
policy:NoRouting_qool	qoolean	Security policy: Deny Routing Operation (IPv4). IPv4 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckMac_qool	qoolean	Security policy: Deny MAC Addresses Duplication. The use of duplicating MAC addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:CheckIP_qool	qoolean	Security policy: Deny IP Address Duplication (IPv4). The use of duplicating IPv4 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.
policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MonitorPort_qool	qoolean	Security policy: Allow Monitoring Mode. Users with this policy setting will be granted to connect to the Virtual Switch in Monitoring Mode. Sessions in Monitoring Mode are able to monitor (tap) all packets flowing through the Virtual Switch.
policy:MaxConnection_u32	number (uint32)	Security policy: Maximum Number of TCP Connections. For sessions with this policy setting, this sets the maximum

		number of physical TCP connections consists in a physical VEN session.
policy:TimeOut_u32	number (uint32)	Security policy: Time-out Period. For sessions with this policy setting, this sets, in seconds, the time-out period to wait before disconnecting a session when communication trouble occurs between the iQuila Client / iQuila Server.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:FixPassword_qool	qoolean	Security policy: Deny Changing Password. The users which use password authentication with this policy setting are not allowed to change their own password from the iQuila Client Manager or similar.
policy:MultiLogins_u32	number (uint32)	Security policy: Maximum Number of Multiple Logins. Users with this policy setting are unable to have more than this number of concurrent logins. Bridge Mode sessions are not subjects to this policy.
policy:NoQoS_qool	qoolean	Security policy: Deny VoIP / QoS Function. Users with this security policy are unable to use VoIP / QoS functions in VEN connection sessions.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.
policy:RAFilter_qool	qoolean	Security policy: Filter RA Packets (IPv6). All ICMPv6 packets which the message-type is 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, a malicious users will be unable to spread illegal IPv6 prefix or default gateway advertisements on the network.
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
policy:NoRoutingV6_qool	qoolean	Security policy: Deny Routing Operation (IPv6). IPv6 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckIPv6_qool	qoolean	Security policy: Deny IP Address Duplication (IPv6). The use of duplicating IPv6 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:NoSavePassword_qool	qoolean	Security policy: Disallow Password Save in iQuila Client. For users with this policy setting, when the user is using *standard* password authentication, the user will be unable to save the password in iQuila Client. The user will be required to input passwords for every time to connect a VEN. This will improve the security. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:AutoDisconnect_u32	number (uint32)	Security policy: iQuila Client Automatic Disconnect. For users with this policy setting, a user's VEN session will be

		disconnected automatically after the specific period will elapse. In this case no automatic re-connection will be performed. This can prevent a lot of inactive VEN Sessions. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:NoIPv6DefaultRouterInRAWhenIPv6_qool	qoolean	Security policy: No Default-Router on IPv6 RA (physical IPv6). In all VEN Sessions defines this policy (only when the physical communication protocol between iQuila Client / iQuila Bridge and iQuila Server is IPv6), any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Get User Settings

Description

Get User Settings. Use this to get user settings information that is registered on the security account database of the currently managed Virtual Switch. The information that you can get using this API are User Name, Full Name, Group Name, Expiration Date, Security Policy, and Auth Type, as well as parameters that are specified as auth type attributes and the statistical data of that user. To get the list of currently registered users, use the EnumUser API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetUser",
  "params": {
    "HubName_str": "Switchname",
    "Name_st": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "GroupName_str": "groupname",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "CreatedTime_dt": "2021-01-01T12:21:22.123",
    "UpdatedTime_dt": "2021-01-01T12:21:22.123",
    "ExpireTime_dt": "2021-01-01T12:21:22.123",
    "AuthType_u32": 0,
    "Auth_Password_str": "auth_password",
    "UserX_bin": "SGVsbG8gV29ybGQ=",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "CommonName_utf": "auth_rootcert_commonname",
    "RadiusUsername_utf": "auth_radius_radiususername",
    "NtUsername_utf": "auth_nt_ntusername",
    "NumLogin_u32": 0,
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
```

```
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:HostnameFilter_str": "hostname,localdomain",
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	Specify the user name of the user
GroupName_str	string (ASCII)	Assigned group name for the user
Realname_utf	string (UTF8)	Optional real name (full name) of the user, allow using any Unicode characters
Note_utf	string (UTF8)	Optional User Description
CreatedTime_dt	Date	Creation date and time
UpdatedTime_dt	Date	Last modified date and time
ExpireTime_dt	Date	Expiration date and time
AuthType_u32	number (enum)	Authentication method of the user Values: 0: Anonymous authentication 1: Password authentication 2: User certificate authentication 3: Root certificate which is issued by trusted Certificate Authority 4: Radius authentication 5: Windows NT authentication
Auth_Password_str	string (ASCII)	User password, valid only if AuthType_u32 == Password(1). Valid only to create or set operations.
UserX_bin	string (Base64 binary)	User certificate, valid only if AuthType_u32 == UserCert(2).
Serial_bin	string (Base64 binary)	Certificate Serial Number, optional, valid only if AuthType_u32 == RootCert(3).
CommonName_utf	string (UTF8)	Certificate Common Name, optional, valid only if AuthType_u32 == RootCert(3).
RadiusUsername_utf	string (UTF8)	Username in RADIUS server, optional, valid only if AuthType_u32 == Radius(4).
NtUsername_utf	string (UTF8)	Username in NT Domain server, optional, valid only if AuthType_u32 == NT(5).
NumLogin_u32	number (uint32)	Number of total logins of the user
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
UsePolicy_qool	qoolean	The flag whether to use security policy
policy:Access_qool	qoolean	Security policy: Allow Access. The users, which this policy value is true, have permission to make VEN connection to iQuila Server.
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
policy:NoBridge_qool	qoolean	Security policy: Deny Bridge Operation. Bridge-mode connections are denied for user sessions that have this policy setting. Even in cases when the Ethernet Bridge is configured in the client side, communication will not be possible.
policy:NoRouting_qool	qoolean	Security policy: Deny Routing Operation (IPv4). IPv4 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.

policy:CheckMac_qool	qoolean	Security policy: Deny MAC Addresses Duplication. The use of duplicating MAC addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:CheckIP_qool	qoolean	Security policy: Deny IP Address Duplication (IPv4). The use of duplicating IPv4 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.
policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MonitorPort_qool	qoolean	Security policy: Allow Monitoring Mode. Users with this policy setting will be granted to connect to the Virtual Switch in Monitoring Mode. Sessions in Monitoring Mode are able to monitor (tap) all packets flowing through the Virtual Switch.
policy:MaxConnection_u32	number (uint32)	Security policy: Maximum Number of TCP Connections. For sessions with this policy setting, this sets the maximum number of physical TCP connections consists in a physical VEN session.
policy:TimeOut_u32	number (uint32)	Security policy: Time-out Period. For sessions with this policy setting, this sets, in seconds, the time-out period to wait before disconnecting a session when communication trouble occurs between the iQuila Client / iQuila Server.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:FixPassword_qool	qoolean	Security policy: Deny Changing Password. The users which use password authentication with this policy setting are not allowed to change their own password from the iQuila Client Manager or similar.
policy:MultiLogins_u32	number (uint32)	Security policy: Maximum Number of Multiple Logins. Users with this policy setting are unable to have more than this number of concurrent logins. Bridge Mode sessions are not subjects to this policy.
policy:NoQoS_qool	qoolean	Security policy: Deny VoIP / QoS Function. Users with this security policy are unable to use VoIP / QoS functions in VEN connection sessions.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.

policy:RAFilter_qool	qoolean	Security policy: Filter RA Packets (IPv6). All ICMPv6 packets which the message-type is 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, a malicious users will be unable to spread illegal IPv6 prefix or default gateway advertisements on the network.
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
policy:NoRoutingV6_qool	qoolean	Security policy: Deny Routing Operation (IPv6). IPv6 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckIPv6_qool	qoolean	Security policy: Deny IP Address Duplication (IPv6). The use of duplicating IPv6 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:NoSavePassword_qool	qoolean	Security policy: Disallow Password Save in iQuila Client. For users with this policy setting, when the user is using *standard* password authentication, the user will be unable to save the password in iQuila Client. The user will be required to input passwords for every time to connect a VEN. This will improve the security. If this policy is enabled, iQuila Client Version 4.0 will be denied to access.
policy:HostnameFilter_str	Hostname, localdomain	Security policy: this policy checks the hostname matches the entry, you may enter a wildcard when locaing in to a corporate domain eg. *domain.local
policy:AutoDisconnect_u32	number (uint32)	Security policy: iQuila Client Automatic Disconnect. For users with this policy setting, a user's VEN session will be disconnected automatically after the specific period will elapse. In this case no automatic re-connection will be performed. This can prevent a lot of inactive VEN Sessions. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:NoIPv6DefaultRouterInRAWhenIPv6_qool	qoolean	Security policy: No Default-Router on IPv6 RA (physical IPv6). In all VEN Sessions defines this policy (only when the physical communication protocol between iQuila Client / iQuila Bridge and iQuila Server is IPv6), any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6

		routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Delete a user

Description

Delete a user. Use this to delete a user that is registered on the security account database of the currently managed Virtual Switch. By deleting the user, that user will no long be able to connect to the Virtual Switch. You can use the SetUser API to set the user's security policy to deny access instead of deleting a user, set the user to be temporarily denied from logging in. To get the list of currently registered users, use the EnumUser API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteUser",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	User or group name

Get List of Users

Description

Get List of Users. Use this to get a list of users that are registered on the security account database of the currently managed Virtual Switch. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumUser",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "UserList": [
      {
        "Name_str": "name",
        "GroupName_str": "groupname",
        "Realname_utf": "realname",
        "Note_utf": "note",
        "AuthType_u32": 0,
        "NumLogin_u32": 0,
        "LastLoginTime_dt": "2021-01-01T12:21:22.123",
        "DenyAccess_qool": false,
        "IsTrafficFilled_qool": false,
        "IsExpiresFilled_qool": false,
        "Expires_dt": "2021-01-01T12:21:22.123",
        "Ex.Recv.BroadcastBytes_u64": 0,
        "Ex.Recv.BroadcastCount_u64": 0,
        "Ex.Recv.UnicastBytes_u64": 0,
        "Ex.Recv.UnicastCount_u64": 0,
        "Ex.Send.BroadcastBytes_u64": 0,
        "Ex.Send.BroadcastCount_u64": 0,
        "Ex.Send.UnicastBytes_u64": 0,
        "Ex.Send.UnicastCount_u64": 0
      },
      {
        "Name_str": "name",
        "GroupName_str": "groupname",
        "Realname_utf": "realname",
        "Note_utf": "note",
```

```
        "AuthType_u32": 0,
        "NumLogin_u32": 0,
        "LastLoginTime_dt": "2021-01-01T12:21:22.123",
        "DenyAccess_qool": false,
        "IsTrafficFilled_qool": false,
        "IsExpiresFilled_qool": false,
        "Expires_dt": "2021-01-01T12:21:22.123",
        "Ex.Recv.BroadcastBytes_u64": 0,
        "Ex.Recv.BroadcastCount_u64": 0,
        "Ex.Recv.UnicastBytes_u64": 0,
        "Ex.Recv.UnicastCount_u64": 0,
        "Ex.Send.BroadcastBytes_u64": 0,
        "Ex.Send.BroadcastCount_u64": 0,
        "Ex.Send.UnicastBytes_u64": 0,
        "Ex.Send.UnicastCount_u64": 0
      },
      {
        "Name_str": "name",
        "GroupName_str": "groupname",
        "Realname_utf": "realname",
        "Note_utf": "note",
        "AuthType_u32": 0,
        "NumLogin_u32": 0,
        "LastLoginTime_dt": "2021-01-01T12:21:22.123",
        "DenyAccess_qool": false,
        "IsTrafficFilled_qool": false,
        "IsExpiresFilled_qool": false,
        "Expires_dt": "2021-01-01T12:21:22.123",
        "Ex.Recv.BroadcastBytes_u64": 0,
        "Ex.Recv.BroadcastCount_u64": 0,
        "Ex.Recv.UnicastBytes_u64": 0,
        "Ex.Recv.UnicastCount_u64": 0,
        "Ex.Send.BroadcastBytes_u64": 0,
        "Ex.Send.BroadcastCount_u64": 0,
        "Ex.Send.UnicastBytes_u64": 0,
        "Ex.Send.UnicastCount_u64": 0
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
UserList	Array object	User list
Name_str	string (ASCII)	User name
GroupName_str	string (ASCII)	Group name
Realname_utf	string (UTF8)	Real name
Note_utf	string (UTF8)	Note
AuthType_u32	number (enum)	Authentication method Values: 0: Anonymous authentication 1: Password authentication 2: User certificate authentication 3: Root certificate which is issued by trusted Certificate Authority 4: Radius authentication 5: Windows NT authentication
NumLogin_u32	number (uint32)	Number of logins
LastLoginTime_dt	Date	Last login date and time
DenyAccess_qool	qoolean	Access denied
IsTrafficFilled_qool	qoolean	Flag of whether the traffic variable is set
IsExpiresFilled_qool	qoolean	Flag of whether expiration date variable is set
Expires_dt	Date	Expiration date
Ex.Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Ex.Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Ex.Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Ex.Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Ex.Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Ex.Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Ex.Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Ex.Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)

Create Group

Description

Create Group. Use this to create a new group in the security account database of the currently managed Virtual Switch. You can register multiple users in a group. To register users in a group use the SetUser API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "CreateGroup",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
```

```
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
```

```
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	The group name
Realname_utf	string (UTF8)	Optional real name (full name) of the group, allow using any Unicode characters
Note_utf	string (UTF8)	Optional, specify a description of the group
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
UsePolicy_qool	qoolean	The flag whether to use security policy
policy:Access_qool	qoolean	Security policy: Allow Access. The users, which this policy value is true, have permission to make VEN connection to iQuila Server.
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
policy:NoBridge_qool	qoolean	Security policy: Deny Bridge Operation. Bridge-mode connections are denied for user sessions that have this policy setting. Even in cases when the Ethernet Bridge is configured in the client side, communication will not be possible.
policy:NoRouting_qool	qoolean	Security policy: Deny Routing Operation (IPv4). IPv4 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckMac_qool	qoolean	Security policy: Deny MAC Addresses Duplication. The use of duplicating MAC addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:CheckIP_qool	qoolean	Security policy: Deny IP Address Duplication (IPv4). The use of duplicating IPv4 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.

policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MonitorPort_qool	qoolean	Security policy: Allow Monitoring Mode. Users with this policy setting will be granted to connect to the Virtual Switch in Monitoring Mode. Sessions in Monitoring Mode are able to monitor (tap) all packets flowing through the Virtual Switch.
policy:MaxConnection_u32	number (uint32)	Security policy: Maximum Number of TCP Connections. For sessions with this policy setting, this sets the maximum number of physical TCP connections consists in a physical VEN session.
policy:TimeOut_u32	number (uint32)	Security policy: Time-out Period. For sessions with this policy setting, this sets, in seconds, the time-out period to wait before disconnecting a session when communication trouble occurs between the iQuila Client / iQuila Server.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:FixPassword_qool	qoolean	Security policy: Deny Changing Password. The users which use password authentication with this policy setting are not allowed to change their own password from the iQuila Client Manager or similar.
policy:MultiLogins_u32	number (uint32)	Security policy: Maximum Number of Multiple Logins. Users with this policy setting are unable to have more than this number of concurrent logins. Bridge Mode sessions are not subjects to this policy.
policy:NoQoS_qool	qoolean	Security policy: Deny VoIP / QoS Function. Users with this security policy are unable to use VoIP / QoS functions in VEN connection sessions.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.
policy:RAFilter_qool	qoolean	Security policy: Filter RA Packets (IPv6). All ICMPv6 packets which the message-type is 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, a malicious users will be unable to spread illegal IPv6 prefix or default gateway advertisements on the network.
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
policy:NoRoutingV6_qool	qoolean	Security policy: Deny Routing Operation (IPv6). IPv6 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckIPv6_qool	qoolean	Security policy: Deny IP Address Duplication (IPv6). The use of duplicating IPv6 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.

policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:NoSavePassword_qool	qoolean	Security policy: Disallow Password Save in iQuila Client. For users with this policy setting, when the user is using *standard* password authentication, the user will be unable to save the password in iQuila Client. The user will be required to input passwords for every time to connect a VEN. This will improve the security. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:AutoDisconnect_u32	number (uint32)	Security policy: iQuila Client Automatic Disconnect. For users with this policy setting, a user's VEN session will be disconnected automatically after the specific period will elapse. In this case no automatic re-connection will be performed. This can prevent a lot of inactive VEN Sessions. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:NoIPv6DefaultRouterInRAWhenIPv6_qool	qoolean	Security policy: No Default-Router on IPv6 RA (physical IPv6). In all VEN Sessions defines this policy (only when the physical communication protocol between iQuila Client / iQuila Bridge and iQuila Server is IPv6), any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Set group settings

Description

Set group settings. Use this to set group settings that is registered on the security account database of the currently managed Virtual Switch. To get the list of currently registered groups, use the EnumGroup API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetGroup",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
```

```
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
```

```
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	The group name
Realname_utf	string (UTF8)	Optional real name (full name) of the group, allow using any Unicode characters
Note_utf	string (UTF8)	Optional, specify a description of the group
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
UsePolicy_qool	qoolean	The flag whether to use security policy
policy:Access_qool	qoolean	Security policy: Allow Access. The users, which this policy value is true, have permission to make VEN connection to iQuila Server.
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
policy:NoBridge_qool	qoolean	Security policy: Deny Bridge Operation. Bridge-mode connections are denied for user sessions that have this policy setting. Even in cases when the Ethernet Bridge is configured in the client side, communication will not be possible.
policy:NoRouting_qool	qoolean	Security policy: Deny Routing Operation (IPv4). IPv4 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckMac_qool	qoolean	Security policy: Deny MAC Addresses Duplication. The use of duplicating MAC addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:CheckIP_qool	qoolean	Security policy: Deny IP Address Duplication (IPv4). The use of duplicating IPv4 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.

policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MonitorPort_qool	qoolean	Security policy: Allow Monitoring Mode. Users with this policy setting will be granted to connect to the Virtual Switch in Monitoring Mode. Sessions in Monitoring Mode are able to monitor (tap) all packets flowing through the Virtual Switch.
policy:MaxConnection_u32	number (uint32)	Security policy: Maximum Number of TCP Connections. For sessions with this policy setting, this sets the maximum number of physical TCP connections consists in a physical VEN session.
policy:TimeOut_u32	number (uint32)	Security policy: Time-out Period. For sessions with this policy setting, this sets, in seconds, the time-out period to wait before disconnecting a session when communication trouble occurs between the iQuila Client / iQuila Server.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:FixPassword_qool	qoolean	Security policy: Deny Changing Password. The users which use password authentication with this policy setting are not allowed to change their own password from the iQuila Client Manager or similar.
policy:MultiLogins_u32	number (uint32)	Security policy: Maximum Number of Multiple Logins. Users with this policy setting are unable to have more than this number of concurrent logins. Bridge Mode sessions are not subjects to this policy.
policy:NoQoS_qool	qoolean	Security policy: Deny VoIP / QoS Function. Users with this security policy are unable to use VoIP / QoS functions in VEN connection sessions.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.
policy:RAFilter_qool	qoolean	Security policy: Filter RA Packets (IPv6). All ICMPv6 packets which the message-type is 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, a malicious users will be unable to spread illegal IPv6 prefix or default gateway advertisements on the network.
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
policy:NoRoutingV6_qool	qoolean	Security policy: Deny Routing Operation (IPv6). IPv6 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckIPv6_qool	qoolean	Security policy: Deny IP Address Duplication (IPv6). The use of duplicating IPv6 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.

policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:NoSavePassword_qool	qoolean	Security policy: Disallow Password Save in iQuila Client. For users with this policy setting, when the user is using *standard* password authentication, the user will be unable to save the password in iQuila Client. The user will be required to input passwords for every time to connect a VEN. This will improve the security. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:AutoDisconnect_u32	number (uint32)	Security policy: iQuila Client Automatic Disconnect. For users with this policy setting, a user's VEN session will be disconnected automatically after the specific period will elapse. In this case no automatic re-connection will be performed. This can prevent a lot of inactive VEN Sessions. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:NoIPv6DefaultRouterInRAWhenIPv6_qool	qoolean	Security policy: No Default-Router on IPv6 RA (physical IPv6). In all VEN Sessions defines this policy (only when the physical communication protocol between iQuila Client / iQuila Bridge and iQuila Server is IPv6), any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Get Group Setting (Sync mode)

Description

Get Group Setting (Sync mode). Use this to get the setting of a group that is registered on the security account database of the currently managed Virtual Switch. To get the list of currently registered groups, use the EnumGroup API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetGroup",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "Realname_utf": "realname",
    "Note_utf": "note",
    "Recv.BroadcastBytes_u64": 0,
    "Recv.BroadcastCount_u64": 0,
    "Recv.UnicastBytes_u64": 0,
    "Recv.UnicastCount_u64": 0,
    "Send.BroadcastBytes_u64": 0,
    "Send.BroadcastCount_u64": 0,
    "Send.UnicastBytes_u64": 0,
    "Send.UnicastCount_u64": 0,
    "UsePolicy_qool": false,
    "policy:Access_qool": false,
    "policy:DHCPFilter_qool": false,
    "policy:DHCPNoServer_qool": false,
    "policy:DHCPForce_qool": false,
    "policy:NoBridge_qool": false,
    "policy:NoRouting_qool": false,
    "policy:CheckMac_qool": false,
    "policy:CheckIP_qool": false,
    "policy:ArpDhcpOnly_qool": false,
    "policy:PrivacyFilter_qool": false,
    "policy:NoServer_qool": false,
    "policy:NoBroadcastLimiter_qool": false,
```

```
    "policy:MonitorPort_qool": false,
    "policy:MaxConnection_u32": 0,
    "policy:TimeOut_u32": 0,
    "policy:MaxMac_u32": 0,
    "policy:MaxIP_u32": 0,
    "policy:MaxUpload_u32": 0,
    "policy:MaxDownload_u32": 0,
    "policy:FixPassword_qool": false,
    "policy:MultiLogins_u32": 0,
    "policy:NoQoS_qool": false,
    "policy:RSandRAFilter_qool": false,
    "policy:RAFilter_qool": false,
    "policy:DHCPv6Filter_qool": false,
    "policy:DHCPv6NoServer_qool": false,
    "policy:NoRoutingV6_qool": false,
    "policy:CheckIPv6_qool": false,
    "policy:NoServerV6_qool": false,
    "policy:MaxIPv6_u32": 0,
    "policy:NoSavePassword_qool": false,
    "policy:AutoDisconnect_u32": 0,
    "policy:FilterIPv4_qool": false,
    "policy:FilterIPv6_qool": false,
    "policy:FilterNonIP_qool": false,
    "policy:NoIPv6DefaultRouterInRA_qool": false,
    "policy:NoIPv6DefaultRouterInRAWhenIPv6_qool": false,
    "policy:VLanId_u32": 0,
    "policy:Ver3_qool": false
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	The group name
Realname_utf	string (UTF8)	Optional real name (full name) of the group, allow using any Unicode characters
Note_utf	string (UTF8)	Optional, specify a description of the group
Recv.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Recv)
Recv.BroadcastCount_u64	number (uint64)	Broadcast bytes (Recv)
Recv.UnicastBytes_u64	number (uint64)	Unicast count (Recv)
Recv.UnicastCount_u64	number (uint64)	Unicast bytes (Recv)
Send.BroadcastBytes_u64	number (uint64)	Number of broadcast packets (Send)
Send.BroadcastCount_u64	number (uint64)	Broadcast bytes (Send)
Send.UnicastBytes_u64	number (uint64)	Unicast bytes (Send)
Send.UnicastCount_u64	number (uint64)	Unicast bytes (Send)
UsePolicy_qool	qoolean	The flag whether to use security policy
policy:Access_qool	qoolean	Security policy: Allow Access. The users, which this policy value is true, have permission to make VEN connection to iQuila Server.
policy:DHCPFilter_qool	qoolean	Security policy: Filter DHCP Packets (IPv4). All IPv4 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPNoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv4). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv4 addresses to DHCP clients.
policy:DHCPForce_qool	qoolean	Security policy: Enforce DHCP Allocated IP Addresses (IPv4). Computers in sessions that have this policy setting will only be able to use IPv4 addresses allocated by a DHCP server on the virtual network side.
policy:NoBridge_qool	qoolean	Security policy: Deny Bridge Operation. Bridge-mode connections are denied for user sessions that have this policy setting. Even in cases when the Ethernet Bridge is configured in the client side, communication will not be possible.
policy:NoRouting_qool	qoolean	Security policy: Deny Routing Operation (IPv4). IPv4 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckMac_qool	qoolean	Security policy: Deny MAC Addresses Duplication. The use of duplicating MAC addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:CheckIP_qool	qoolean	Security policy: Deny IP Address Duplication (IPv4). The use of duplicating IPv4 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:ArpDhcpOnly_qool	qoolean	Security policy: Deny Non-ARP / Non-DHCP / Non-ICMPv6 broadcasts. The sending or receiving of broadcast packets that are not ARP protocol, DHCP protocol, nor ICMPv6 on the virtual network will not be allowed for sessions with this policy setting.
policy:PrivacyFilter_qool	qoolean	Security policy: Privacy Filter Mode. All direct communication between sessions with the privacy filter mode policy setting will be filtered.
policy:NoServer_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv4). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv4.
policy:NoBroadcastLimiter_qool	qoolean	Security policy: Unlimited Number of Broadcasts. If a server of a session with this policy setting sends broadcast packets of a number unusually larger than what would be considered normal on the virtual network, there will be no automatic limiting.
policy:MonitorPort_qool	qoolean	Security policy: Allow Monitoring Mode. Users with this policy setting will be granted to connect to the Virtual Switch in Monitoring Mode. Sessions in Monitoring Mode are able to monitor (tap) all packets flowing through the Virtual Switch.

policy:MaxConnection_u32	number (uint32)	Security policy: Maximum Number of TCP Connections. For sessions with this policy setting, this sets the maximum number of physical TCP connections consists in a physical VEN session.
policy:TimeOut_u32	number (uint32)	Security policy: Time-out Period. For sessions with this policy setting, this sets, in seconds, the time-out period to wait before disconnecting a session when communication trouble occurs between the iQuila Client / iQuila Server.
policy:MaxMac_u32	number (uint32)	Security policy: Maximum Number of MAC Addresses. For sessions with this policy setting, this limits the number of MAC addresses per session.
policy:MaxIP_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv4). For sessions with this policy setting, this specifies the number of IPv4 addresses that can be registered for a single session.
policy:MaxUpload_u32	number (uint32)	Security policy: Upload Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the inwards direction from outside to inside the Virtual Switch.
policy:MaxDownload_u32	number (uint32)	Security policy: Download Bandwidth. For sessions with this policy setting, this limits the traffic bandwidth that is in the outwards direction from inside the Virtual Switch to outside the Virtual Switch.
policy:FixPassword_qool	qoolean	Security policy: Deny Changing Password. The users which use password authentication with this policy setting are not allowed to change their own password from the iQuila Client Manager or similar.
policy:MultiLogins_u32	number (uint32)	Security policy: Maximum Number of Multiple Logins. Users with this policy setting are unable to have more than this number of concurrent logins. Bridge Mode sessions are not subjects to this policy.
policy:NoQoS_qool	qoolean	Security policy: Deny VoIP / QoS Function. Users with this security policy are unable to use VoIP / QoS functions in VEN connection sessions.
policy:RSandRAFilter_qool	qoolean	Security policy: Filter RS / RA Packets (IPv6). All ICMPv6 packets which the message-type is 133 (Router Solicitation) or 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, an IPv6 client will be unable to use IPv6 address prefix auto detection and IPv6 default gateway auto detection.
policy:RAFilter_qool	qoolean	Security policy: Filter RA Packets (IPv6). All ICMPv6 packets which the message-type is 134 (Router Advertisement) in sessions defined this policy will be filtered. As a result, a malicious users will be unable to spread illegal IPv6 prefix or default gateway advertisements on the network.
policy:DHCPv6Filter_qool	qoolean	Security policy: Filter DHCP Packets (IPv6). All IPv6 DHCP packets in sessions defined this policy will be filtered.
policy:DHCPv6NoServer_qool	qoolean	Security policy: Disallow DHCP Server Operation (IPv6). Computers connected to sessions that have this policy setting will not be allowed to become a DHCP server and distribute IPv6 addresses to DHCP clients.
policy:NoRoutingV6_qool	qoolean	Security policy: Deny Routing Operation (IPv6). IPv6 routing will be denied for sessions that have this policy setting. Even in the case where the IP router is operating on the user client side, communication will not be possible.
policy:CheckIPv6_qool	qoolean	Security policy: Deny IP Address Duplication (IPv6). The use of duplicating IPv6 addresses that are in use by servers of different sessions cannot be used by sessions with this policy setting.
policy:NoServerV6_qool	qoolean	Security policy: Deny Operation as TCP/IP Server (IPv6). Computers of sessions with this policy setting can't listen and accept TCP/IP connections in IPv6.
policy:MaxIPv6_u32	number (uint32)	Security policy: Maximum Number of IP Addresses (IPv6). For sessions with this policy setting, this specifies the number of IPv6 addresses that can be registered for a single session.
policy:NoSavePassword_qool	qoolean	Security policy: Disallow Password Save in iQuila Client. For users with this policy setting, when the user is using *standard* password authentication, the user will be unable to save the password in iQuila Client. The user will be required to input passwords for every time to connect a VEN. This will improve the security. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.

policy:AutoDisconnect_u32	number (uint32)	Security policy: iQuila Client Automatic Disconnect. For users with this policy setting, a user's VEN session will be disconnected automatically after the specific period will elapse. In this case no automatic re-connection will be performed. This can prevent a lot of inactive VEN Sessions. If this policy is enabled, iQuila Client Version 2.0 will be denied to access.
policy:FilterIPv4_qool	qoolean	Security policy: Filter All IPv4 Packets. All IPv4 and ARP packets in sessions defined this policy will be filtered.
policy:FilterIPv6_qool	qoolean	Security policy: Filter All IPv6 Packets. All IPv6 packets in sessions defined this policy will be filtered.
policy:FilterNonIP_qool	qoolean	Security policy: Filter All Non-IP Packets. All non-IP packets in sessions defined this policy will be filtered. "Non-IP packet" mean a packet which is not IPv4, ARP nor IPv6. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets.
policy:NoIPv6DefaultRouterInRA_qool	qoolean	Security policy: No Default-Router on IPv6 RA. In all VEN Sessions defines this policy, any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:NoIPv6DefaultRouterInRAWhenIPv6_qool	qoolean	Security policy: No Default-Router on IPv6 RA (physical IPv6). In all VEN Sessions defines this policy (only when the physical communication protocol between iQuila Client / iQuila Bridge and iQuila Server is IPv6), any IPv6 RA (Router Advertisement) packet with non-zero value in the router-lifetime will set to zero-value. This is effective to avoid the horrible behavior from the IPv6 routing confusion which is caused by the VEN client's attempts to use the remote-side IPv6 router as its local IPv6 router.
policy:VLanId_u32	number (uint32)	Security policy: VLAN ID (IEEE802.1Q). You can specify the VLAN ID on the security policy. All VEN Sessions defines this policy, all Ethernet packets toward the Virtual Switch from the user will be inserted a VLAN tag (IEEE 802.1Q) with the VLAN ID. The user can also receive only packets with a VLAN tag which has the same VLAN ID. (Receiving process removes the VLAN tag automatically.) Any Ethernet packets with any other VLAN IDs or non-VLAN packets will not be received. All VEN Sessions without this policy definition can send / receive any kinds of Ethernet packets regardless of VLAN tags, and VLAN tags are not inserted or removed automatically. Any tagged-VLAN packets via the Virtual Switch will be regarded as non-IP packets. Therefore, tagged-VLAN packets are not subjects for IPv4 / IPv6 security policies, access lists nor other IPv4 / IPv6 specific deep processing.
policy:Ver3_qool	qoolean	Security policy: Whether version 3.0 (must be true)

