Custom Queries

This page will serve as a reference for how to use the AAI custom query interface and as a living catalog of available custom queries.

Before You Start!

It's important that you engage the AAI team before using these queries. We're are actively evolving our schema, queries, and other things in AAI. Queries and query parameters may change or be removed in the future. It's important that, at the very least, we know who is using which queries so we can be cautious of changes in the future. And we can help you find the best way to get the data you need.

Getting Started with the Custom Query API

To execute a custom query, a client will perform a PUT on the query API and include a payload indicating the starting node and the query to be run. While the example below is for v9, this can be called in any version v9 or higher. The version dictates which release's REST API version the output will be based on.

API URI

```
PUT /aai/v$/query?format={format}
```

When calling the query API, the client must specify the output format as a query string. The currently available output formats are below, along with examples. Click the Example Output to expand.

count

Provides an count of the objects returned in the query.

Example Output

```
{
  "results": [
    {
      "generic-vnf": 4,
      "p-interface": 1,
      "vserver": 3,
      "service-instance": 1,
      "tenant": 1,
      "pserver": 1
    }
  ]
}
```

id

Provides an array of objects containing resource-type (AAI's node type; i.e., generic-vnf) and a resource-link using the vertex ID from AAI's graph.

Example Output

```
{
  "results": [
    {
      "resource-type": "generic-vnf",
      "resource-link": "/aai/v9/resources/id/2388112"
    },
    {
      "resource-type": "generic-vnf",
      "resource-link": "/aai/v9/resources/id/4694112"
    },...]

}
```

pathed

Provides an array of objects containing resource-type (AAI's node type; i.e., generic-vnf) and a resource-link using the AAI REST API pathed URIs.

Example Output

```
{
  "results": [
    {
      "resource-type": "generic-vnf",
      "resource-link": "/aai/v9/network/generic-vnfs/generic-vnf/lab20105v"
    },
    {
      "resource-type": "generic-vnf",
      "resource-link": "/aai/v9/network/generic-vnfs/generic-vnf/ro-stack01"
    },...]

}
```

resource

Provides the each object in the results array in the same format as AAI's REST API with depth = 1 (first level children and cousin relationships).

Example Output

```
{
  "results": [
    {
      "complex": {
        "physical-location-id": "complex5349-06",
        "complex-name": "complexsa-test",
        "resource-version": "1485403105490",
        "physical-location-type": "POP",
        "street1": "1111 Main Street",
        "city": "Anywhere",
        "state": "CA",
        "postal-code": "90210",
        "country": "USA",
        "region": "USA",
        "latitude": "34.07",
        "longitude": "-118.40",
        "relationship-list": {
          "relationship": [
            {
              "related-to": "pserver",
              "related-link": "https://aai.onap:8443/aai/v9/cloud-
infrastructure/pservers/pserver/pserver5349-06",
              "relationship-data": [
                {
                  "relationship-key": "pserver.hostname",
                  "relationship-value": "pserver5349-06"
                }
              ],
              "related-to-property": [
                {
                  "property-key": "pserver.pserver-name2"
                }
              ]
            }
          ]
        }
      }
    },...]

}
```

resource_and_url

Provides each object in the results array in the same format as AAI's REST API with depth = 1 (first level children and cousin relationships) plus the pathed url for the result object in AAI.

Example Output

```
 {
    "results": [
        {
            "url": "https://aai.onap:8443/aai/v10/cloud-
infrastructure/complexes/complex/MSCWRUGY",
            "complex": {
                "physical-location-id": "MSCWRUGY",
                "complex-name": "arumscw2",
                "resource-version": "1486837035912",
                "physical-location-type": "",
                "street1": "UL. 8-MARTA - BLDG 14",
                "city": "MOSCOW",
                "state": "RU",
                "postal-code": "0",
                "country": "RUS",
                "region": "EMEA"
            }
        },...]
}
```

simple

Provides each result object in a simplified format. The node-type, graph vertex id, pathed url, object properties, and directly related objects in the graph are all returned. Both direct parent/child objects and cousin objects are included in the related-to array.

