

This page illustrates NBI API call flow. This is provided as an example of how these APIs could be used.

## Table of Contents

1/ Get nbi healthcheck .....	1
2/ Find serviceSpec(s) in the catalog.....	1
3/ Retrieve one service in the catalog.....	4
4/ Service Order Creation.....	6
5/ Find Service Order(s).....	10
6/ Retrieve one Service Order .....	10
7/ Find instantiated Service(s).....	11
8/ Retrieve one Service instantiated.....	12

## 1/ Get nbi healthcheck

Send a GET message to

```
https://{api_url}/nbi/api/v1/status
```

If everything is working, you can expect to get a '200 OK' status with following body:

```
{  
  "name": "nbi",  
  "status": "ok",  
  "version": "v1"  
}
```

Else a '503 Service Unavailable' is send back

## 2/ Find serviceSpec(s) in the catalog

Send a GET message to

```
https://{api_url}/nbi/api/v1/serviceSpecification/
```

*Note: you can filter request on category and/or distributionStatus*

*On category: https://{api\_url}/nbi/api/v1/serviceSpecification/?category=NetworkService*

On distribution Status: [https://{api\\_url}/nbi/api/v1/serviceSpecification/?distributionStatus=DISTRIBUTED](https://{api_url}/nbi/api/v1/serviceSpecification/?distributionStatus=DISTRIBUTED)

On both: [https://{api\\_url}/nbi/api/v1/serviceSpecification/?category=NetworkService&distributionStatus=DISTRIBUTED](https://{api_url}/nbi/api/v1/serviceSpecification/?category=NetworkService&distributionStatus=DISTRIBUTED)

Wait for a response from the server with the following characteristics

- response Code 200-OK
- The body of the response includes a list of service specification. This list could be empty if no service specification

```
[
  {
    "id": "446afaf6-79b5-420e-aff8-7551b00bb510",
    "name": "FreeRadius-service",
    "invariantUUID": "7e4781e8-6c6e-41c5-b889-6a321d5f2490",
    "category": "Network L4+",
    "distributionStatus": "DISTRIBUTED",
    "version": "1.0",
    "lifecycleStatus": "CERTIFIED",
    "relatedParty": {
      "id": "jm0007",
      "role": "lastUpdater"
    }
  },
  {
    "id": "e245a0a2-34fc-46c7-912c-bd39305275c1",
    "name": "vMRFaas",
    "invariantUUID": "29db0b2c-a877-45e8-8175-4ac526d87eee",
    "category": "Network L4+",
    "distributionStatus": "DISTRIBUTED",
    "version": "1.0",
    "lifecycleStatus": "CERTIFIED",
    "relatedParty": {
      "id": "jm0007",
      "role": "lastUpdater"
    }
  },
  {
    "id": "b9e06990-99e2-480f-96ac-c6e50c83f2bb",
    "name": "vMRFaas2",
    "invariantUUID": "008943ad-73be-4d57-9105-962c4cb16a71",
    "category": "Network L4+",
    "distributionStatus": "DISTRIBUTED",
    "version": "1.0",
    "lifecycleStatus": "CERTIFIED",
    "relatedParty": {
      "id": "jm0007",
      "role": "lastUpdater"
    }
  },
  {
    "id": "98d95267-5e0f-4531-abf8-f14b90031dc5",
    "name": "NewFreeRadius-service",
    "invariantUUID": "709d157b-52fb-4250-976e-7133dff5c347",
    "category": "Network L4+",
    "distributionStatus": "DISTRIBUTED",
    "version": "1.0",
```

```
    "lifecycleStatus": "CERTIFIED",
    "relatedParty": {
      "id": "jm0007",
      "role": "lastUpdater"
    }
  },
  {
    "id": "d8d2429b-0863-462c-adc7-cd38d77331ed",
    "name": "vMRFaaS4",
    "invariantUUID": "e1685613-136b-4990-a590-d0651a640a68",
    "category": "Network Service",
    "distributionStatus": "DISTRIBUTION_NOT_APPROVED",
    "version": "1.1",
    "lifecycleStatus": "NOT_CERTIFIED_CHECKOUT",
    "relatedParty": {
      "id": "cs0008",
      "role": "lastUpdater"
    }
  },
  {
    "id": "9cfc73ad-795d-42ad-8a4f-54e6b3f33ef9",
    "name": "vMRFaaS3",
    "invariantUUID": "79f0574e-caf8-4bac-8189-909b2127e9e9",
    "category": "Network Service",
    "distributionStatus": "DISTRIBUTED",
    "version": "1.0",
    "lifecycleStatus": "CERTIFIED",
    "relatedParty": {
      "id": "jm0007",
      "role": "lastUpdater"
    }
  },
  {
    "id": "af1ac9db-9d8b-4ca0-964c-58755826b59a",
    "name": "vIMSService3",
    "invariantUUID": "d2b9c343-c41a-4f85-ae36-c53cd4b7abcc",
    "category": "VoIP Call Control",
    "distributionStatus": "DISTRIBUTED",
    "version": "2.0",
    "lifecycleStatus": "CERTIFIED",
    "relatedParty": {
      "id": "jm0007",
      "role": "lastUpdater"
    }
  },
  {
    "id": "1e3feeb0-8e36-46c6-862c-236d9c626439",
    "name": "vFW-service-2VF-based",
    "invariantUUID": "b58a118e-eeb9-4f6e-bdca-e292f84d17df",
    "category": "Network Service",
    "distributionStatus": "DISTRIBUTED",
    "version": "2.0",
    "lifecycleStatus": "CERTIFIED",
    "relatedParty": {
      "id": "jm0007",
      "role": "lastUpdater"
    }
  }
}
```