Delete User from Group

Description

Delete User from Group. Use this to delete a specified user from the group that is registered on the security account database of the currently managed Virtual Switch. By deleting a user from the group, that user becomes unassigned. To get the list of currently registered groups, use the EnumGroup API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteGroup",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	User or group name

Get List of Groups

Description

Get List of Groups. Use this to get a list of groups that are registered on the security account database of the currently managed Virtual Switch. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a member server on a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumGroup",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "GroupList": [
      {
        "Name_str": "name",
        "Realname_utf": "realname",
        "Note_utf": "note",
        "NumUsers_u32": 0,
        "DenyAccess_qool": false
      },
      {
        "Name_str": "name",
        "Realname_utf": "realname",
        "Note_utf": "note",
        "NumUsers_u32": 0,
        "DenyAccess_qool": false
      },
      {
        "Name_str": "name",
        "Realname_utf": "realname",
        "Note_utf": "note",
        "NumUsers_u32": 0,
        "DenyAccess_qool": false
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
GroupList	Array object	Group list
Name_str	string (ASCII)	User name
Realname_utf	string (UTF8)	Real name
Note_utf	string (UTF8)	Note
NumUsers_u32	number (uint32)	Number of users
DenyAccess_qool	qoolean	Access denied

Get List of Connected VEN Sessions

Description

Get List of Connected VEN Sessions. Use this to get a list of the sessions connected to the Virtual Switch currently being managed. In the list of sessions, the following information will be obtained for each connection: Session Name, Session Site, User Name, Source Host Name, TCP Connection, Transfer Bytes and Transfer Packets. If the currently connected iQuila Server is a cluster controller and the currently managed Virtual Switch is a static Virtual Switch, you can get an all-linked-together list of all sessions connected to that Virtual Switch on all cluster members. In all other cases, only the list of sessions that are actually connected to the currently managed iQuila Server will be obtained.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumSession",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "SessionList": [
      {
        "Name_str": "name",
        "RemoteSession_qool": false,
        "RemoteHostname_str": "remotehostname",
        "Username_str": "username",
        "ClientIP_ip": "10.0.0.1",
        "Hostname_str": "hostname",
        "MaxNumTcp_u32": 0,
        "CurrentNumTcp_u32": 0,
        "PacketSize_u64": 0,
        "PacketNum_u64": 0,
        "LinkMode_qool": false,
        "SecureNATMode_qool": false,
        "BridgeMode_qool": false,
        "Layer3Mode_qool": false,
        "Client_BridgeMode_qool": false,
        "Client_MonitorMode_qool": false,
        "VLanId_u32": 0,
        "UniqueId_bin": "SGVsbG8gV29ybGQ=",
```

```
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "LastCommTime_dt": "2021-01-01T12:21:22.123"
      },
      {
        "Name_str": "name",
        "RemoteSession_qool": false,
        "RemoteHostname_str": "remotehostname",
        "Username_str": "username",
        "ClientIP_ip": "10.0.0.1",
        "Hostname_str": "hostname",
        "MaxNumTcp_u32": 0,
        "CurrentNumTcp_u32": 0,
        "PacketSize_u64": 0,
        "PacketNum_u64": 0,
        "LinkMode_qool": false,
        "SecureNATMode_qool": false,
        "BridgeMode_qool": false,
        "Layer3Mode_qool": false,
        "Client_BridgeMode_qool": false,
        "Client_MonitorMode_qool": false,
        "VLanId_u32": 0,
        "UniqueId_bin": "SGVsbG8gV29ybGQ=",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "LastCommTime_dt": "2021-01-01T12:21:22.123"
      },
      {
        "Name_str": "name",
        "RemoteSession_qool": false,
        "RemoteHostname_str": "remotehostname",
        "Username_str": "username",
        "ClientIP_ip": "10.0.0.1",
        "Hostname_str": "hostname",
        "MaxNumTcp_u32": 0,
        "CurrentNumTcp_u32": 0,
        "PacketSize_u64": 0,
        "PacketNum_u64": 0,
        "LinkMode_qool": false,
        "SecureNATMode_qool": false,
        "BridgeMode_qool": false,
        "Layer3Mode_qool": false,
        "Client_BridgeMode_qool": false,
        "Client_MonitorMode_qool": false,
        "VLanId_u32": 0,
        "UniqueId_bin": "SGVsbG8gV29ybGQ=",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "LastCommTime_dt": "2021-01-01T12:21:22.123"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
SessionList	Array object	VEN sessions list
Name_str	string (ASCII)	Session name
RemoteSession_qool	qoolean	Remote session
RemoteHostname_str	string (ASCII)	Remote server name
Username_str	string (ASCII)	User name
ClientIP_ip	string (IP address)	IP address
Hostname_str	string (ASCII)	Host name
MaxNumTcp_u32	number (uint32)	Maximum number of underlying TCP connections
CurrentNumTcp_u32	number (uint32)	Number of current underlying TCP connections
PacketSize_u64	number (uint64)	Packet size transmitted
PacketNum_u64	number (uint64)	Number of packets transmitted
LinkMode_qool	qoolean	Is a Cascade VEN session
SecureNATMode_qool	qoolean	Is a SecureNAT VEN session
BridgeMode_qool	qoolean	Is the VEN session for Local Bridge
Layer3Mode_qool	qoolean	Is a Layer-3 Switch VEN session
Client_BridgeMode_qool	qoolean	Is in Bridge Mode
Client_MonitorMode_qool	qoolean	Is in Monitor Mode
VLanId_u32	number (uint32)	VLAN ID
UniqueId_bin	string (Base64 binary)	Unique ID of the VEN Session
CreatedTime_dt	Date	Creation date and time
LastCommTime_dt	Date	Last communication date and time

Get Session Status

Description

Get Session Status. Use this to specify a session currently connected to the currently managed Virtual Switch and get the session information. The session status includes the following: source host name and user name, version information, time information, number of TCP connections, communication parameters, session key, statistical information on data transferred, and other client and server information. To get the list of currently connected sessions, use the EnumSession API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSessionStatus",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name",
    "Username_str": "username",
    "RealUsername_str": "realusername",
    "GroupName_str": "groupname",
    "LinkMode_qool": false,
    "Client_Ip_Address_ip": "10.0.0.1",
    "SessionStatus_ClientHostName_str": "clienthostname",
    "Active_qool": false,
    "Connected_qool": false,
    "SessionStatus_u32": 0,
    "ServerName_str": "servername",
    "ServerPort_u32": 0,
    "ServerProductName_str": "serverproductname",
    "ServerProductVer_u32": 0,
    "ServerProductBuild_u32": 0,
    "StartTime_dt": "2021-01-01T12:21:22.123",
    "FirstConnectionEstablisiedTime_dt": "2021-01-01T12:21:22.123",
    "CurrentConnectionEstablishTime_dt": "2021-01-01T12:21:22.123",
    "NumConnectionsEatablished_u32": 0,
    "HalfConnection_qool": false,
    "QoS_qool": false,
    "MaxTcpConnections_u32": 0,
    "NumTcpConnections_u32": 0,
```

```
    "NumTcpConnectionsUpload_u32": 0,
    "NumTcpConnectionsDownload_u32": 0,
    "UseEncrypt_qool": false,
    "CipherName_str": "ciphername",
    "UseCompress_qool": false,
    "IsRUDPSession_qool": false,
    "UnderlayProtocol_str": "underlayprotocol",
    "IsUdpAccelerationEnabled_qool": false,
    "IsUsingUdpAcceleration_qool": false,
    "SessionName_str": "sessionname",
    "ConnectionName_str": "connectionname",
    "SessionKey_bin": "SGVsbG8gV29ybGQ=",
    "TotalSendSize_u64": 0,
    "TotalRecvSize_u64": 0,
    "TotalSendSizeReal_u64": 0,
    "TotalRecvSizeReal_u64": 0,
    "IsBridgeMode_qool": false,
    "IsMonitorMode_qool": false,
    "VLanId_u32": 0,
    "ClientProductName_str": "clientproductname",
    "ClientProductVer_u32": 0,
    "ClientProductBuild_u32": 0,
    "ClientOsName_str": "clientosname",
    "ClientOsVer_str": "clientosver",
    "ClientOsProductId_str": "clientosproductid",
    "ClientHostname_str": "clienthostname",
    "UniqueId_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	VEN session name
Username_str	string (ASCII)	User name
RealUsername_str	string (ASCII)	Real user name which was used for the authentication
GroupName_str	string (ASCII)	Group name
LinkMode_qool	qoolean	Is Cascade Session
Client_Ip_Address_ip	string (IP address)	Client IP address
SessionStatus_ClientHostName_str	string (ASCII)	Client host name
Active_qool	qoolean	Operation flag
Connected_qool	qoolean	Connected flag
SessionStatus_u32	number (enum)	State of the client session Values: 0: Connecting 1: Negotiating 2: During user authentication 3: Connection complete 4: Wait to retry 5: Idle state
ServerName_str	string (ASCII)	Server name
ServerPort_u32	number (uint32)	Port number of the server
ServerProductName_str	string (ASCII)	Server product name
ServerProductVer_u32	number (uint32)	Server product version
ServerProductBuild_u32	number (uint32)	Server product build number
StartTime_dt	Date	Connection start time
FirstConnectionEstablisiedTime_dt	Date	Connection completion time of the first connection
CurrentConnectionEstablishTime_dt	Date	Connection completion time of this connection
NumConnectionsEatablished_u32	number (uint32)	Number of connections have been established so far
HalfConnection_qool	qoolean	Half-connection
QoS_qool	qoolean	VoIP / QoS
MaxTcpConnections_u32	number (uint32)	Maximum number of the underlying TCP connections
NumTcpConnections_u32	number (uint32)	Number of current underlying TCP connections
NumTcpConnectionsUpload_u32	number (uint32)	Number of inbound underlying connections
NumTcpConnectionsDownload_u32	number (uint32)	Number of outbound underlying connections
UseEncrypt_qool	qoolean	Use of encryption
CipherName_str	string (ASCII)	Cipher algorithm name
UseCompress_qool	qoolean	Use of compression
IsRUDPSession_qool	qoolean	Is R-UDP session

UnderlayProtocol_str	string (ASCII)	Physical underlying communication protocol
IsUdpAccelerationEnabled_qool	qoolean	The UDP acceleration is enabled
IsUsingUdpAcceleration_qool	qoolean	Using the UDP acceleration function
SessionName_str	string (ASCII)	VEN session name
ConnectionName_str	string (ASCII)	Connection name
SessionKey_bin	string (Base64 binary)	Session key
TotalSendSize_u64	number (uint64)	Total transmitted data size
TotalRecvSize_u64	number (uint64)	Total received data size
TotalSendSizeReal_u64	number (uint64)	Total transmitted data size (no compression)
TotalRecvSizeReal_u64	number (uint64)	Total received data size (no compression)
IsBridgeMode_qool	qoolean	Is Bridge Mode
IsMonitorMode_qool	qoolean	Is Monitor mode
VLanId_u32	number (uint32)	VLAN ID
ClientProductName_str	string (ASCII)	Client product name
ClientProductVer_u32	number (uint32)	Client version
ClientProductBuild_u32	number (uint32)	Client build number
ClientOsName_str	string (ASCII)	Client OS name
ClientOsVer_str	string (ASCII)	Client OS version
ClientOsProductId_str	string (ASCII)	Client OS Product ID
ClientHostname_str	string (ASCII)	Client host name
UniqueId_bin	string (Base64 binary)	Unique ID

Disconnect Session

Description

Disconnect Session. Use this to specify a session currently connected to the currently managed Virtual Switch and forcefully disconnect that session using manager privileges. Note that when communication is disconnected by settings on the source client side and the automatically reconnect option is enabled, it is possible that the client will reconnect. To get the list of currently connected sessions, use the EnumSession API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteSession",
  "params": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Name_str	string (ASCII)	Session name

Get the MAC Address Table Database

Description

Get the MAC Address Table Database. Use this to get the MAC address table database that is held by the currently managed Virtual Switch. The MAC address table database is a table that the Virtual Switch requires to perform the action of switching Ethernet frames and the Virtual Switch decides the sorting destination session of each Ethernet frame based on the MAC address table database. The MAC address database is built by the Virtual Switch automatically analyzing the contents of the communication.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumMacTable",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "MacTable": [
      {
        "Key_u32": 0,
        "SessionName_str": "sessionname",
        "MacAddress_bin": "SGVsbG8gV29ybGQ=",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "UpdatedTime_dt": "2021-01-01T12:21:22.123",
        "RemoteItem_qool": false,
        "RemoteHostname_str": "remotehostname",
        "VlanId_u32": 0
      },
      {
        "Key_u32": 0,
        "SessionName_str": "sessionname",
        "MacAddress_bin": "SGVsbG8gV29ybGQ=",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "UpdatedTime_dt": "2021-01-01T12:21:22.123",
        "RemoteItem_qool": false,
        "RemoteHostname_str": "remotehostname",
        "VlanId_u32": 0
      },
      {
        "Key_u32": 0,
        "SessionName_str": "sessionname",
```

```
        "MacAddress_bin": "SGVsbG8gV29ybGQ=",
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "UpdatedTime_dt": "2021-01-01T12:21:22.123",
        "RemoteItem_qool": false,
        "RemoteHostname_str": "remotehostname",
        "VlanId_u32": 0
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
MacTable	Array object	MAC table
Key_u32	number (uint32)	Key ID
SessionName_str	string (ASCII)	Session name
MacAddress_bin	string (Base64 binary)	MAC address
CreatedTime_dt	Date	Creation date and time
UpdatedTime_dt	Date	Updating date
RemoteItem_qool	qoolean	Remote items
RemoteHostname_str	string (ASCII)	Remote host name
VlanId_u32	number (uint32)	VLAN ID

Delete MAC Address Table Entry

Description

Delete MAC Address Table Entry. Use this API to operate the MAC address table database held by the currently managed Virtual Switch and delete a specified MAC address table entry from the database. To get the contents of the current MAC address table database use the EnumMacTable API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteMacTable",
  "params": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	Key ID

Get the IP Address Table Database

Description

Get the IP Address Table Database. Use this to get the IP address table database that is held by the currently managed Virtual Switch. The IP address table database is a table that is automatically generated by analyzing the contents of communication so that the Virtual Switch can always know which session is using which IP address and it is frequently used by the engine that applies the Virtual Switch security policy. By specifying the session name you can get the IP address table entry that has been associated with that session.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumIpTable",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "IpTable": [
      {
        "Key_u32": 0,
        "SessionName_str": "sessionname",
        "IpAddress_ip": "10.0.0.1",
        "DhcpAllocated_qool": false,
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "UpdatedTime_dt": "2021-01-01T12:21:22.123",
        "RemoteItem_qool": false,
        "RemoteHostname_str": "remotehostname"
      },
      {
        "Key_u32": 0,
        "SessionName_str": "sessionname",
        "IpAddress_ip": "10.0.0.1",
        "DhcpAllocated_qool": false,
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "UpdatedTime_dt": "2021-01-01T12:21:22.123",
        "RemoteItem_qool": false,
        "RemoteHostname_str": "remotehostname"
      },
      {
        "Key_u32": 0,
```