Example Output

```
{
    "results": [
        {
            "id": "739696712",
            "node-type": "generic-vnf",
            "url": "https://aai.onap:8443/aai/v10/network/generic-
vnfs/generic-vnf/85f60b5e-6eff-49c8-9a79-550ee9eb4806",
            "properties": {
                "vnf-type": "WX",
                "service-id": "d7bb0a21-66f2-4e6d-87d9-9ef3ced63ae4",
                "equipment-role": "UCPE",
                "orchestration-status": "created",
                "management-option": "ONAP",
                "ipv4-oam-address": "12.80.1.20",
                "ipv4-loopback0-address": "32.40.70.237",
                "nm-lan-v6-address": "2001:1890:e00e:fffe::1:2806",
                "management-v6-address": "2001:1890:e00e:fffd::773",
                "vcpu": 4,
                "vmemory": 8,
                "vmemory-units": "GB",
                "vdisk": 150,
                "vdisk-units": "GB",
                "in-maint": false,
                "is-closed-loop-disabled": false,
                "resource-version": "1499958805125",
                "vnf-id": "85f60b5e-6eff-49c8-9a79-550ee9eb4806",
                "vnf-name": "USUCP5PHLPA0703UJWX01"
            },
            "related-to": [
                {
                    "id": "739700808",
                    "node-type": "license",
                    "url": "https://aai.onap:8443/aai/v10/network/generic-
vnfs/generic-vnf/85f60b5e-6eff-49c8-9a79-550ee9eb4806/licenses/license/VCX-
SUB-00755-M/VCX-SUB-00755-M-TAG-20161206-81"
                },
                {
                    "id": "411750576",
                    "node-type": "service-instance",
                    "url":
"https://aai.onap:8443/aai/v10/business/customers/customer/nFOD_GLOB_57829499
9/service-subscriptions/service-subscription/uCPE-VMS/service-
instances/service-instance/USUCP5PHLPA0703UJZZ01"
                },
                {
                    "id": "15372328",
                    "node-type": "vnf-image",
                    "url": "https://aai.onap:8443/aai/v10/service-design-and-
creation/vnf-images/vnf-image/78252548-efb6-4b42-9cf7-2b3900c5e7e2"
                },
                {
                    "id": "700920024",
                    "node-type": "vserver",
                    "url": "https://aai.onap:8443/aai/v10/cloud-
infrastructure/cloud-regions/cloud-
region/owner1/region1/tenants/tenant/USUCP5PHLPA0703UJZZ01%3A%3AuCPE-
VMS/vservers/vserver/a6085509-8da5-4f26-8f0f-a82739743b88"
                }
            ]
        }
    ]
}
```

Walking back through relationships in the simple format:

Let's say you got back a large tree of output in the simple format and need to go through the list of objects to understand their relationships. For example, the output returned vnfs, vservers, pservers and complexes but you want to only look at the results of a particular complex. First, we'll find the JSON object for the complex by looking at the result objects for one with "node-type": "complex" and "physical-location-id": (the CLLI of the location you want to filter on) within the "properties" object. Next you would check the "related-to" object array for objects with "node-type": "pserver", take the "id"s and search for objects with those IDs in the results object array. You can keep crawling through the results in this way until you reach the objects you need. You can use this method for any property you want to filter on.

```
{
    "results": [
                ...,
        {
            "id": "14147624",
            "node-type": "complex",
            "url": "https://aai.onap:8443/aai/v10/cloud-
infrastructure/complexes/complex/STLSMO0901",
            "properties": {
                ...
                "physical-location-id": "STLSMO0901",
                ...
            },
            "related-to": [
                {
                    "id": "2134056",
                    "node-type": "pserver",
                    "url": "https://aai.onap:8443/aai/v10/cloud-
infrastructure/pservers/pserver/USUCP6CMIIL0237UJZZ01"
                },
                {
                    "id": "2158632",
                    "node-type": "pserver",
                    "url": "https://aai.onap:8443/aai/v10/cloud-
infrastructure/pservers/pserver/USBARUBCTIL0118UJZZ01"
                },...
                       ]
               },...
        ]
}
```

graphson

Provides the results using the graphson standard.