]

### 3/ Retrieve one service in the catalog

It is also possible to use the GET operation to retrieve all information for one service specification.

In order to do that, send a GET message to

```
https://{api_url}/nbi/api/v1/serviceSpecification/1e3feeb0-8e36-46c6-862c-236d9c626439
```

**1e3feeb0-8e36-46c6-862c-236d9c626439** is the id of a serviceSpecification (it corresponds to the SDC service uuid).

Wait for a response from the server with the following characteristics

- response Code 200-OK (sunny day)
- The body of the response includes one serviceSpecification resource referring to its id.

```
{
  "id": "1e3feeb0-8e36-46c6-862c-236d9c626439",
  "name": "vFW-service-2VF-based",
  "invariantUUID": "b58a118e-eeb9-4f6e-bdca-e292f84d17df",
  "toscaModelURL": "/sdc/v1/catalog/services/1e3feeb0-8e36-46c6-862c-236d9c626439/toscaModel",
  "distributionStatus": "DISTRIBUTED",
  "version": "2.0",
  "lifecycleStatus": "CERTIFIED",
  "attachment": [
    {
      "id": "2e6cd967-93c2-4f53-82bd-c56ab98e8df7",
      "name": "AAI-vFW-service-2VF-based-service-2.0.xml",
      "description": "AAI Service Model",
      "artifactLabel": "aaiservice1603481860",
      "artifactGroupType": "DEPLOYMENT",
      "artifactChecksum": "ZTY5ZTJmYTY4YzE2NGUxMTQxNWNkN2QzMmI4MmWIZNDU=",
      "artifactVersion": "1",
      "url": "/sdc/v1/catalog/services/1e3feeb0-8e36-46c6-862c-236d9c626439/artifacts/2e6cd967-93c2-4f53-82bd-c56ab98e8df7",
      "mimeType": "MODEL_INVENTORY_PROFILE",
      "@type": "ONAPartifact"
    },
    {
      "id": "ad600a09-edde-4356-bb0a-9e638a671d06",
      "name": "AAI-Vpkg..base_vpkg..module-0-resource-2.xml",
      "description": "AAI Resource Model",
      "artifactLabel": "aairesource529289386",
      "artifactGroupType": "DEPLOYMENT",
      "artifactChecksum": "MzQyMjc0OGVmYzM1OWQ1NmFhZjBhOWUxM2JjMmYxZTQ=",
      "artifactVersion": "1",
      "url": "/sdc/v1/catalog/services/1e3feeb0-8e36-46c6-862c-236d9c626439/artifacts/ad600a09-edde-4356-bb0a-9e638a671d06",
      "mimeType": "MODEL_INVENTORY_PROFILE",
      "@type": "ONAPartifact"
    },
    {
      "id": "ac9a3bbf-a052-4176-abff-fe2d2741194a",
      "name": "AAI-VfwVsink..base_vfw..module-0-resource-2.xml",
      "description": "AAI Resource Model",
      "artifactLabel": "aairesource1106409880",
      "artifactGroupType": "DEPLOYMENT",
    }
  ]
}
```

```

        "artifactChecksum": "OWIzNjVhNmI2ZWYyZTh1Mjk1NjA2MDFhZTU3MGQ0ZDU=",
        "artifactVersion": "1",
        "url": "/sdc/v1/catalog/services/1e3feeb0-8e36-46c6-862c-
236d9c626439/artifacts/ac9a3bbf-a052-4176-abff-fe2d2741194a",
        "mimeType": "MODEL_INVENTORY_PROFILE",
        "@type": "ONAPartifact"
    }
],
"relatedParty": {
    "id": "jm0007",
    "name": "Joni Mitchell",
    "role": "lastUpdater"
},
"resourceSpecification": [
    {
        "id": "89a6b4c5-3973-4c19-b651-fae3713ca8d5",
        "version": "2.0",
        "name": "vFW-vSINK",
        "instanceName": "vFW-vSINK 0",
        "resourceInvariantUUID": "18b90934-aa82-456f-938e-e74a07a426f3",
        "@type": "ONAPresource",
        "modelCustomizationId": "f7ae574e-fd5f-41e7-9b21-75e001561c96",
        "modelCustomizationName": "vFW-vSINK"
    },
    {
        "id": "31961e27-2a2c-4beb-87c9-bfe0067088f5",
        "version": "2.0",
        "name": "vPkg",
        "instanceName": "vPkg 0",
        "resourceInvariantUUID": "8d8a20c0-746c-4d5e-ala2-fa49fa5786ad",
        "@type": "ONAPresource",
        "modelCustomizationId": "067f5672-51ec-48a6-8ffd-0a7874672666",
        "modelCustomizationName": "vPkg"
    }
],
"href": "servicespecification/1e3feeb0-8e36-46c6-862c-236d9c626439",
"@type": "ONAPservice",
"serviceSpecCharacteristic": [
    {
        "name": "fortigate_image_url",
        "description": null,
        "valueType": "string",
        "@type": "ONAPserviceCharacteristic",
        "required": false,
        "status": null,
        "serviceSpecCharacteristicValue": []
    },
    {
        "name": "flavor",
        "description": null,
        "valueType": "string",
        "@type": "ONAPserviceCharacteristic",
        "required": null,
        "status": null,
        "serviceSpecCharacteristicValue": []
    },
    {
        "name": "external_network_name",
        "description": null,
        "valueType": "string",
        "@type": "ONAPserviceCharacteristic",
        "required": null,
        "status": "inactive",
    }
]