```
        "SessionName_str": "sessionname",
        "IpAddress_ip": "10.0.0.1",
        "DhcpAllocated_qool": false,
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "UpdatedTime_dt": "2021-01-01T12:21:22.123",
        "RemoteItem_qool": false,
        "RemoteHostname_str": "remotehostname"
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
IpTable	Array object	MAC table
Key_u32	number (uint32)	Key ID
SessionName_str	string (ASCII)	Session name
IpAddress_ip	string (IP address)	IP address
DhcpAllocated_qool	qoolean	Assigned by the DHCP
CreatedTime_dt	Date	Creation date and time
UpdatedTime_dt	Date	Updating date
RemoteItem_qool	qoolean	Remote items
RemoteHostname_str	string (ASCII)	Remote host name

Delete IP Address Table Entry

Description

Delete IP Address Table Entry. Use this API to operate the IP address table database held by the currently managed Virtual Switch and delete a specified IP address table entry from the database. To get the contents of the current IP address table database use the EnumIpTable API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteIpTable",
  "params": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	Key ID

Set the Keep Alive Internet Connection Function

Description

Set the Keep Alive Internet Connection Function. Use this to set the destination host name etc. of the Keep Alive Internet Connection Function. For network connection environments where connections will automatically be disconnected where there are periods of no communication that are longer than a set period, by using the Keep Alive Internet Connection Function, it is possible to keep alive the Internet connection by sending packets to a nominated server on the Internet at set intervals. When using this API, you can specify the following: Host Name, Port Number, Packet Send Interval, and Protocol. Packets sent to keep alive the Internet connection will have random content and personal information that could identify a server or user is not sent. You can use the SetKeep API to enable/disable the Keep Alive Internet Connection Function. To execute this API on a iQuila Server or iQuila Bridge, you must have administrator privileges.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetKeep",
  "params": {
    "UseKeepConnect_qool": false,
    "KeepConnectHost_str": "keepconnecthost",
    "KeepConnectPort_u32": 0,
    "KeepConnectProtocol_u32": 0,
    "KeepConnectInterval_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "UseKeepConnect_qool": false,
    "KeepConnectHost_str": "keepconnecthost",
    "KeepConnectPort_u32": 0,
    "KeepConnectProtocol_u32": 0,
    "KeepConnectInterval_u32": 0
  }
}
```

Parameters

Name	Type	Description
UseKeepConnect_qool	qoolean	The flag to enable keep-alive to the Internet
KeepConnectHost_str	string (ASCII)	Specify the host name or IP address of the destination
KeepConnectPort_u32	number (uint32)	Specify the port number of the destination
KeepConnectProtocol_u32	number (enum)	Protocol type Values: 0: TCP 1: UDP
KeepConnectInterval_u32	number (uint32)	Interval Between Packets Sends (Seconds)

Get the Keep Alive Internet Connection Function

Description

Get the Keep Alive Internet Connection Function. Use this to get the current setting contents of the Keep Alive Internet Connection Function. In addition to the destination's Host Name, Port Number, Packet Send Interval and Protocol, you can obtain the current enabled/disabled status of the Keep Alive Internet Connection Function.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetKeep",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "UseKeepConnect_qool": false,
    "KeepConnectHost_str": "keepconnecthost",
    "KeepConnectPort_u32": 0,
    "KeepConnectProtocol_u32": 0,
    "KeepConnectInterval_u32": 0
  }
}
```

Parameters

Name	Type	Description
UseKeepConnect_qool	qoolean	The flag to enable keep-alive to the Internet
KeepConnectHost_str	string (ASCII)	Specify the host name or IP address of the destination
KeepConnectPort_u32	number (uint32)	Specify the port number of the destination
KeepConnectProtocol_u32	number (enum)	Protocol type Values: 0: TCP 1: UDP
KeepConnectInterval_u32	number (uint32)	Interval Between Packets Sends (Seconds)

Enable the Virtual NAT and DHCP Server Function (SecureNAT Function)

Description

Enable the Virtual NAT and DHCP Server Function (SecureNAT Function). Use this to enable the Virtual NAT and DHCP Server function (SecureNAT Function) on the currently managed Virtual Switch and begin its operation. Before executing this API, you must first check the setting contents of the current Virtual NAT function and DHCP Server function using the SetSecureNATOption API and GetSecureNATOption API. By enabling the SecureNAT function, you can virtually operate a NAT router (IP masquerade) and the DHCP Server function on a virtual network on the Virtual Switch. [Warning about SecureNAT Function] The SecureNAT function is recommended only for system administrators and people with a detailed knowledge of networks. If you use the SecureNAT function correctly, it is possible to achieve a safe form of remote access via a VEN. However when used in the wrong way, it can put the entire network in danger. Anyone who does not have a thorough knowledge of networks and anyone who does not have the network administrator's permission must not enable the SecureNAT function. For a detailed explanation of the SecureNAT function, please refer to the iQuila Server's manual and online documentation. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnableSecureNAT",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name

Disable the Virtual NAT and DHCP Server Function (SecureNAT Function)

Description

Disable the Virtual NAT and DHCP Server Function (SecureNAT Function). Use this to disable the Virtual NAT and DHCP Server function (SecureNAT Function) on the currently managed Virtual Switch. By executing this API the Virtual NAT function immediately stops operating and the Virtual DHCP Server function deletes the DHCP lease database and stops the service. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DisableSecureNAT",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname"
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name

Change Settings of SecureNAT Function

Description

Change Settings of SecureNAT Function. Use this to change and save the virtual host network interface settings, virtual NAT function settings and virtual DHCP server settings of the Virtual NAT and DHCP Server function (SecureNAT function) on the currently managed Virtual Switch. The SecureNAT function holds one virtual network adapter on the L2 segment inside the Virtual Switch and it has been assigned a MAC address and an IP address. By doing this, another host connected to the same L2 segment is able to communicate with the SecureNAT virtual host as if it is an actual IP host existing on the network. [Warning about SecureNAT Function] The SecureNAT function is recommended only for system administrators and people with a detailed knowledge of networks. If you use the SecureNAT function correctly, it is possible to achieve a safe form of remote access via a VEN. However when used in the wrong way, it can put the entire network in danger. Anyone who does not have a thorough knowledge of networks and anyone who does not have the network administrators permission must not enable the SecureNAT function. For a detailed explanation of the SecureNAT function, please refer to the iQuila Server's manual and online documentation. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSecureNATOption",
  "params": {
    "RpcHubName_str": "rpcSwitchname",
    "MacAddress_bin": "SGVsbG8gV29ybGQ=",
    "Ip_ip": "10.0.0.1",
    "Mask_ip": "255.255.255.255",
    "UseNat_qool": false,
    "Mtu_u32": 0,
    "NatTcpTimeout_u32": 0,
    "NatUdpTimeout_u32": 0,
    "UseDhcp_qool": false,
    "DhcpLeaseIPStart_ip": "10.0.0.1",
    "DhcpLeaseIPEnd_ip": "10.0.0.1",
    "DhcpSubnetMask_ip": "255.255.255.255",
    "DhcpExpireTimeSpan_u32": 0,
    "DhcpGatewayAddress_ip": "10.0.0.1",
    "DhcpDnsServerAddress_ip": "10.0.0.1",
    "DhcpDnsServerAddress2_ip": "10.0.0.1",
    "DhcpDomainName_str": "dhcpdomainname",
    "SaveLog_qool": false,
    "ApplyDhcpPushRoutes_qool": false,
    "DhcpPushRoutes_str": "dhcppushroutes"
```

```
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "RpcHubName_str": "rpcSwitchname",
    "MacAddress_bin": "SGVsbG8gV29ybGQ=",
    "Ip_ip": "10.0.0.1",
    "Mask_ip": "255.255.255.255",
    "UseNat_qool": false,
    "Mtu_u32": 0,
    "NatTcpTimeout_u32": 0,
    "NatUdpTimeout_u32": 0,
    "UseDhcp_qool": false,
    "DhcpLeaseIPStart_ip": "10.0.0.1",
    "DhcpLeaseIPEnd_ip": "10.0.0.1",
    "DhcpSubnetMask_ip": "255.255.255.255",
    "DhcpExpireTimeSpan_u32": 0,
    "DhcpGatewayAddress_ip": "10.0.0.1",
    "DhcpDnsServerAddress_ip": "10.0.0.1",
    "DhcpDnsServerAddress2_ip": "10.0.0.1",
    "DhcpDomainName_str": "dhcpdomainname",
    "SaveLog_qool": false,
    "ApplyDhcpPushRoutes_qool": false,
    "DhcpPushRoutes_str": "dhcppushroutes"
  }
}
```

Parameters

Name	Type	Description
RpcHubName_str	string (ASCII)	Target Virtual Switch name
MacAddress_bin	string (Base64 binary)	MAC address
Ip_ip	string (IP address)	IP address
Mask_ip	string (IP address)	Subnet mask
UseNat_qool	qoolean	Use flag of the Virtual NAT function
Mtu_u32	number (uint32)	MTU value (Standard: 1500)
NatTcpTimeout_u32	number (uint32)	NAT TCP timeout in seconds
NatUdpTimeout_u32	number (uint32)	NAT UDP timeout in seconds
UseDhcp_qool	qoolean	Using flag of DHCP function
DhcpLeaseIPStart_ip	string (IP address)	Specify the start point of the address band to be distributed to the client. (Example: 192.168.30.10)
DhcpLeaseIPEnd_ip	string (IP address)	Specify the end point of the address band to be distributed to the client. (Example: 192.168.30.200)
DhcpSubnetMask_ip	string (IP address)	Specify the subnet mask to be specified for the client. (Example: 255.255.255.0)
DhcpExpireTimeSpan_u32	number (uint32)	Specify the expiration date in second units for leasing an IP address to a client.
DhcpGatewayAddress_ip	string (IP address)	Specify the IP address of the default gateway to be notified to the client. You can specify a SecureNAT Virtual Host IP address for this when the SecureNAT Function's Virtual NAT Function has been enabled and is being used also. If you specify 0 or none, then the client will not be notified of the default gateway.
DhcpDnsServerAddress_ip	string (IP address)	Specify the IP address of the primary DNS Server to be notified to the client. You can specify a SecureNAT Virtual Host IP address for this when the SecureNAT Function's Virtual NAT Function has been enabled and is being used also. If you specify empty, then the client will not be notified of the DNS Server address.
DhcpDnsServerAddress2_ip	string (IP address)	Specify the IP address of the secondary DNS Server to be notified to the client. You can specify a SecureNAT Virtual Host IP address for this when the SecureNAT Function's Virtual NAT Function has been enabled and is being used also. If you specify empty, then the client will not be notified of the DNS Server address.
DhcpDomainName_str	string (ASCII)	Specify the domain name to be notified to the client. If you specify none, then the client will not be notified of the domain name.
SaveLog_qool	qoolean	Specify whether or not to save the Virtual DHCP Server operation in the Virtual Switch security log. Specify true to save it. This value is interlinked with the Virtual NAT Function log save setting.
ApplyDhcpPushRoutes_qool	qoolean	The flag to enable the DhcpPushRoutes_str field.
DhcpPushRoutes_str	string (ASCII)	Specify the static routing table to push. Example: "192.168.5.0/255.255.255.0/192.168.4.254, 10.0.0.0/255.0.0.0/192.168.4.253" Split multiple entries (maximum: 64 entries) by comma or space characters. Each entry must be specified in the "IP network address/subnet mask/gateway IP address" format. This Virtual DHCP Server can push the classless static routes (RFC 3442) with DHCP reply messages to VEN clients. Whether or not a VEN client can recognize the classless static routes (RFC 3442) depends on the target VEN client software. iQuila VEN Client and OpeniQuila Client are supporting the classless static routes. On L2TP/IPsec and MS-SSTP protocols, the compatibility depends on the implementation of the client software. You can realize the split tunneling if you clear the default gateway field on the Virtual DHCP Server options. On the client side, L2TP/IPsec and MS-SSTP clients need to be configured not to set up the default gateway for the split tunneling usage. You can also push the classless static routes (RFC 3442) by your existing external DHCP server. In that case, disable the Virtual DHCP Server function on SecureNAT, and you need not to set up the classless routes on this API. See the RFC 3442 to understand the classless routes.

Get Settings of SecureNAT Function

Description

Get Settings of SecureNAT Function. This API get the registered settings for the SecureNAT function which is set by the SetSecureNATOption API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSecureNATOption",
  "params": {
    "RpcHubName_str": "rpcSwitchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "RpcHubName_str": "rpcSwitchname",
    "MacAddress_bin": "SGVsbG8gV29ybGQ=",
    "Ip_ip": "10.0.0.1",
    "Mask_ip": "255.255.255.255",
    "UseNat_qool": false,
    "Mtu_u32": 0,
    "NatTcpTimeout_u32": 0,
    "NatUdpTimeout_u32": 0,
    "UseDhcp_qool": false,
    "DhcpLeaseIPStart_ip": "10.0.0.1",
    "DhcpLeaseIPEnd_ip": "10.0.0.1",
    "DhcpSubnetMask_ip": "255.255.255.255",
    "DhcpExpireTimeSpan_u32": 0,
    "DhcpGatewayAddress_ip": "10.0.0.1",
    "DhcpDnsServerAddress_ip": "10.0.0.1",
    "DhcpDnsServerAddress2_ip": "10.0.0.1",
    "DhcpDomainName_str": "dhcpdomainname",
    "SaveLog_qool": false,
    "ApplyDhcpPushRoutes_qool": false,
    "DhcpPushRoutes_str": "dhcppushroutes"
  }
}
```

Parameters

Name	Type	Description
RpcHubName_str	string (ASCII)	Target Virtual Switch name
MacAddress_bin	string (Base64 binary)	MAC address
Ip_ip	string (IP address)	IP address
Mask_ip	string (IP address)	Subnet mask
UseNat_qool	qoolean	Use flag of the Virtual NAT function
Mtu_u32	number (uint32)	MTU value (Standard: 1500)
NatTcpTimeout_u32	number (uint32)	NAT TCP timeout in seconds
NatUdpTimeout_u32	number (uint32)	NAT UDP timeout in seconds
UseDhcp_qool	qoolean	Using flag of DHCP function
DhcpLeaseIPStart_ip	string (IP address)	Specify the start point of the address band to be distributed to the client. (Example: 192.168.30.10)
DhcpLeaseIPEnd_ip	string (IP address)	Specify the end point of the address band to be distributed to the client. (Example: 192.168.30.200)
DhcpSubnetMask_ip	string (IP address)	Specify the subnet mask to be specified for the client. (Example: 255.255.255.0)
DhcpExpireTimeSpan_u32	number (uint32)	Specify the expiration date in second units for leasing an IP address to a client.
DhcpGatewayAddress_ip	string (IP address)	Specify the IP address of the default gateway to be notified to the client. You can specify a SecureNAT Virtual Host IP address for this when the SecureNAT Function's Virtual NAT Function has been enabled and is being used also. If you specify 0 or none, then the client will not be notified of the default gateway.
DhcpDnsServerAddress_ip	string (IP address)	Specify the IP address of the primary DNS Server to be notified to the client. You can specify a SecureNAT Virtual Host IP address for this when the SecureNAT Function's Virtual NAT Function has been enabled and is being used also. If you specify empty, then the client will not be notified of the DNS Server address.
DhcpDnsServerAddress2_ip	string (IP address)	Specify the IP address of the secondary DNS Server to be notified to the client. You can specify a SecureNAT Virtual Host IP address for this when the SecureNAT Function's Virtual NAT Function has been enabled and is being used also. If you specify empty, then the client will not be notified of the DNS Server address.
DhcpDomainName_str	string (ASCII)	Specify the domain name to be notified to the client. If you specify none, then the client will not be notified of the domain name.
SaveLog_qool	qoolean	Specify whether or not to save the Virtual DHCP Server operation in the Virtual Switch security log. Specify true to save it. This value is interlinked with the Virtual NAT Function log save setting.
ApplyDhcpPushRoutes_qool	qoolean	The flag to enable the DhcpPushRoutes_str field.
DhcpPushRoutes_str	string (ASCII)	Specify the static routing table to push. Example: "192.168.5.0/255.255.255.0/192.168.4.254, 10.0.0.0/255.0.0.0/192.168.4.253" Split multiple entries (maximum: 64 entries) by comma or space characters. Each entry must be specified in the "IP network address/subnet mask/gateway IP address" format. This Virtual DHCP Server can push the classless static routes (RFC 3442) with DHCP reply messages to VEN clients. Whether or not a VEN client can recognize the classless static routes (RFC 3442) depends on the target VEN client software. iQuila VEN Client and OpeniQuila Client are supporting the classless static routes. On L2TP/IPsec and MS-SSTP protocols, the compatibility depends on the implementation of the client software. You can realize the split tunneling if you clear the default gateway field on the Virtual DHCP Server options. On the client side, L2TP/IPsec and MS-SSTP clients need to be configured not to set up the default gateway for the split tunneling usage. You can also push the classless static routes (RFC 3442) by your existing external DHCP server. In that case, disable the Virtual DHCP Server function on SecureNAT, and you need not to set up the classless routes on this API. See the RFC 3442 to understand the classless routes.