Example Output

```
 {
            "id": 2213998664,
            "label": "vertex",
            "outE": {
                "hasInstance": [
                    {
                        "id": "38lint5-10m5oaw-8ph-z88l3c",
                        "inV": 2130153672,
                        "properties": {
                            "SVC-INFRA": "OUT",
                            "prevent-delete": "NONE",
                            "delete-other-v": "NONE",
                            "contains-other-v": "NONE"
                        }
                    }
                ],
                "runsOnVserver": [
                    {
                        "id": "38lio7d-10m5oaw-15w5-11yxuso",
                        "inV": 2295935016,
                        "properties": {
                            "prevent-delete": "NONE",
                            "SVC-INFRA": "OUT",
                            "delete-other-v": "NONE",
                            "contains-other-v": "NONE"
                        }
                    }
                ]
            },
            "properties": {
                "aai-last-mod-ts": [
                    {
                        "id": "38lijgp-10m5oaw-2rk5",
                        "value": 1499913739139
                    }
                ],
                "service-id": [
                    {
                        "id": "38lilft-10m5oaw-1czp",
                        "value": "d7bb0a21-66f2-4e6d-87d9-9ef3ced63ae4"
                    }
                ],
                "vnf-id": [
                    {
                        "id": "38lik95-10m5oaw-6uit",
                        "value": "2d42aa64-c3e6-455b-af80-d9cc94247dc6"
                    }
                ],
                "aai-uri": [
                    {
                        "id": "38liia1-10m5oaw-kw79",
                        "value": "/network/generic-vnfs/generic-vnf/2d42aa64-
c3e6-455b-af80-d9cc94247dc6"
                    }
                ],
                "prov-status": [
                    {
                        "id": "38lilu1-10m5oaw-afb9",
                        "value": ""
                    }
                ],
                "equipment-role": [
                    {
                        "id": "38lim89-10m5oaw-agw5",
                        "value": ""
                    }
                ],
                "aai-created-ts": [
                    {
                        "id": "38ligp5-10m5oaw-c3d1",
                        "value": 1499913739138
                    }
                ],
                "source-of-truth": [
                    {
                        "id": "38ligax-10m5oaw-4kcl",
                        "value": "SDNC"
                    }
                ],
                "vnf-type": [
                    {
                        "id": "38lil1l-10m5oaw-9og5",
                        "value": "SW"
                    }
                ],
                "aai-node-type": [
                    {
                        "id": "38liio9-10m5oaw-1n9h",
                        "value": "generic-vnf"
                    }
                ],
                "orchestration-status": [
                    {
                        "id": "38limmh-10m5oaw-bjlx",
                        "value": "Created"
                    }
                ],
                "in-maint": [
                    {
                        "id": "38lin0p-10m5oaw-3sp1",
                        "value": false
                    }
                ],
                "resource-version": [
                    {
                        "id": "38lij2h-10m5oaw-6io5",
                        "value": "1499913739139"
                    }
                ],
                "last-mod-source-of-truth": [
                    {
                        "id": "38lijux-10m5oaw-79j9",
                        "value": "SDNC"
                    }
                ],
                "is-closed-loop-disabled": [
                    {
                        "id": "38linex-10m5oaw-6net",
                        "value": false
                    }
                ],
                "vnf-name": [
                    {
                        "id": "38liknd-10m5oaw-85xh",
                        "value": "USMSOADOMIL0186UJSW01"
                    }
                ]
            }
        },...]
}
```

Optional Query Parameters

You can pass a depth query parameter on the resource or resource_and_url formats to indicate what level of child objects you want returned. By default the output will be depth = 1 (first level children).

```
PUT /aai/v$/query?format={resource OR resource_and_url}&depth=0
```

You can pass a nodes only query parameter to have the output only contain the object properties with no relationships. By default the output will be of depth = 1 (first level children and cousin relationships).

```
PUT /aai/v$/query?format={format}&nodesOnly=true
```

You can pass a subgraph query parameter that determines the behavior of the output. By default, a query returns all of the objects from the query and all of their relationships. Using subgraph=prune returns all of the objects from the query and only the edges between those objects. Using subgraph=star returns all of the objects from the query plus all of the objects they relate to.

```
PUT /aai/v$/query?format={format}&subgraph={subgraph}
```

Payload

Typically the query payload will include both a "start" and a "query" portion. The "start" can indicate one or more starting nodes in the graph. If multiple nodes are specified, the result will contain the query results for all of the start nodes. The "query" indicates the name of the query to be run and also takes query parameters depending on the query. Please reference the page for each specific saved query for how it should be used, but keep in mind that any URI can be used in the start parameter as long as it provides the same object types. **Note: The start URI must adhere to standard percent-encoding rules to properly account for special characters.**

```
{
        "start" : ["{namespace}/{resource}"],
        "query" : "query/{query-name}"
}
```

There also the option to pass a "start" to the query API with no specified query. This will return the input node(s) in the format requested.

```
{
        "start" : ["{namespace}/{resource}"]
}
```

Available Queries

access-service-fromServiceInstance

The **"access-service-fromServiceInstance"** query allows a client to provide AAI a global-customer-id a service-type for a service-subscription, and a service-instance-id to retrieve service-subscription, customer, forwarding-path, configuration, evc, forwarder, forwarder-evc, p-interface, pnf, lag-interface, and logical-link of link-type LAG.

Input:

global-customer-id

service-subscription

service-instance-id

Output:

service-subscription

customer

forwarding-path

configuration

evc

forwarder

forwarder-evc

p-interface

pnf

lag-interface

logical-link (of link-type LAG)

availabilityZoneAndComplex-fromCloudRegion

The **"availabilityZoneAndComplex-fromCloudRegion"** query allows a client to provide AAI a cloud-owner and cloud-region-id to retrieve the availability-zones and complex.

Input:

cloud-owner

cloud-region-id

Output:

availability-zones

complex

cloud-region-and-source-FromConfiguration

The **"cloud-region-and-source-FromConfiguration"** query allows a client to provide AAI with a configuration-id and retrieve the source cloud-region and source vnf..

Query needs to be submitted using format=simple&nodesOnly=true

Input:

configuration-id

Output:

cloud-region

generic-vnf (source vnf)

cloudRegion-fromCountry

The **"cloudRegion-fromCountry"** query allows a client to provide AAI with a country and retrieve all appropriate cloud-regions.

Input:

country

Output:

cloud-region

cloudRegion-fromCountryCloudRegionVersion

The **"cloudRegion-fromCountryCloudRegionVersion"** query allows a client to provide AAI with a country code and cloud-region-version and returns the appropriate cloud-regions.

Input:

country

cloud-region-version

Output:

cloud-region

cloudRegion-fromNfType

The **"cloudRegion-fromNfType"** query allows a client to provide AAI with an nf-type and returns the cloud-regions running those vnfs.

Input:

nf-type

Output:

cloud-region

cloudRegion-fromNfTypeVendorVersion

The **"cloudRegion-fromNfTypeVendorVersion"** query allows a client to provide AAI with an nf-type, application-vendor, and optional application-version and retrieve the cloud-regions.

Input:

nf-type

application-vendor

optional application-version

Output:

cloud-region

cloud-region-fromVnf

The "cloud-region-fromVnf" query allows a client to provide AAI with a vnf-id and retrieves the tenant, cloud-region, and cloud-owner.

Input:

vnf-id

Output:

vserver

vnfc

tenant

cloud-region

cloud-region-sites

The **"cloud-region-sites"** query allows a client to provide AAI with a cloud-owner and retrieves the cloud-regions having that owner and all of the complexes containing those cloud-regions.

Input:

cloud-regions

The set of cloud-regions is determined by the cloud-owner.

Output:

the cloud-regions

the complexes

colocated-devices

The **"colocated-devices"** query allows a client to provide AAI a physical server and retrieves all other physical devices in the same location along with details on their physical interfaces and links.

Input:

pserver

Output:

pservers

pnfs

p-interfaces

physical-links

complex-fromVnf

The **"complex-fromVnf"** query allows a client to provide AAI a vnf name or ID to retrieve the generic-vnf, pserver, complex, licenses, and entitlements.

Input:

vnf name or vnf ID

Output:

generic-vnf

pserver

complex

licenses

entitlements

count-vnf-byVnfType

The **"count-vnf-byVnfType"** query allows a client to get a list of the number of generic-vnfs for each vnf type.

Format must be set to "console", otherwise no data will be displayed.

Input:

None

Output:

vnf-types = count of generic-vnfs

destination-FromConfiguration

The **"destination-FromConfiguration"** query allows a client to provide AAI with a configuration-id and retrieve the destination vnf or pnf..

Query needs to be submitted using format=simple&nodesOnly=true

Input:

configuration-id

Output:

pnf

generic-vnf (destination vnf)

fabric-information-fromVnf

The **fabric-information-fromVnf** query will retrieve fabric information for a given VNF.

Input:

generic-vnf

Output:

vserver, p-interface, pserver, vlan-tag

fn-topology

The **"fn-topology"** query allows a client to provide AAI service-instance-id or line-of-business-name then return vnf, vnfc, vserver, pserver, pnf.

Input:

service-instance-id OR line-of-business-name

Output:

vnf, vnfc, vserver, pserver, pnf

generic-vnfFromModelbyRegion

The **"generic-vnfFromModelbyRegion"** query allows a client to provide AAI with a global-customer-id, service-type, model parameters, and cloud-region-id and retrieves the related generic-vnfs.

Input:

global-customer-id

service-type

model-version-id

model-invariant-id

cloud-region-id

Output:

generic-vnf

getNetworks

The **getNetworks** query will retrieve l3-networks for a given network-role, cloud-region and owning-entity

Input:

network-role of l3-network

cloud-region-id where the l3-network resides

owning-entity-id of the service instance running on the l3-network

Output:

l3-network

getNetworksByServiceInstance

The **"getNetworksByServiceInstance"** query allows a client to return provider networks with associated vlan-tags and tenant networks with associated vlan-tags by service-instance-id.

Input:

service-instance-id

Output:

l3-network, vlan-tag

getServiceTopology

The **"getServiceTopology"** query allows a client to provide AAI with a service-instance and retrieve the generic-vnfs, vlans, vservers, l-interfaces, pservers, complexes, and allotted-resources. It then finds any service-instances attached to the allotted-resources and retrieves the above values for those service-instances except for pservers, complexes, and allotted-resources. The client must provide a path to the service-instance from customer and service-subscription.