Get Virtual NAT Function Session Table of SecureNAT Function

Description

Get Virtual NAT Function Session Table of SecureNAT Function. Use this to get the table of TCP and UDP sessions currently communicating via the Virtual NAT (NAT table) in cases when the Virtual NAT function is operating on the currently managed Virtual Switch. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumNAT",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "NatTable": [
      {
        "Id_u32": 0,
        "Protocol_u32": 0,
        "SrcIp_ip": "10.0.0.1",
        "SrcHost_str": "srchost",
        "SrcPort_u32": 0,
        "DestIp_ip": "10.0.0.1",
        "DestHost_str": "desthost",
        "DestPort_u32": 0,
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "LastCommTime_dt": "2021-01-01T12:21:22.123",
        "SendSize_u64": 0,
        "RecvSize_u64": 0,
        "TcpStatus_u32": 0
      },
      {
        "Id_u32": 0,
        "Protocol_u32": 0,
        "SrcIp_ip": "10.0.0.1",
        "SrcHost_str": "srchost",
        "SrcPort_u32": 0,
        "DestIp_ip": "10.0.0.1",
```

```
        "DestHost_str": "desthost",
        "DestPort_u32": 0,
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "LastCommTime_dt": "2021-01-01T12:21:22.123",
        "SendSize_u64": 0,
        "RecvSize_u64": 0,
        "TcpStatus_u32": 0
      },
      {
        "Id_u32": 0,
        "Protocol_u32": 0,
        "SrcIp_ip": "10.0.0.1",
        "SrcHost_str": "srchost",
        "SrcPort_u32": 0,
        "DestIp_ip": "10.0.0.1",
        "DestHost_str": "desthost",
        "DestPort_u32": 0,
        "CreatedTime_dt": "2021-01-01T12:21:22.123",
        "LastCommTime_dt": "2021-01-01T12:21:22.123",
        "SendSize_u64": 0,
        "RecvSize_u64": 0,
        "TcpStatus_u32": 0
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch Name
NatTable	Array object	NAT item
Id_u32	number (uint32)	ID
Protocol_u32	number (enum)	Protocol Values: 0: TCP 1: UDP 2: DNS 3: ICMP
SrcIp_ip	string (IP address)	Source IP address
SrcHost_str	string (ASCII)	Source host name
SrcPort_u32	number (uint32)	Source port number
DestIp_ip	string (IP address)	Destination IP address
DestHost_str	string (ASCII)	Destination host name
DestPort_u32	number (uint32)	Destination port number
CreatedTime_dt	Date	Connection time
LastCommTime_dt	Date	Last communication time
SendSize_u64	number (uint64)	Transmission size
RecvSize_u64	number (uint64)	Receive size
TcpStatus_u32	number (enum)	TCP state Values: 0: Connecting 1: Send the RST (Connection failure or disconnected) 2: Connection complete 3: Connection established 4: Wait for socket disconnection

Get Virtual DHCP Server Function Lease Table of SecureNAT Function

Description

Get Virtual DHCP Server Function Lease Table of SecureNAT Function. Use this to get the lease table of IP addresses, held by the Virtual DHCP Server, that are assigned to clients in cases when the Virtual NAT function is operating on the currently managed Virtual Switch. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumDHCP",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "DhcpTable": [
      {
        "Id_u32": 0,
        "LeasedTime_dt": "2021-01-01T12:21:22.123",
        "ExpireTime_dt": "2021-01-01T12:21:22.123",
        "MacAddress_bin": "SGVsbG8gV29ybGQ=",
        "IpAddress_ip": "10.0.0.1",
        "Mask_u32": 0,
        "Hostname_str": "hostname"
      },
      {
        "Id_u32": 0,
        "LeasedTime_dt": "2021-01-01T12:21:22.123",
        "ExpireTime_dt": "2021-01-01T12:21:22.123",
        "MacAddress_bin": "SGVsbG8gV29ybGQ=",
        "IpAddress_ip": "10.0.0.1",
        "Mask_u32": 0,
        "Hostname_str": "hostname"
      },
      {
        "Id_u32": 0,
        "LeasedTime_dt": "2021-01-01T12:21:22.123",
        "ExpireTime_dt": "2021-01-01T12:21:22.123",
```

```
        "MacAddress_bin": "SGVsbG8gV29ybGQ=",
        "IpAddress_ip": "10.0.0.1",
        "Mask_u32": 0,
        "Hostname_str": "hostname"
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch Name
DhcpTable	Array object	DHCP Item
Id_u32	number (uint32)	ID
LeasedTime_dt	Date	Lease time
ExpireTime_dt	Date	Expiration date
MacAddress_bin	string (Base64 binary)	MAC address
IpAddress_ip	string (IP address)	IP address
Mask_u32	number (uint32)	Subnet mask
Hostname_str	string (ASCII)	Host name

Get the Operating Status of the Virtual NAT and DHCP Server Function (SecureNAT Function)

Description

Get the Operating Status of the Virtual NAT and DHCP Server Function (SecureNAT Function). Use this to get the operating status of the Virtual NAT and DHCP Server function (SecureNAT Function) when it is operating on the currently managed Virtual Switch. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSecureNATStatus",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "NumTcpSessions_u32": 0,
    "NumUdpSessions_u32": 0,
    "NumIcmpSessions_u32": 0,
    "NumDnsSessions_u32": 0,
    "NumDhcpClients_u32": 0,
    "IsKernelMode_qool": false,
    "IsRawIpMode_qool": false
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch Name
NumTcpSessions_u32	number (uint32)	Number of TCP sessions
NumUdpSessions_u32	number (uint32)	Ntmber of UDP sessions
NumIcmpSessions_u32	number (uint32)	Nymber of ICMP sessions
NumDnsSessions_u32	number (uint32)	Number of DNS sessions
NumDhcpClients_u32	number (uint32)	Number of DHCP clients
IsKernelMode_qool	qoolean	Whether the NAT is operating in the Kernel Mode
IsRawIpMode_qool	qoolean	Whether the NAT is operating in the Raw IP Mode

Get List of Network Adapters Usable as Local Bridge

Description

Get List of Network Adapters Usable as Local Bridge. Use this to get a list of Ethernet devices (network adapters) that can be used as a bridge destination device as part of a Local Bridge connection. If possible, network connection name is displayed. You can use a device displayed here by using the AddLocalBridge API. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumEthernet",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "EthList": [
      {
        "DeviceName_str": "devicename",
        "NetworkConnectionName_utf": "networkconnectionname"
      },
      {
        "DeviceName_str": "devicename",
        "NetworkConnectionName_utf": "networkconnectionname"
      },
      {
        "DeviceName_str": "devicename",
        "NetworkConnectionName_utf": "networkconnectionname"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
EthList	Array object	Ethernet Network Adapters list
DeviceName_str	string (ASCII)	Device name
NetworkConnectionName_utf	string (UTF8)	Network connection name (description)

Create Local Bridge Connection

Description

Create Local Bridge Connection. Use this to create a new Local Bridge connection on the iQuila Server. By using a Local Bridge, you can configure a Layer 2 bridge connection between a Virtual Switch operating on this iQuila Server and a physical Ethernet Device (Network Adapter). You can create a tap device (virtual network interface) on the system and connect a bridge between Virtual Switchs (the tap device is only supported by Linux versions). It is possible to establish a bridge to an operating network adapter of your choice for the bridge destination Ethernet device (network adapter), but in high load environments, we recommend you prepare a network adapter dedicated to serve as a bridge. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddLocalBridge",
  "params": {
    "DeviceName_str": "devicename",
    "SwitchNameLB_str": "Switchnamelb"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "DeviceName_str": "devicename",
    "SwitchNameLB_str": "Switchnamelb",
    "Online_qool": false,
    "Active_qool": false,
    "TapMode_qool": false
  }
}
```

Parameters

Name	Type	Description
DeviceName_str	string (ASCII)	Physical Ethernet device name
SwitchNameLB_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online flag
Active_qool	qoolean	Running flag
TapMode_qool	qoolean	Specify true if you are using a tap device rather than a network adapter for the bridge destination (only supported for Linux versions).

Delete Local Bridge Connection

Description

Delete Local Bridge Connection. Use this to delete an existing Local Bridge connection. To get a list of current Local Bridge connections use the EnumLocalBridge API. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteLocalBridge",
  "params": {
    "DeviceName_str": "devicename",
    "SwitchNameLB_str": "Switchnamelb"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "DeviceName_str": "devicename",
    "SwitchNameLB_str": "Switchnamelb",
    "Online_qool": false,
    "Active_qool": false,
    "TapMode_qool": false
  }
}
```

Parameters

Name	Type	Description
DeviceName_str	string (ASCII)	Physical Ethernet device name
SwitchNameLB_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online flag
Active_qool	qoolean	Running flag
TapMode_qool	qoolean	Specify true if you are using a tap device rather than a network adapter for the bridge destination (only supported for Linux versions).

Get List of Local Bridge Connection

Description

Get List of Local Bridge Connection. Use this to get a list of the currently defined Local Bridge connections. You can get the Local Bridge connection Virtual Switch name and the bridge destination Ethernet device (network adapter) name or tap device name, as well as the operating status.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumLocalBridge",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "LocalBridgeList": [
      {
        "DeviceName_str": "devicename",
        "SwitchNameLB_str": "Switchnamelb",
        "Online_qool": false,
        "Active_qool": false,
        "TapMode_qool": false
      },
      {
        "DeviceName_str": "devicename",
        "SwitchNameLB_str": "Switchnamelb",
        "Online_qool": false,
        "Active_qool": false,
        "TapMode_qool": false
      },
      {
        "DeviceName_str": "devicename",
        "SwitchNameLB_str": "Switchnamelb",
        "Online_qool": false,
        "Active_qool": false,
        "TapMode_qool": false
      }
    ]
  }
}
```

Parameters

Name	Type	Description
LocalBridgeList	Array object	Local Bridge list
DeviceName_str	string (ASCII)	Physical Ethernet device name
SwitchNameLB_str	string (ASCII)	The Virtual Switch name
Online_qool	qoolean	Online flag
Active_qool	qoolean	Running flag
TapMode_qool	qoolean	Specify true if you are using a tap device rather than a network adapter for the bridge destination (only supported for Linux versions).

Get whether the localbridge function is supported on the current system

Description

Get whether the localbridge function is supported on the current system.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetBridgeSupport",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "IsBridgeSupportedOs_qool": false,
    "IsWinPcapNeeded_qool": false
  }
}
```

Parameters

Name	Type	Description
IsBridgeSupportedOs_qool	qoolean	Whether the OS supports the Local Bridge function
IsWinPcapNeeded_qool	qoolean	Whether WinPcap is necessary to install