This query is meant to replace the named query "dhv-service-topology-2". The format closest to this original query can be found with format=simple&depth=0&nodesOnly=true

Input:

service-instance (full path starting from customer and service-subscription)

Output:

service-instance

generic-vnf

l-interface

vlan

l3-interface-ipv4/6-address list

vserver

pserver

complex

allotted-resource

service-instance (from allotted-resources)

images-fromCloudRegionNfType

The **"images-fromCloudRegionNfType"** query allows a client to provide AAI with a cloud-region-id and nf-type and retrieve all related images.

Input:

cloud-region-id

nf-type

Output:

image

instance-group-byCloudRegion

The **"instance-group-byCloudRegion"** query allows the user to get all instance-groups by cloud-region-id and filter by instance-group type/role/function.

Input:

cloud-region

instance-group type, role and function

Output:

instance-group

ips-networks-fromVnf

The **"ips-networks-fromVnf"** query allows a client to provide AAI one or more VNFs and retrieve various data all associated VIP and fixed IPs and their related networks.

Input:

generic-vnf

Output:

the generic-vnf

the related vnfs and their related vip-ipv4/6-address-list and l3-interface-ipv4/6-address-list

the related vservers and their related l3-interface-ipv4/6-address-list

the l3-networks related to all of the address lists

the related complex

l3-networks-by-cloud-region-network-role

The **"l3-networks-by-cloud-region-network-role"** query retrieves l3-networks for a given cloud-region-id, tenant.tenant-id (Optional) and network-role.

Input:

cloud-region

network-role

Output:

l3-network

linked-devices

The **"linked-devices"** query allows a client to provide AAI a generic-vnf, vserver, or newvce and retrieve all connected generic-vnfs, vservers, and newvces.

Input:

generic-vnf, vserver, or newvce

Output:

all connected generic-vnfs, vservers, and newvces

locationNetTypeNetRole-fromCloudRegion

The **"locationNetTypeNetRole-fromCloudRegion"** query allows a client to provide AAI with a cloud-region-id and returns the cloud-region, complex, and l3-networks.

Input:

cloud-region-id

Output:

cloud-region

complex

l3-network

network-collection-ByServiceInstance

The **"network-collection-ByServiceInstance"** query

Query to return the service-instance and associated collection, instance-group and associated l3-networks for a given service-instance-id.

Input:

service-instance

Output:

Service-instance and associated collection, instance-group and l3-networks for a given service-instance-id.

network-name-fromNetwork-role

The **" network-name-fromNetwork-role"** query allows a client to provide AAI with a cloud-owner and cloud-region-id and retrieves the related l3-networks and network-policies.

Input:

network-role

cloud-region

cloud-owner

Output:

l3-network

network-policy

nfType-fromCloudRegion

The **"nfType-fromCloudRegion"** query allows a client to provide AAI with a cloud-region-id and returns a list of all generic-vnfs with an nf-type.

Input:

cloud-region-id

Output:

nf-type

owning-entity-fromService-instance

The "owning-entity-fromService-instance" query allows a client to provide AAI with a service-instance-id and retrieves the owning-entity.

Input:

service-instance-id

Output:

owning-entity

pending-topology-detail

The **"pending-topology-detail"** query allows a client to provide AAI a generic as input and returns the generic-vnf, platform(s), line(s)-of-business, owning-entity, project, vnfc(s), vnfc ip address(es), vip ip addresses subnet(s), and l3-networks.

Input:

generic-vnf

Output:

generic-vnf

platform

line-of-business

owning-entity

project

vnfc

vnfc, vnfc.l3-interface-ipv4/6-address-list

vip-ipv4/6-address-list

subnet, l3-network

pnf-fromModel-byRegion

The **"pnf-fromModel-byRegion"** query allows a client to provide AAI with a cloud-region, equip-vendor, equip-model, model-invariant-id of service-instance, model-version-id of service-instance and retrieves the pnf.

Input:

cloud-region

equip-vendor

equip-model

model-invariant-id of service-instance

model-version-id of service-instance

Output:

pnf

pnf-topology

The **"pnf-topology"** query allows a client to provide AAI a D1 Device, using the hostname, and retrieve data related to that device and its connected uCPE and/or other D1 device. This includes data about the D1 device itself (the pnf, and location) as well as about a connected uCPE (the pserver, interfaces and physical links used for the connection) and/or other D1 device (the pnf, interfaces and physical links used for the connection).

Input:

pnf

Using pnf.pnf-name to identify the pnf is preferred.

Output:

the pnf

related complex

related p-interface(s), physical-links(s), pserver(s) and pserver p-interface(s), and/or pnf(s) and pnf p-interface(s)

pserver-fromConfiguration

The **"pserver-fromConfiguration"** query allows a client to provide AAI with a configuration-id and retrieves the configuration and related l-interfaces, pservers, and generic-vnfs.

Input:

configuration-id

Output:

l-interface

pserver

generic-vnf

pnf

pserver-fromConfigurationFilterInterfaceId

The **"pserver-fromConfigurationFilterInterfaceId"** query allows a client to provide AAI with a configuration-id and interface-id and retrieves the configuration, l-interface, and related pservers, and generic-vnfs.

Input:

configuration-id

interface-id

Output:

configuration

l-interface

pserver

generic-vnf

pservers-fromVnf

The **"pservers-fromVnf"** query allows a client to provide AAI a vNF and retrieve all of the pservers hosting that vNF.

Input:

generic-vnf

Output:

pserver

pservers-withNoComplex

The **"pservers-withNoComplex"** query allows a client to get a list of pservers that have no edge to any complex.

Format must be set to "console", otherwise no data will be displayed.

Input:

None

Output:

pserver

related-to

The **"related-to"** query allows a client to provide AAI any starting node and request all related nodes of a requested node-type.

Input:

A start node of any single node-type

A related to parameter value of any single node-type, related to the starting node type. (Providing a node-type not related to the starting type will result in an error)

Output:

The related to node-type

service-fromPserverandSubsName

The **"service-fromPServerandSubsName"** query allows a client to provide AAI a hostname and subscriber name, then return service instance and service subscription information.

Input:

hostname and subscriber-name

Output:

service-instance and service-subscription

serviceModels-byDistributionStatus

The **"serviceModels-byDistributionStatus"** query allows a client to provide AAI with a distribution-status and optional model-invariant-id and retrieve the model and model-ver.

Input:

distribution-status

Optional model-invaraint-id

Output:

model

model-ver

service-sites

The **"service-sites"** query allows a client to provide AAI a service type and a customer id to retrieve the service-instances, cloud regions, generic-vnfs, and complexes.

Input:

Service type

Customer ID

Output:

all attributes of the following vertices:

service-instance, cloud region, generic-vnf, complex

service-topology

The **"service-topology"** query allows a client to provide AAI with a service-instance and retrieve the generic-vnfs, connected tenants, vservers, vnfcs, pservers, and their interfaces.

This query is intended to use with format=resource_and_url and depth=0, using the node urls to identify parent-child relationships.

Input:

service-instance

Output:

service-instance

generic-vnf

tenant

vserver

vnfc

l-interface

l3-interface-ipv4/6-address list

subnet

l3-network

pserver

p-interface

physical-link

site-l3network-cloudRegion

The **"site-l3network-cloudRegion"** query allows a client to provide AAI with a physical-location-id and returns the network-role, country, cloud-region-id and cloud-region-version in that location.

Input:

physical-location-id

Output:

network-role

country

cloud-region-id

cloud-region-version

sites-byCloudRegionId

The **"sites-byCloudRegionId "** query allows a client to provide AAI with a cloud-region-id and an optional cloud-region-version and returns the appropriate complexes.

Input:

cloud-region-id

optional cloud-region-version

Output:

complex object(s)

sites-byCountryFilterCloudRegionId

The **"sites-byCountryFilterCloudRegionId"** query allows a client to provide AAI with a 3-digit country code and cloud-region-id to retrieve the appropriate complexes.

Input:

country code

cloud-region-Id

Output:

complex object(s)

sites-byCountryFilterCloudRegionVer

The **"sites-byCountryFilterCloudRegionVer"** query allows a client to provide AAI with a 3-digit country code and cloud-region-version number to retrieve the appropriate complexes.

Input:

country code

cloud-region-version

Output:

complex object(s)

so-request-vfModule

The **"so-request-vfModule"** query allows a client to provide AAI a vf-module then return all the reference objects needed to send MSO an orchestration request.

Input:

vf-module

Output:

vf-module

generic-vnf

service-instance

volume-group

cloud-region

spaas-topology-fromServiceInstance

The **"spaas-topology-fromServiceInstance"** query allows a client to provide AAI global-custom-id and service-type, then return vertical topology for overlay and underlay information.

Input:

global-custom-id service-type

Output:

vserver, flavor, image, l-interface, logical-link, l3-interface-ipv4(6)-address-list, subnet, l3-network, pserver, complex, physical-link

topology-detail

The **"topology-detail"** query allows a client to provide AAI a generic-vnf as input and returns the generic-vnf, platform(s), line(s)-of-business, owning-entity, project, vnfc(s), vserver(s), vserver l-interface(s), ip address(es), subnet(s), l3-networks, cloud-region and complex.