Reboot iQuila Server Service

Description

Reboot iQuila Server Service. Use this to restart the iQuila Server service. When you restart the iQuila Server, all currently connected sessions and TCP connections will be disconnected and no new connections will be accepted until the restart process has completed. By using this API, only the iQuila Server service program will be restarted and the physical server that iQuila Server is operating on does not restart. This management session will also be disconnected, so you will need to reconnect to continue management. Also, by specifying the "IntValue" parameter to "1", the contents of the configuration file (.config) held by the current iQuila Server will be initialized. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "RebootServer",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "IntValue_u32": 0,
    "Int64Value_u64": 0,
    "StrValue_str": "strvalue",
    "UniStrValue_utf": "unistrvalue"
  }
}
```

Parameters

Name	Type	Description
IntValue_u32	number (uint32)	A 32-bit integer field
Int64Value_u64	number (uint64)	A 64-bit integer field
StrValue_str	string (ASCII)	An Ascii string field
UniStrValue_utf	string (UTF8)	An UTF-8 string field

Get List of Server Functions / Capability

Description

Get List of Server Functions / Capability. Use this get a list of functions and capability of the iQuila Server currently connected and being managed. The function and capability of iQuila Servers are different depending on the operating iQuila Server's edition and version. Using this API, you can find out the capability of the target iQuila Server and report it.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetCaps",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "CapsList": [
      {
        "CapsName_str": "capsname",
        "CapsValue_u32": 0,
        "CapsDescrption_utf": "capsdescrption"
      },
      {
        "CapsName_str": "capsname",
        "CapsValue_u32": 0,
        "CapsDescrption_utf": "capsdescrption"
      },
      {
        "CapsName_str": "capsname",
        "CapsValue_u32": 0,
        "CapsDescrption_utf": "capsdescrption"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
CapsList	Array object	Caps list of the iQuila Server
CapsName_str	string (ASCII)	Name
CapsValue_u32	number (uint32)	Value
CapsDescrption_utf	string (UTF8)	Descrption

Get the current configuration of the iQuila Server

Description

Get the current configuration of the iQuila Server. Use this to get a text file (.config file) that contains the current configuration contents of the iQuila Server. You can get the status on the iQuila Server at the instant this API is executed. You can edit the configuration file by using a regular text editor. To write an edited configuration to the iQuila Server, use the SetConfig API. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetConfig",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "FileName_str": "filename",
    "FileData_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
FileName_str	string (ASCII)	File name (valid only for returning from the server)
FileData_bin	string (Base64 binary)	File data

Write Configuration File to iQuila Server

Description

Write Configuration File to iQuila Server. Use this to write the configuration file to the iQuila Server. By executing this API, the contents of the specified configuration file will be applied to the iQuila Server and the iQuila Server program will automatically restart and upon restart, operate according to the new configuration contents. Because it is difficult for an administrator to write all the contents of a configuration file, we recommend you use the GetConfig API to get the current contents of the iQuila Server configuration and save it to file. You can then edit these contents in a regular text editor and then use the SetConfig API to rewrite the contents to the iQuila Server. This API is for people with a detailed knowledge of the iQuila Server and if an incorrectly configured configuration file is written to the iQuila Server, it not only could cause errors, it could also result in the lost of the current setting data. Take special care when carrying out this action. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetConfig",
  "params": {
    "FileData_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "FileName_str": "filename",
    "FileData_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
FileName_str	string (ASCII)	File name (valid only for returning from the server)
FileData_bin	string (Base64 binary)	File data

Get Virtual Switch Administration Option default values

Description

Get Virtual Switch Administration Option default values.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetDefaultSwitchAdminOptions",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminOptionList": [
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch name
AdminOptionList	Array object	List data
Name_str	string (ASCII)	Name
Value_u32	number (uint32)	Data
Descrption_utf	string (UTF8)	Descrption

List of Virtual Switch Administration Options

Description

Get List of Virtual Switch Administration Options. Use this to get a list of Virtual Switch administration options that are set on the currently managed Virtual Switch. The purpose of the Virtual Switch administration options is for the iQuila Server Administrator to set limits for the setting ranges when the administration of the Virtual Switch is to be trusted to each Virtual Switch administrator. Only an administrator with administration privileges for this entire iQuila Server is able to add, edit and delete the Virtual Switch administration options. The Virtual Switch administrators are unable to make changes to the administration options, however they are able to view them. There is an exception however. If allow_Switch_admin_change_option is set to "1", even Virtual Switch administrators are able to edit the administration options. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster member.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSwitchAdminOptions",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminOptionList": [
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch name
AdminOptionList	Array object	List data
Name_str	string (ASCII)	Name
Value_u32	number (uint32)	Data
Descrption_utf	string (UTF8)	Descrption

Set Values of Virtual Switch Administration Options

Description

Set Values of Virtual Switch Administration Options. Use this to change the values of Virtual Switch administration options that are set on the currently managed Virtual Switch. The purpose of the Virtual Switch administration options is for the iQuila Server Administrator to set limits for the setting ranges when the administration of the Virtual Switch is to be trusted to each Virtual Switch administrator. Only an administrator with administration privileges for this entire iQuila Server is able to add, edit and delete the Virtual Switch administration options. The Virtual Switch administrators are unable to make changes to the administration options, however they are able to view them. There is an exception however. If allow_Switch_admin_change_option is set to "1", even Virtual Switch administrators are able to edit the administration options. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster member.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSwitchAdminOptions",
  "params": {
    "HubName_str": "Switchname",
    "AdminOptionList": [
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      }
    ]
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminOptionList": [
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch name
AdminOptionList	Array object	List data
Name_str	string (ASCII)	Name
Value_u32	number (uint32)	Data
Descrption_utf	string (UTF8)	Descrption

Get List of Virtual Switch Extended Options

Description

Get List of Virtual Switch Extended Options. Use this to get a Virtual Switch Extended Options List that is set on the currently managed Virtual Switch. Virtual Switch Extended Option enables you to configure more detail settings of the Virtual Switch. By default, both iQuila Server's global administrators and individual Virtual Switch's administrators can modify the Virtual Switch Extended Options. However, if the deny_Switch_admin_change_ext_option is set to 1 on the Virtual Switch Admin Options, the individual Virtual Switch's administrators cannot modify the Virtual Switch Extended Options. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster member.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSwitchExtOptions",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminOptionList": [
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch name
AdminOptionList	Array object	List data
Name_str	string (ASCII)	Name
Value_u32	number (uint32)	Data
Descrption_utf	string (UTF8)	Descrption

Set a Value of Virtual Switch Extended Options

Description

Set a Value of Virtual Switch Extended Options. Use this to set a value in the Virtual Switch Extended Options List that is set on the currently managed Virtual Switch. Virtual Switch Extended Option enables you to configure more detail settings of the Virtual Switch. By default, both iQuila Server's global administrators and individual Virtual Switch's administrators can modify the Virtual Switch Extended Options. However, if the deny_Switch_admin_change_ext_option is set to 1 on the Virtual Switch Admin Options, the individual Virtual Switch's administrators cannot modify the Virtual Switch Extended Options. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster member.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSwitchExtOptions",
  "params": {
    "HubName_str": "Switchname",
    "AdminOptionList": [
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      }
    ]
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "AdminOptionList": [
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      },
      {
        "Name_str": "name",
        "Value_u32": 0,
        "Descrption_utf": "descrption"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	Virtual Switch name
AdminOptionList	Array object	List data
Name_str	string (ASCII)	Name
Value_u32	number (uint32)	Data
Descrption_utf	string (UTF8)	Descrption

Define New Virtual Layer 3 Switch

Description

Define New Virtual Layer 3 Switch. Use this to define a new Virtual Layer 3 Switch on the iQuila Server. To call this API. Also, this API does not operate on iQuila Bridge. [Explanation on Virtual Layer 3 Switch Function] You can define Virtual Layer 3 Switches between multiple Virtual Switchs operating on this iQuila Server and configure routing between different IP networks. [Caution about the Virtual Layer 3 Switch Function] The Virtual Layer 3 Switch functions are provided for network administrators and other people who know a lot about networks and IP routing. If you are using the regular VEN functions, you do not need to use the Virtual Layer 3 Switch functions. If the Virtual Layer 3 Switch functions are to be used, the person who configures them must have sufficient knowledge of IP routing and be perfectly capable of not impacting the network.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddL3Switch",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	Layer-3 Switch name

Delete Virtual Layer 3 Switch

Description

Delete Virtual Layer 3 Switch. Use this to delete an existing Virtual Layer 3 Switch that is defined on the iQuila Server. When the specified Virtual Layer 3 Switch is operating, it will be automatically deleted after operation stops. To get a list of existing Virtual Layer 3 Switches, use the EnumL3Switch API. To call this API. Also, this API does not operate on iQuila Bridge.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DelL3Switch",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	Layer-3 Switch name

Get List of Virtual Layer 3 Switches

Description

Get List of Virtual Layer 3 Switches. Use this to define a new Virtual Layer 3 Switch on the iQuila Server. To call this API. Also, this API does not operate on iQuila Bridge. [Explanation on Virtual Layer 3 Switch Function] You can define Virtual Layer 3 Switches between multiple Virtual Switchs operating on this iQuila Server and configure routing between different IP networks. [Caution about the Virtual Layer 3 Switch Function] The Virtual Layer 3 Switch functions are provided for network administrators and other people who know a lot about networks and IP routing. If you are using the regular VEN functions, you do not need to use the Virtual Layer 3 Switch functions. If the Virtual Layer 3 Switch functions are to be used, the person who configures them must have sufficient knowledge of IP routing and be perfectly capable of not impacting the network.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumL3Switch",
  "params": {}
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "L3SWList": [
      {
        "Name_str": "name",
        "NumInterfaces_u32": 0,
        "NumTables_u32": 0,
        "Active_qool": false,
        "Online_qool": false
      },
      {
        "Name_str": "name",
        "NumInterfaces_u32": 0,
        "NumTables_u32": 0,
        "Active_qool": false,
        "Online_qool": false
      },
      {
        "Name_str": "name",
        "NumInterfaces_u32": 0,
        "NumTables_u32": 0,
        "Active_qool": false,
        "Online_qool": false
      }
    ]
  }
}
```


Parameters

Name	Type	Description
L3SWList	Array object	Layer-3 switch list
Name_str	string (ASCII)	Name of the layer-3 switch
NumInterfaces_u32	number (uint32)	Number of layer-3 switch virtual interfaces
NumTables_u32	number (uint32)	Number of routing tables
Active_qool	qoolean	Activated flag
Online_qool	qoolean	Online flag

Start Virtual Layer 3 Switch Operation

Description

Start Virtual Layer 3 Switch Operation. Use this to start the operation of an existing Virtual Layer 3 Switch defined on the iQuila Server whose operation is currently stopped. To get a list of existing Virtual Layer 3 Switches, use the EnumL3Switch API. To call this API. Also, this API does not operate on iQuila Bridge. [Explanation on Virtual Layer 3 Switch Function] You can define Virtual Layer 3 Switches between multiple Virtual Switchs operating on this iQuila Server and configure routing between different IP networks. [Caution about the Virtual Layer 3 Switch Function] The Virtual Layer 3 Switch functions are provided for network administrators and other people who know a lot about networks and IP routing. If you are using the regular VEN functions, you do not need to use the Virtual Layer 3 Switch functions. If the Virtual Layer 3 Switch functions are to be used, the person who configures them must have sufficient knowledge of IP routing and be perfectly capable of not impacting the network.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "StartL3Switch",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	Layer-3 Switch name

Stop Virtual Layer 3 Switch Operation

Description

Stop Virtual Layer 3 Switch Operation. Use this to stop the operation of an existing Virtual Layer 3 Switch defined on the iQuila Server whose operation is currently operating. To get a list of existing Virtual Layer 3 Switches, use the EnumL3Switch API. To call this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "StopL3Switch",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name"
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	Layer-3 Switch name

Add Virtual Interface to Virtual Layer 3 Switch

Description

Add Virtual Interface to Virtual Layer 3 Switch. Use this to add to a specified Virtual Layer 3 Switch, a virtual interface that connects to a Virtual Switch operating on the same iQuila Server. You can define multiple virtual interfaces and routing tables for a single Virtual Layer 3 Switch. A virtual interface is associated to a Virtual Switch and operates as a single IP host on the Virtual Switch when that Virtual Switch is operating. When multiple virtual interfaces that respectively belong to a different IP network of a different Virtual Switch are defined, IP routing will be automatically performed between these interfaces. You must define the IP network space that the virtual interface belongs to and the IP address of the interface itself. Also, you must specify the name of the Virtual Switch that the interface will connect to. You can specify a Virtual Switch that currently doesn't exist for the Virtual Switch name. The virtual interface must have one IP address in the Virtual Switch. You also must specify the subnet mask of an IP network that the IP address belongs to. Routing via the Virtual Layer 3 Switches of IP spaces of multiple Virtual Switchs operates based on the IP address is specified here. To call this API. Also, this API does not operate on iQuila Bridge. To execute this API, the target Virtual Layer 3 Switch must be stopped. If it is not stopped, first use the StopL3Switch API to stop it and then execute this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddL3If",
  "params": {
    "Name_str": "name",
    "HubName_str": "Switchname",
    "IpAddress_ip": "10.0.0.1",
    "SubnetMask_ip": "255.255.255.255"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name",
    "HubName_str": "Switchname",
    "IpAddress_ip": "10.0.0.1",
    "SubnetMask_ip": "255.255.255.255"
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	L3 switch name
HubName_str	string (ASCII)	Virtual Switch name
IpAddress_ip	string (IP address)	IP address
SubnetMask_ip	string (IP address)	Subnet mask

Delete Virtual Interface of Virtual Layer 3 Switch

Description

Delete Virtual Interface of Virtual Layer 3 Switch. Use this to delete a virtual interface already defined in the specified Virtual Layer 3 Switch. You can get a list of the virtual interfaces currently defined, by using the EnumL3If API. To call this API. Also, this API does not operate on iQuila Bridge. To execute this API, the target Virtual Layer 3 Switch must be stopped. If it is not stopped, first use the StopL3Switch API to stop it and then execute this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DelL3If",
  "params": {
    "Name_str": "name",
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name",
    "HubName_str": "Switchname",
    "IpAddress_ip": "10.0.0.1",
    "SubnetMask_ip": "255.255.255.255"
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	L3 switch name
HubName_str	string (ASCII)	Virtual Switch name
IpAddress_ip	string (IP address)	IP address
SubnetMask_ip	string (IP address)	Subnet mask

Get List of Interfaces Registered on the Virtual Layer 3 Switch

Description

Get List of Interfaces Registered on the Virtual Layer 3 Switch. Use this to get a list of virtual interfaces when virtual interfaces have been defined on a specified Virtual Layer 3 Switch. You can define multiple virtual interfaces and routing tables for a single Virtual Layer 3 Switch. A virtual interface is associated to a Virtual Switch and operates as a single IP host on the Virtual Switch when that Virtual Switch is operating. When multiple virtual interfaces that respectively belong to a different IP network of a different Virtual Switch are defined, IP routing will be automatically performed between these interfaces. To call this API. Also, this API does not operate on iQuila Bridge.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumL3If",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name",
    "L3IFList": [
      {
        "Name_str": "name",
        "HubName_str": "Switchname",
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Name_str": "name",
        "HubName_str": "Switchname",
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Name_str": "name",
        "HubName_str": "Switchname",
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	Layer-3 switch name
L3IFList	Array object	Layer-3 interface list
Name_str	string (ASCII)	L3 switch name
HubName_str	string (ASCII)	Virtual Switch name
IpAddress_ip	string (IP address)	IP address
SubnetMask_ip	string (IP address)	Subnet mask

Add Routing Table Entry for Virtual Layer 3 Switch

Description

Add Routing Table Entry for Virtual Layer 3 Switch. Here you can add a new routing table entry to the routing table of the specified Virtual Layer 3 Switch. If the destination IP address of the IP packet does not belong to any IP network that belongs to a virtual interface, the IP routing engine of the Virtual Layer 3 Switch will reference the routing table and execute routing. You must specify the contents of the routing table entry to be added to the Virtual Layer 3 Switch. You must specify any IP address that belongs to the same IP network in the virtual interface of this Virtual Layer 3 Switch as the gateway address. To call this API. Also, this API does not operate on iQuila Bridge. To execute this API, the target Virtual Layer 3 Switch must be stopped. If it is not stopped, first use the StopL3Switch API to stop it and then execute this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddL3Table",
  "params": {
    "Name_str": "name",
    "NetworkAddress_ip": "10.0.0.1",
    "SubnetMask_ip": "255.255.255.255",
    "GatewayAddress_ip": "10.0.0.1",
    "Metric_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name",
    "NetworkAddress_ip": "10.0.0.1",
    "SubnetMask_ip": "255.255.255.255",
    "GatewayAddress_ip": "10.0.0.1",
    "Metric_u32": 0
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	L3 switch name
NetworkAddress_ip	string (IP address)	Network address
SubnetMask_ip	string (IP address)	Subnet mask
GatewayAddress_ip	string (IP address)	Gateway address
Metric_u32	number (uint32)	Metric

Delete Routing Table Entry of Virtual Layer 3 Switch

Description

Delete Routing Table Entry of Virtual Layer 3 Switch. Use this to delete a routing table entry that is defined in the specified Virtual Layer 3 Switch. You can get a list of the already defined routing table entries by using the EnumL3Table API. To call this API. Also, this API does not operate on iQuila Bridge. To execute this API, the target Virtual Layer 3 Switch must be stopped. If it is not stopped, first use the StopL3Switch API to stop it and then execute this API.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DelL3Table",
  "params": {
    "Name_str": "name",
    "NetworkAddress_ip": "10.0.0.1",
    "SubnetMask_ip": "255.255.255.255",
    "GatewayAddress_ip": "10.0.0.1",
    "Metric_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name",
    "NetworkAddress_ip": "10.0.0.1",
    "SubnetMask_ip": "255.255.255.255",
    "GatewayAddress_ip": "10.0.0.1",
    "Metric_u32": 0
  }
}
```

Parameters

Name	Type	Description
Name_str	string (ASCII)	L3 switch name
NetworkAddress_ip	string (IP address)	Network address
SubnetMask_ip	string (IP address)	Subnet mask
GatewayAddress_ip	string (IP address)	Gateway address
Metric_u32	number (uint32)	Metric

Get List of Routing Tables of Virtual Layer 3 Switch

Description

Get List of Routing Tables of Virtual Layer 3 Switch. Use this to get a list of routing tables when routing tables have been defined on a specified Virtual Layer 3 Switch. If the destination IP address of the IP packet does not belong to any IP network that belongs to a virtual interface, the IP routing engine of the Virtual Layer 3 Switch will reference this routing table and execute routing. To call this API. Also, this API does not operate on iQuila Bridge.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumL3Table",
  "params": {
    "Name_str": "name"
  }
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Name_str": "name",
    "L3Table": [
      {
        "Name_str": "name",
        "NetworkAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255",
        "GatewayAddress_ip": "10.0.0.1",
        "Metric_u32": 0
      },
      {
        "Name_str": "name",
        "NetworkAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255",
        "GatewayAddress_ip": "10.0.0.1",
        "Metric_u32": 0
      },
      {
        "Name_str": "name",
        "NetworkAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255",
        "GatewayAddress_ip": "10.0.0.1",
        "Metric_u32": 0
      }
    ]
  }
}
```