Input:

generic-vnf

Output:

generic-vnf

platform

line-of-business

owning-entity

project

vnfc

vserver, vserver.l-interface, vserver.l-interface.l3-interface-ipv4/6-address-list

subnet, l3-network

cloud-region

complex

topology-detail-fromVnf

The **"topology-detail-fromVnf"** query allows a client to provide AAI with a service-id of a VNF and retrieve various data related to that VNF. This includes data about the VNF itself (the generic-vnf), the related vnfc, the related vserver (along with the tenant, cloud-region, image and flavor) and the related pserver (along with the complex) as done in the topology-summary query. In addition, this query returns availability-zone, service-instance, l-interface, l3-interface-ipv4-address-list, l3-interface-ipv6-address-list, and volume-group.

Input:

generic-vnf service-id.

Output:

generic-vnf

vnfc

vserver, tenant, cloud-region, image, flavor

pserver, complex

availability-zone

service-instance

l-interface

l3-interface-ipv4-address-list

l3-interface-ipv6-address-list

volume-group

topology-detail-fromVserver

The **"topology-detail-fromVserver"** query allows a client to provide AAI a vserver as input and returns the generic-vnf, platform(s), line(s)-of-business, owning-entity, project, vnfc(s), vserver(s), vserver l-interface(s), ip address(es), subnet(s), l3-networks, cloud-region and complex. Updated in 1806 to return the following additional objects: pserver, availability-zone, tenant, image, flavor, virtual-data-center, vf-module, and volume-group.

Input:

vserver

Output:

generic-vnf

platform

line-of-business

owning-entity

project

vnfc

vserver, vserver.l-interface, vserver.l-interface.l3-interface-ipv4/6-address-list

subnet, l3-network

cloud-region

complex

pserver

availability-zone

tenant

image

flavor

virtual-data-center

vf-module

volume-group

topology-fromCloudRegionIdandServiceId

The **"topology-fromCloudRegionIdandServiceId"** query allows a client to provide AAI cloud-owner, cloud-region-id and service-id, then return topology related to the service id.

Input:

cloud-owner cloud-region-id service-id

Output:

service-instance, vserver, flavor, image, volume, vnfc, snapshot, vf-module, l-interface, logical-link, l3-interface-ipv4(6)-address-list, subnet, l3-network

topology-summary

The **"topology-summary"** query allows a client to provide AAI one or more VNFs and retrieve various data related to that VNF. This includes data about the VNF itself (the generic-vnf), the related vnfc, the related vserver (along with the tenant, cloud-region, image and flavor) and the related pserver (along with the complex).

Input:

generic-vnf

The original intent was to pass all generic-vnfs with a specified service-id.

(adding owning-entity, project, platform and line-of-business in 1810 as first step to depreciating service-id)

Output:

the generic-vnf

related platform, line-of-business (1810)

related owning-entity, project (from related service-instance) (1810)

related vnfc

related vserver, tenant, cloud-region, image, flavor

related pserver, complex

topology-summary-fromCloudRegion

The **"topology-summary-fromCloudRegion"** query allows a client to provide AAI a cloud region and retrieve a summary of the topology within that cloud region including the tenants, VMs, VNFs and physical servers.

Input:

cloud-region

Note: In the interest of performance, it is highly recommended to run this query for only one cloud-region at a time.

Output:

cloud-region

tenants

vservers

generic-vnfs

pservers

topology-summary-fromTenant

The **"topology-summary-fromTenant"** query allows a client to provide AAI a tenant and retrieve a summary of the topology within that tenant including VMs, VNFs and physical servers and the containing cloud region.

Input:

tenant

Output:

tenant

cloud-region

vservers

generic-vnfs

pservers

ucpe-instance

The **"ucpe-instance"** query allows a client to provide AAI a physical server or physical network device, using the hostname, and retrieve the device and the complex it is located in. This includes the pserver or pnf itself and the complex.

Input:

pserver or pnf

Using pserver hostname or pnf pnf-name to identify the starting node is preferred.

Output:

the pserver or pnf

related complex

ucpe-topology

The **"ucpe-topology"** query allows a client to provide AAI a uCPE physical server, using the hostname, and retrieve various data related to that uCPE. This includes data about the uCPE itself (the pserver, location, interfaces, hosted vnfs, service instances, service subscriptions and customer) as well as about a connected physical D1 device (the pnf, interfaces and physical links).

Input:

pserver

Using pserver.hostname to identify the pserver is preferred.

Output:

the pserver

related complex

related p-interface(s), physical-links(s), pnf(s) and pnf p-interface(s)

related generic-vnf(s), vnf-image(s), service-instance(s), service-subscription(s) and customer

vfModule-fromServiceInstance

The **"vfModule-fromServiceInstance"** query allows a client to provide AAI a service-instance-id to retrieve vf-module only.

Input:

service-instance-id

Output:

vf-module

vnf-instances-fromServiceInstancebyModelVersion

The **"vnf-instances-fromServiceInstancebyModelVersion"** query allows a client to provide AAI a list of service-instances for a customer and service-type and return the generic-vnfs using a particular model-version-id.

Input:

service-instances

filters for generic-vnf vnf-type and model-version-id

Output:

generic-vnf(s)

vnfs-fromPserver

The **"vnfs-fromPserver"** query allows a client to provide AAI with a pserver hostname and retrieve the generic-vnfs related to it. This query also supports pre-filtering the vnf results.

Input:

pserver hostname

optional generic-vnf properties

Output:

generic-vnfs

vnfs-fromServiceInstance

The **"vnfs-fromServiceInstance"** query allows a client to provide AAI a service-instance and retrieve the related VNFs.

Input:

service-instance

Output:

generic-vnf

vnfs-vlans-fromServiceInstance

The **"vnfs-vlans-fromServiceInstance"** query allows a client to provide AAI a service-instance id, then return associated vnfs and corresponding VLAN ID assignment information for each VNF that is associated to the VNF.

Input:

service-instance

Output:

generic-vnf

vlan

vnf-topology-fromServiceInstance

The **"vnf-topology-fromServiceInstance"** query allows a client to provide AAI a service-instance and retrieve much of the topology related to it. The related VNF, vservers and pserver, along with any IP addresses and l3-networks on the VNF or vserver, the service-instance and allotted-resource, the tenant and cloud region.

In 1810 this custom query is modified to add provider, tenant networks

Input:

service-instance

Output:

generic-vnf all related vf-modules, service-instance, configuration, customer, allotted-resource(s), generic-vnf.l3-interface-ipv4-address-list, generic-vnf.l3-interface-ipv6-address-list, l3-network (vlan-tag), all related vservers, tenant, cloud-region, vserver.l3-interface-ipv4-address-list, vserver.l3-interface-ipv6-address-list, l3-network, and pserver

vnf-topology-fromVfModule

The **"vnf-topology-fromVfModule"** query allows a client to provide AAI a vf-module and retrieve much of the topology related to it. The related VNF, vservers and pserver, along with any IP addresses and l3-networks on the VNF or vserver, the service-instance and allotted-resource, the tenant and cloud region.

Input:

vf-module

Output:

generic-vnf all related vf-modules, service-instance, customer, allotted-resource(s), generic-vnf.l3-interface-ipv4-address-list, generic-vnf.l3-interface-ipv6-address-list, l3-network, all related vservers, tenant, cloud-region, vserver.l3-interface-ipv4-address-list, vserver.l3-interface-ipv6-address-list, l3-network, and pserver

vnf-topology-fromVnf

The **"vnf-topology-fromVnf"** query allows a client to provide AAI a generic-vnf and retrieve much of the topology related to it. The related VNF, vservers and pserver, along with any IP addresses and l3-networks on the VNF or vserver, the service-instance and allotted-resource, the tenant and cloud region.

Input:

generic-vnf

Output:

generic-vnf all related vf-modules, service-instance, service-subscription, customer, allotted-resource(s), generic-vnf.l-interface, l3-network, vservers, tenant, cloud-region, vserver.l-interface, l-interface.l3-interface-ipv4-address-list, l-interface.l3-interface-ipv6-address-list, l3-network, and pserver

vserver-fromInstanceGroup

The "vserver-fromInstanceGroup" query allows a client to provide AAI a instance-group.id to retrieve VNF and vserver information.

Input:

id

Output:

generic-vnf

vserver

vserver-fromVnf

The "vserver-fromVnf" query allows a client to provide AAI with a vnf-id and nfc-function of the vnfc and retrieves the vserver, vnfc, and l-interface.

Input:

vnf-id

nfc-function

Output:

vserver

vnfc

l-interface

vserverlogicallink-frompServer

The **"vserverlogicallink-frompServer"** query allows a client to provide AAI a hostname, then return logical link of vserver from the compute node.

Input:

hostname

Output:

logical-link object

vservers-fromPserver-tree

The **"vservers-fromPserver-tree"** query allows a client to provide AAI one or more pservers and retrieve each pserver with the vservers it hosts nested under it in the output.

Input:

pserver

Output:

pserver

vserver