Parameters

Name	Type	Description
Name_str	string (ASCII)	L3 switch name
L3Table	Array object	Routing table item list
Name_str	string (ASCII)	L3 switch name
NetworkAddress_ip	string (IP address)	Network address
SubnetMask_ip	string (IP address)	Subnet mask
GatewayAddress_ip	string (IP address)	Gateway address
Metric_u32	number (uint32)	Metric

Get List of Certificates Revocation List

Description

Get List of Certificates Revocation List. Use this to get a Certificates Revocation List that is set on the currently managed Virtual Switch. By registering certificates in the Certificates Revocation List, the clients who provide these certificates will be unable to connect to this Virtual Switch using certificate authentication mode. Normally with this function, in cases where the security of a private key has been compromised or where a person holding a certificate has been stripped of their privileges, by registering that certificate as invalid on the Virtual Switch, it is possible to deny user authentication when that certificate is used by a client to connect to the Virtual Switch. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumCrl",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "CRLList": [
      {
        "Key_u32": 0,
        "CrlInfo_utf": "crlinfo"
      },
      {
        "Key_u32": 0,
        "CrlInfo_utf": "crlinfo"
      },
      {
        "Key_u32": 0,
        "CrlInfo_utf": "crlinfo"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
CRLList	Array object	CRL list
Key_u32	number (uint32)	Key ID
CrlInfo_utf	string (UTF8)	The contents of the CRL item

Add a Revoked Certificate

Description

Add a Revoked Certificate. Use this to add a new revoked certificate definition in the Certificate Revocation List that is set on the currently managed Virtual Switch. Specify the contents to be registered in the Certificate Revocation List by using the parameters of this API. When a user connects to a Virtual Switch in certificate authentication mode and that certificate matches 1 or more of the contents registered in the certificates revocation list, the user is denied connection. A certificate that matches all the conditions that are defined by the parameters specified by this API will be judged as invalid. The items that can be set are as follows: Name (CN), Organization (O), Organization Unit (OU), Country (C), State (ST), Locale (L), Serial Number (hexadecimal), MD5 Digest Value (hexadecimal, 128 bit), and SHA-1 Digest Value (hexadecimal, 160 bit). For the specification of a digest value (hash value) a certificate is optionally specified depending on the circumstances. Normally when a MD5 or SHA-1 digest value is input, it is not necessary to input the other items. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddCrl",
  "params": {
    "HubName_str": "Switchname",
    "CommonName_utf": "commonname",
    "Organization_utf": "organization",
    "Unit_utf": "unit",
    "Country_utf": "country",
    "State_utf": "state",
    "Local_utf": "local",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "DigestMD5_bin": "SGVsbG8gV29ybGQ=",
    "DigestSHA1_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0,
    "CommonName_utf": "commonname",
    "Organization_utf": "organization",
    "Unit_utf": "unit",
    "Country_utf": "country",
    "State_utf": "state",
    "Local_utf": "local",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "DigestMD5_bin": "SGVsbG8gV29ybGQ=",
    "DigestSHA1_bin": "SGVsbG8gV29ybGQ="
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	Key ID
CommonName_utf	string (UTF8)	CN, optional
Organization_utf	string (UTF8)	O, optional
Unit_utf	string (UTF8)	OU, optional
Country_utf	string (UTF8)	C, optional
State_utf	string (UTF8)	ST, optional
Local_utf	string (UTF8)	L, optional
Serial_bin	string (Base64 binary)	Serial, optional
DigestMD5_bin	string (Base64 binary)	MD5 Digest, optional
DigestSHA1_bin	string (Base64 binary)	SHA1 Digest, optional

Delete a Revoked Certificate

Description

Delete a Revoked Certificate. Use this to specify and delete a revoked certificate definition from the certificate revocation list that is set on the currently managed Virtual Switch. To get the list of currently registered revoked certificate definitions, use the EnumCrl API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DelCrl",
  "params": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0,
    "CommonName_utf": "commonname",
    "Organization_utf": "organization",
    "Unit_utf": "unit",
    "Country_utf": "country",
    "State_utf": "state",
    "Local_utf": "local",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "DigestMD5_bin": "SGVsbG8gV29ybGQ=",
    "DigestSHA1_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	Key ID
CommonName_utf	string (UTF8)	CN, optional
Organization_utf	string (UTF8)	O, optional
Unit_utf	string (UTF8)	OU, optional
Country_utf	string (UTF8)	C, optional
State_utf	string (UTF8)	ST, optional
Local_utf	string (UTF8)	L, optional
Serial_bin	string (Base64 binary)	Serial, optional
DigestMD5_bin	string (Base64 binary)	MD5 Digest, optional
DigestSHA1_bin	string (Base64 binary)	SHA1 Digest, optional

Get a Revoked Certificate

Description

Get a Revoked Certificate. Use this to specify and get the contents of a revoked certificate definition from the Certificates Revocation List that is set on the currently managed Virtual Switch. To get the list of currently registered revoked certificate definitions, use the EnumCrl API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetCrl",
  "params": {
    "HubName_str": "Switchname",
    "Key_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0,
    "CommonName_utf": "commonname",
    "Organization_utf": "organization",
    "Unit_utf": "unit",
    "Country_utf": "country",
    "State_utf": "state",
    "Local_utf": "local",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "DigestMD5_bin": "SGVsbG8gV29ybGQ=",
    "DigestSHA1_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	Key ID
CommonName_utf	string (UTF8)	CN, optional
Organization_utf	string (UTF8)	O, optional
Unit_utf	string (UTF8)	OU, optional
Country_utf	string (UTF8)	C, optional
State_utf	string (UTF8)	ST, optional
Local_utf	string (UTF8)	L, optional
Serial_bin	string (Base64 binary)	Serial, optional
DigestMD5_bin	string (Base64 binary)	MD5 Digest, optional
DigestSHA1_bin	string (Base64 binary)	SHA1 Digest, optional

Change Existing CRL (Certificate Revocation List) Entry

Description

Change Existing CRL (Certificate Revocation List) Entry. Use this to alter an existing revoked certificate definition in the Certificate Revocation List that is set on the currently managed Virtual Switch. Specify the contents to be registered in the Certificate Revocation List by using the parameters of this API. When a user connects to a Virtual Switch in certificate authentication mode and that certificate matches 1 or more of the contents registered in the certificates revocation list, the user is denied connection. A certificate that matches all the conditions that are defined by the parameters specified by this API will be judged as invalid. The items that can be set are as follows: Name (CN), Organization (O), Organization Unit (OU), Country (C), State (ST), Locale (L), Serial Number (hexadecimal), MD5 Digest Value (hexadecimal, 128 bit), and SHA-1 Digest Value (hexadecimal, 160 bit). For the specification of a digest value (hash value) a certificate is optionally specified depending on the circumstances. Normally when a MD5 or SHA-1 digest value is input, it is not necessary to input the other items. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetCrl",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Key_u32": 0,
    "CommonName_utf": "commonname",
    "Organization_utf": "organization",
    "Unit_utf": "unit",
    "Country_utf": "country",
    "State_utf": "state",
    "Local_utf": "local",
    "Serial_bin": "SGVsbG8gV29ybGQ=",
    "DigestMD5_bin": "SGVsbG8gV29ybGQ=",
    "DigestSHA1_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Key_u32	number (uint32)	Key ID
CommonName_utf	string (UTF8)	CN, optional
Organization_utf	string (UTF8)	O, optional
Unit_utf	string (UTF8)	OU, optional
Country_utf	string (UTF8)	C, optional
State_utf	string (UTF8)	ST, optional
Local_utf	string (UTF8)	L, optional
Serial_bin	string (Base64 binary)	Serial, optional
DigestMD5_bin	string (Base64 binary)	MD5 Digest, optional
DigestSHA1_bin	string (Base64 binary)	SHA1 Digest, optional

Add Rule to Source IP Address Limit List

Description

Add Rule to Source IP Address Limit List. Use this to add a new rule to the Source IP Address Limit List that is set on the currently managed Virtual Switch. The items set here will be used to decide whether to allow or deny connection from a iQuila Client when this client attempts connection to the Virtual Switch. You can specify a client IP address, or IP address or mask to match the rule as the contents of the rule item. By specifying an IP address only, there will only be one specified server that will match the rule, but by specifying an IP net mask address or subnet mask address, all the servers in the range of that subnet will match the rule. You can specify the priority for the rule. You can specify an integer of 1 or greater for the priority and the smaller the number, the higher the priority. To get a list of the currently registered Source IP Address Limit List, use the GetAcList API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetAcList",
  "params": {
    "HubName_str": "Switchname",
    "ACList": [
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      }
    ]
  }
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "ACList": [
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
ACList	Array object	Source IP Address Limit List
Id_u32	number (uint32)	ID
Priority_u32	number (uint32)	Priority
Deny_qool	qoolean	Deny access
Masked_qool	qoolean	Set true if you want to specify the SubnetMask_ip item.
IpAddress_ip	string (IP address)	IP address
SubnetMask_ip	string (IP address)	Subnet mask, valid only if Masked_qool == true

Get List of Rule Items of Source IP Address Limit List

Description

Get List of Rule Items of Source IP Address Limit List. Use this to get a list of Source IP Address Limit List rules that is set on the currently managed Virtual Switch. You can allow or deny VEN connections to this Virtual Switch according to the client server's source IP address. You can define multiple rules and set a priority for each rule. The search proceeds from the rule with the highest order or priority and based on the action of the rule that the IP address first matches, the connection from the client is either allowed or denied. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetAcList",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format


```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "ACList": [
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      },
      {
        "Id_u32": 0,
        "Priority_u32": 0,
        "Deny_qool": false,
        "Masked_qool": false,
        "IpAddress_ip": "10.0.0.1",
        "SubnetMask_ip": "255.255.255.255"
      }
    ]
  }
}
```


Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
ACList	Array object	Source IP Address Limit List
Id_u32	number (uint32)	ID
Priority_u32	number (uint32)	Priority
Deny_qool	qoolean	Deny access
Masked_qool	qoolean	Set true if you want to specify the SubnetMask_ip item.
IpAddress_ip	string (IP address)	IP address
SubnetMask_ip	string (IP address)	Subnet mask, valid only if Masked_qool == true

Get List of Log Files

Description

Get List of Log Files. Use this to display a list of log files outputted by the iQuila Server that have been saved on the iQuila Server server. By specifying a log file file name displayed here and calling it using the ReadLogFile API you can download the contents of the log file. If you are connected to the iQuila Server in server admin mode, you can display or download the packet logs and security logs of all Virtual Switchs and the server log of the iQuila Server. When connected in Virtual Switch Admin Mode, you are able to view or download only the packet log and security log of the Virtual Switch that is the target of management.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumLogFile",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "LogFiles": [
      {
        "ServerName_str": "servername",
        "FilePath_str": "filepath",
        "FileSize_u32": 0,
        "UpdatedTime_dt": "2021-01-01T12:21:22.123"
      },
      {
        "ServerName_str": "servername",
        "FilePath_str": "filepath",
        "FileSize_u32": 0,
        "UpdatedTime_dt": "2021-01-01T12:21:22.123"
      },
      {
        "ServerName_str": "servername",
        "FilePath_str": "filepath",
        "FileSize_u32": 0,
        "UpdatedTime_dt": "2021-01-01T12:21:22.123"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
LogFiles	Array object	Log file list
ServerName_str	string (ASCII)	Server name
FilePath_str	string (ASCII)	File path
FileSize_u32	number (uint32)	File size
UpdatedTime_dt	Date	Last write date

Download a part of Log File

Description

Download a part of Log File. Use this to download the log file that is saved on the iQuila Server server. To download the log file first get the list of log files using the EnumLogFile API and then download the log file using the ReadLogFile API. If you are connected to the iQuila Server in server admin mode, you can display or download the packet logs and security logs of all Virtual Switchs and the server log of the iQuila Server. When connected in Virtual Switch Admin Mode, you are able to view or download only the packet log and security log of the Virtual Switch that is the target of management.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "ReadLogFile",
  "params": {
    "FilePath_str": "filepath"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "ServerName_str": "servername",
    "FilePath_str": "filepath",
    "Offset_u32": 0,
    "Buffer_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
ServerName_str	string (ASCII)	Server name
FilePath_str	string (ASCII)	File Path
Offset_u32	number (uint32)	Offset to download. You have to call the ReadLogFile API multiple times to download the entire log file with requesting a part of the file by specifying the Offset_u32 field.
Buffer_bin	string (Base64 binary)	Received buffer

Set syslog Send Function

Description

Set syslog Send Function. Use this to set the usage of syslog send function and which syslog server to use.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSysLog",
  "params": {
    "SaveType_u32": 0,
    "Hostname_str": "hostname",
    "Port_u32": 0
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "SaveType_u32": 0,
    "Hostname_str": "hostname",
    "Port_u32": 0
  }
}
```

Parameters

Name	Type	Description
SaveType_u32	number (enum)	The behavior of the syslog function Values: 0: Do not use syslog 1: Only server log 2: Server and Virtual Switch security log 3: Server, Virtual Switch security, and packet log
Hostname_str	string (ASCII)	Specify the host name or IP address of the syslog server
Port_u32	number (uint32)	Specify the port number of the syslog server

Get syslog Send Function

Description

Get syslog Send Function. This allows you to get the current setting contents of the syslog send function. You can get the usage setting of the syslog function and the host name and port number of the syslog server to use.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSysLog",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "SaveType_u32": 0,
    "Hostname_str": "hostname",
    "Port_u32": 0
  }
}
```

Parameters

Name	Type	Description
SaveType_u32	number (enum)	The behavior of the syslog function Values: 0: Do not use syslog 1: Only server log 2: Server and Virtual Switch security log 3: Server, Virtual Switch security, and packet log
Hostname_str	string (ASCII)	Specify the host name or IP address of the syslog server
Port_u32	number (uint32)	Specify the port number of the syslog server

Set Today's Message of Virtual Switch

Description

Set Today's Message of Virtual Switch. The message will be displayed on iQuila Client UI when a user will establish a connection to the Virtual Switch.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetSwitchMsg",
  "params": {
    "HubName_str": "Switchname",
    "Msg_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Msg_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Msg_bin	string (Base64 binary)	Message (Unicode strings acceptable)

Get Today's Message of Virtual Switch

Description

Get Today's Message of Virtual Switch. The message will be displayed on iQuila Client UI when a user will establish a connection to the Virtual Switch.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetSwitchMsg",
  "params": {
    "HubName_str": "Switchname"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Msg_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Msg_bin	string (Base64 binary)	Message (Unicode strings acceptable)

Raise a vital error on the iQuila Server / Bridge to terminate the process forcefully

Description

Raise a vital error on the iQuila Server / Bridge to terminate the process forcefully. This API will raise a fatal error (memory access violation) on the iQuila Server / Bridge running process in order to crash the process. As the result, iQuila Server / Bridge will be terminated and restarted if it is running as a service mode. If the iQuila Server is running as a user mode, the process will not automatically restarted. This API is for a situation when the iQuila Server / Bridge is under a non-recoverable error or the process is in an infinite loop. This API will disconnect all VEN Sessions on the iQuila Server / Bridge. All unsaved settings in the memory of iQuila Server / Bridge will be lost. Before run this API, call the Flush API to try to save volatile data to the configuration file. To execute this API, you must have iQuila Server / iQuila Bridge administrator privileges.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "Crash",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "IntValue_u32": 0,
    "Int64Value_u64": 0,
    "StrValue_str": "strvalue",
    "UniStrValue_utf": "unistrvalue"
  }
}
```

Parameters

Name	Type	Description
IntValue_u32	number (uint32)	A 32-bit integer field
Int64Value_u64	number (uint64)	A 64-bit integer field
StrValue_str	string (ASCII)	An Ascii string field
UniStrValue_utf	string (UTF8)	An UTF-8 string field

Get the message for administrators

Description

Get the message for administrators.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetAdminMsg",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "HubName_str": "Switchname",
    "Msg_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
HubName_str	string (ASCII)	The Virtual Switch name
Msg_bin	string (Base64 binary)	Message (Unicode strings acceptable)

Save All Volatile Data of iQuila Server / Bridge to the Configuration File

Description

Save All Volatile Data of iQuila Server / Bridge to the Configuration File. The number of configuration file bytes will be returned as the "IntValue" parameter. Normally, the iQuila Server / iQuila Bridge retains the volatile configuration data in memory. It is flushed to the disk as vpn_server.config or vpn_bridge.config periodically. The period is 300 seconds (5 minutes) by default. (The period can be altered by modifying the AutoSaveConfigSpan item in the configuration file.) The data will be saved on the timing of shutting down normally of the iQuila Server / Bridge. Execute the Flush API to make the iQuila Server / Bridge save the settings to the file immediately. The setting data will be stored on the disk drive of the server server. Use the Flush API in a situation that you do not have an enough time to shut down the server process normally. To call this API. To execute this API, you must have iQuila Server / iQuila Bridge administrator privileges.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "Flush",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "IntValue_u32": 0,
    "Int64Value_u64": 0,
    "StrValue_str": "strvalue",
    "UniStrValue_utf": "unistrvalue"
  }
}
```

Parameters

Name	Type	Description
IntValue_u32	number (uint32)	A 32-bit integer field
Int64Value_u64	number (uint64)	A 64-bit integer field
StrValue_str	string (ASCII)	An Ascii string field
UniStrValue_utf	string (UTF8)	An UTF-8 string field

Enable or Disable IPsec iQuila Server Function

Description

Enable or Disable IPsec iQuila Server Function. Enable or Disable IPsec iQuila Server Function on the iQuila Server. If you enable this function, Virtual Switchs on the iQuila Server will be able to accept Remote-Access VEN connections from L2TP-compatible PCs, Mac OS X and Smartphones, and also can accept EtherIP Site-to-Site VEN Connection. VEN Connections from Smartphones suchlike iPhone, iPad and Android, and also from native iQuila Clients on Mac OS X and Windows can be accepted. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetIPsecServices",
  "params": {
    "L2TP_Raw_qool": false,
    "L2TP_IPsec_qool": false,
    "EtherIP_IPsec_qool": false,
    "IPsec_Secret_str": "ipsec_secret",
    "L2TP_DefaultSwitch_str": "l2tp_defaultSwitch"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "L2TP_Raw_qool": false,
    "L2TP_IPsec_qool": false,
    "EtherIP_IPsec_qool": false,
    "IPsec_Secret_str": "ipsec_secret",
    "L2TP_DefaultSwitch_str": "l2tp_defaultSwitch"
  }
}
```

Parameters

Name	Type	Description
L2TP_Raw_qool	qoolean	Enable or Disable the L2TP Server Function (Raw L2TP with No Encryptions). To accept special VEN clients, enable this option.
L2TP_IPsec_qool	qoolean	Enable or Disable the L2TP over IPsec Server Function. To accept VEN connections from iPhone, iPad, Android, Windows or Mac OS X, enable this option.
EtherIP_IPsec_qool	qoolean	Enable or Disable the EtherIP / L2TPv3 over IPsec Server Function (for site-to-site iQuila Server function). Router Products which are compatible with EtherIP over IPsec can connect to Virtual Switchs on the iQuila Server and establish Layer-2 (Ethernet) Bridging.
IPsec_Secret_str	string (ASCII)	Specify the IPsec Pre-Shared Key. An IPsec Pre-Shared Key is also called as "PSK" or "secret". Specify it equal or less than 8 letters, and distribute it to every users who will connect to the iQuila Server. Please note: Google Android 4.0 has a bug which a Pre-Shared Key with 10 or more letters causes a unexpected behavior. For that reason, the letters of a Pre-Shared Key should be 9 or less characters.
L2TP_DefaultSwitch_str	string (ASCII)	Specify the default Virtual Switch in a case of omitting the name of Switch on the Username. Users should specify their username such as "Username@Target Virtual Switch Name" to connect this L2TP Server. If the designation of the Virtual Switch is omitted, the above Switch will be used as the target.

Get the Current IPsec iQuila Server Settings

Description

Get the Current IPsec iQuila Server Settings. Get and view the current IPsec iQuila Server settings on the iQuila Server. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetIPsecServices",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "L2TP_Raw_qool": false,
    "L2TP_IPsec_qool": false,
    "EtherIP_IPsec_qool": false,
    "IPsec_Secret_str": "ipsec_secret",
    "L2TP_DefaultSwitch_str": "l2tp_defaultSwitch"
  }
}
```

Parameters

Name	Type	Description
L2TP_Raw_qool	qoolean	Enable or Disable the L2TP Server Function (Raw L2TP with No Encryptions). To accept special VEN clients, enable this option.
L2TP_IPsec_qool	qoolean	Enable or Disable the L2TP over IPsec Server Function. To accept VEN connections from iPhone, iPad, Android, Windows or Mac OS X, enable this option.
EtherIP_IPsec_qool	qoolean	Enable or Disable the EtherIP / L2TPv3 over IPsec Server Function (for site-to-site iQuila Server function). Router Products which are compatible with EtherIP over IPsec can connect to Virtual Switchs on the iQuila Server and establish Layer-2 (Ethernet) Bridging.
IPsec_Secret_str	string (ASCII)	Specify the IPsec Pre-Shared Key. An IPsec Pre-Shared Key is also called as "PSK" or "secret". Specify it equal or less than 8 letters, and distribute it to every users who will connect to the iQuila Server. Please note: Google Android 4.0 has a bug which a Pre-Shared Key with 10 or more letters causes a unexpected behavior. For that reason, the letters of a Pre-Shared Key should be 9 or less characters.
L2TP_DefaultSwitch_str	string (ASCII)	Specify the default Virtual Switch in a case of omitting the name of Switch on the Username. Users should specify their username such as "Username@Target Virtual Switch Name" to connect this L2TP Server. If the designation of the Virtual Switch is omitted, the above Switch will be used as the target.

Add New EtherIP / L2TPv3 over IPsec Client Setting to Accept EthreIP / L2TPv3 Client Devices

Description

Add New EtherIP / L2TPv3 over IPsec Client Setting to Accept EthreIP / L2TPv3 Client Devices. Add a new setting entry to enable the EtherIP / L2TPv3 over IPsec Server Function to accept client devices. In order to accept connections from routers by the EtherIP / L2TPv3 over IPsec Server Function, you have to define the relation table between an IPsec Phase 1 string which is presented by client devices of EtherIP / L2TPv3 over IPsec compatible router, and the designation of the destination Virtual Switch. After you add a definition entry by AddEtherIpId API, the defined connection setting to the Virtual Switch will be applied on the login-attepting session from an EtherIP / L2TPv3 over IPsec client device. The username and password in an entry must be registered on the Virtual Switch. An EtherIP / L2TPv3 client will be regarded as it connected the Virtual Switch with the identification of the above user information. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "AddEtherIpId",
  "params": {
    "Id_str": "id",
    "HubName_str": "Switchname",
    "UserName_str": "username",
    "Password_str": "password"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Id_str": "id",
    "HubName_str": "Switchname",
    "UserName_str": "username",
    "Password_str": "password"
  }
}
```

Parameters

Name	Type	Description
Id_str	string (ASCII)	Specify an ISAKMP Phase 1 ID. The ID must be exactly same as a ID in the configuration of the EtherIP / L2TPv3 Client. You can specify IP address as well as characters as ID, if the EtherIP Client uses IP address as Phase 1 ID. If you specify '*' (asterisk), it will be a wildcard to match any clients which doesn't match other explicit rules.
HubName_str	string (ASCII)	Specify the name of the Virtual Switch to connect.
UserName_str	string (ASCII)	Specify the username to login to the destination Virtual Switch.
Password_str	string (ASCII)	Specify the password to login to the destination Virtual Switch.

Get the Current List of EtherIP / L2TPv3 Client Device Entry Definitions

Description

Get the Current List of EtherIP / L2TPv3 Client Device Entry Definitions. This API gets and shows the list of entries to accept VEN clients by EtherIP / L2TPv3 over IPsec Function. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetEtherIpId",
  "params": {
    "Id_str": "id"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Id_str": "id",
    "HubName_str": "Switchname",
    "UserName_str": "username",
    "Password_str": "password"
  }
}
```

Parameters

Name	Type	Description
Id_str	string (ASCII)	Specify an ISAKMP Phase 1 ID. The ID must be exactly same as a ID in the configuration of the EtherIP / L2TPv3 Client. You can specify IP address as well as characters as ID, if the EtherIP Client uses IP address as Phase 1 ID. If you specify '*' (asterisk), it will be a wildcard to match any clients which doesn't match other explicit rules.
HubName_str	string (ASCII)	Specify the name of the Virtual Switch to connect.
UserName_str	string (ASCII)	Specify the username to login to the destination Virtual Switch.
Password_str	string (ASCII)	Specify the password to login to the destination Virtual Switch.

Delete an EtherIP / L2TPv3 over IPsec Client Setting

Description

Delete an EtherIP / L2TPv3 over IPsec Client Setting. This API deletes an entry to accept VEN clients by EtherIP / L2TPv3 over IPsec Function. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "DeleteEtherIpId",
  "params": {
    "Id_str": "id"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Id_str": "id",
    "HubName_str": "Switchname",
    "UserName_str": "username",
    "Password_str": "password"
  }
}
```

Parameters

Name	Type	Description
Id_str	string (ASCII)	Specify an ISAKMP Phase 1 ID. The ID must be exactly same as a ID in the configuration of the EtherIP / L2TPv3 Client. You can specify IP address as well as characters as ID, if the EtherIP Client uses IP address as Phase 1 ID. If you specify '*' (asterisk), it will be a wildcard to match any clients which doesn't match other explicit rules.
HubName_str	string (ASCII)	Specify the name of the Virtual Switch to connect.
UserName_str	string (ASCII)	Specify the username to login to the destination Virtual Switch.
Password_str	string (ASCII)	Specify the password to login to the destination Virtual Switch.

Get the Current List of EtherIP / L2TPv3 Client Device Entry Definitions

Description

Get the Current List of EtherIP / L2TPv3 Client Device Entry Definitions. This API gets and shows the list of entries to accept VEN clients by EtherIP / L2TPv3 over IPsec Function. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "EnumEtherIpId",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "Settings": [
      {
        "Id_str": "id",
        "HubName_str": "Switchname",
        "UserName_str": "username",
        "Password_str": "password"
      },
      {
        "Id_str": "id",
        "HubName_str": "Switchname",
        "UserName_str": "username",
        "Password_str": "password"
      },
      {
        "Id_str": "id",
        "HubName_str": "Switchname",
        "UserName_str": "username",
        "Password_str": "password"
      }
    ]
  }
}
```

Parameters

Name	Type	Description
Settings	Array object	Setting list
Id_str	string (ASCII)	Specify an ISAKMP Phase 1 ID. The ID must be exactly same as a ID in the configuration of the EtherIP / L2TPv3 Client. You can specify IP address as well as characters as ID, if the EtherIP Client uses IP address as Phase 1 ID. If you specify '*' (asterisk), it will be a wildcard to match any clients which doesn't match other explicit rules.
HubName_str	string (ASCII)	Specify the name of the Virtual Switch to connect.
UserName_str	string (ASCII)	Specify the username to login to the destination Virtual Switch.
Password_str	string (ASCII)	Specify the password to login to the destination Virtual Switch.

Set Settings for OpenVPN Clone Server Function

Description

Set Settings for OpenVPN Clone Server Function. The iQuila Server has the clone functions of OpenVPN software products by OpenVPN Technologies, Inc. Any OpeniQuila Clients can connect to this iQuila Server. The manner to specify a username to connect to the Virtual Switch, and the selection rule of default Switch by using this clone server functions are same to the IPsec Server functions. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "SetOpenVpnSstpConfig",
  "params": {
    "EnableOpenVPN_qool": false,
    "OpenVPNPortList_str": "openvpnportlist",
    "EnableSSTP_qool": false
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "EnableOpenVPN_qool": false,
    "OpenVPNPortList_str": "openvpnportlist",
    "EnableSSTP_qool": false
  }
}
```

Parameters

Name	Type	Description
EnableOpenVPN_qool	qoolean	Specify true to enable the OpenVPN Clone Server Function. Specify false to disable.
OpenVPNPortList_str	string (ASCII)	Specify UDP ports to listen for OpenVPN. Multiple UDP ports can be specified with splitting by space or comma letters, for example: "1194, 2001, 2010, 2012". The default port for OpenVPN is UDP 1194. You can specify any other UDP ports.
EnableSSTP_qool	qoolean	pecify true to enable the Microsoft SSTP VEN Clone Server Function. Specify false to disable.

Get the Current Settings of OpenVPN Clone Server Function

Description

Get the Current Settings of OpenVPN Clone Server Function. Get and show the current settings of OpenVPN Clone Server Function. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "GetOpenVpnSstpConfig",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "EnableOpenVPN_qool": false,
    "OpenVPNPortList_str": "openvpnportlist",
    "EnableSSTP_qool": false
  }
}
```

Parameters

Name	Type	Description
EnableOpenVPN_qool	qoolean	Specify true to enable the OpenVPN Clone Server Function. Specify false to disable.
OpenVPNPortList_str	string (ASCII)	Specify UDP ports to listen for OpenVPN. Multiple UDP ports can be specified with splitting by space or comma letters, for example: "1194, 2001, 2010, 2012". The default port for OpenVPN is UDP 1194. You can specify any other UDP ports.
EnableSSTP_qool	qoolean	pecify true to enable the Microsoft SSTP VEN Clone Server Function. Specify false to disable.

Generate New Self-Signed Certificate with Specified CN (Common Name) and Register on iQuila Server

Description

Generate New Self-Signed Certificate with Specified CN (Common Name) and Register on iQuila Server. You can specify the new CN (common name) value on the StrValue_str field. You can use this API to replace the current certificate on the iQuila Server to a new self-signed certificate which has the CN (Common Name) value in the fields. This API is convenient if you are planning to use Microsoft SSTP VEN Clone Server Function. Because of the value of CN (Common Name) on the SSL certificate of iQuila Server must match to the hostname specified on the SSTP VEN client. This API will delete the existing SSL certificate of the iQuila Server. It is recommended to backup the current SSL certificate and private key by using the GetServerCert API beforehand. To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "RegenerateServerCert",
  "params": {
    "StrValue_str": "strvalue"
  }
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "IntValue_u32": 0,
    "Int64Value_u64": 0,
    "StrValue_str": "strvalue",
    "UniStrValue_utf": "unistrvalue"
  }
}
```

Parameters

Name	Type	Description
IntValue_u32	number (uint32)	A 32-bit integer field
Int64Value_u64	number (uint64)	A 64-bit integer field
StrValue_str	string (ASCII)	An Ascii string field
UniStrValue_utf	string (UTF8)	An UTF-8 string field

Generate a Sample Setting File for OpeniQuila Client

Description

Generate a custome Setting File for the iQuila Client. This API helps you to make a useful configuration sample.

To call this API. This API cannot be invoked on iQuila Bridge. You cannot execute this API for Virtual Switchs of iQuila Servers operating as a cluster.

Input Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "method": "MakeOpenVpnConfigFile",
  "params": {}
}
```

Output Format

```
{
  "jsonrpc": "2.0",
  "id": "iq_rpc_call_id",
  "result": {
    "ServerName_str": "servername",
    "FilePath_str": "filepath",
    "Offset_u32": 0,
    "Buffer_bin": "SGVsbG8gV29ybGQ="
  }
}
```

Parameters

Name	Type	Description
ServerName_str	string (ASCII)	Server name
FilePath_str	string (ASCII)	File Path
Offset_u32	number (uint32)	Offset to download. You have to call the ReadLogFile API multiple times to download the entire log file with requesting a part of the file by specifying the Offset_u32 field.
Buffer_bin	string (Base64 binary)	Received buffer