```

```

    "serviceSpecCharacteristicValue": []
  },
  {
    "name": "cpus",
    "description": "Number of CPUs for the server.",
    "valueType": "integer",
    "@type": "ONAPserviceCharacteristic",
    "required": null,
    "status": null,
    "serviceSpecCharacteristicValue": [
      {
        "isDefault": false,
        "value": "1",
        "valueType": "integer"
      },
      {
        "isDefault": true,
        "value": "2",
        "valueType": "integer"
      },
      {
        "isDefault": false,
        "value": "4",
        "valueType": "integer"
      },
      {
        "isDefault": false,
        "value": "8",
        "valueType": "integer"
      }
    ]
  }
]
}

```

**Note:**

If the id specified did not correspond to an existing SDC service, a HTTP 404 code is retrieved.

If for any reason, NBI is able to only retrieve partial information (No Tosca file for example), this retrieved information are send with a HTTP 206 code.

## 4/ Service Order Creation

Send a POST message to /productOrder with the following contents

```

{
  "externalId": "LudONAP001",
  "priority": "1",
  "description": "Ludo first ONAP Order",
  "category": "Consumer",
  "requestedStartDate": "2018-02-28T13:33:37.299Z",
  "requestedCompletionDate": "2018-02-28T13:33:37.299Z",
  "@baseType": null,
  "@type": null,
  "@schemaLocation": null,
  "relatedParty": [
    {

```

```

        "id": "6490",
        "href": null,
        "role": "ONAPcustomer",
        "name": "Jean Pontus",
        "@REFERREDTYPE": "individual"
    }
],
"orderRelationship": null,
"orderItem": [
    {
        "id": "1",
        "action": "add",
        "@type": null,
        "@schemaLocation": null,
        "@baseType": null,
        "orderItemRelationship": [],
        "service": {
            "id": null,
            "href": null,
            "name": "Jean Service",
            "serviceState": "active",
            "@type": null,
            "@schemaLocation": null,
            "serviceCharacteristic": null,
            "serviceRelationship": null,
            "relatedParty": null,
            "serviceSpecification": {
                "id": "1e3feeb0-8e36-46c6-862c-236d9c626439",
                "href": null,
                "name": null,
                "version": null,
                "targetServiceSchema": null,
                "@type": null,
                "@schemaLocation": null,
                "@baseType": null
            }
        }
    }
]
}

```

Some information needs to be highlighted:

```
"externalId": "LudONAP001",
```

This is the service order id from the requester app. It could be used to retrieve the order.

```

"relatedParty": [
    {
        "id": "6490",
        "href": null,
        "role": "ONAPcustomer",
        "name": "Jean Pontus",
        "@REFERREDTYPE": "individual"
    }
]

```

RelatedParty is optional – if it's not provided the service will be instantiated under generic customer (!  
Be sure to have created before 'generic' customer in AAI!)

```
"orderRelationship": null,
```

You can describe relationship between serviceOrder but NBI do not use this to orchestrate fulfillment – that just information.

```
"orderItem": [  
  {  
    "id": "1",
```

It is possible to have a multiple serviceOrder items in one serviceOrder. Just provide distinct id for each of them.

```
"action": "ADD",
```

You could use 3 actions with NBI:

- **add** for new service to add
- **delete** to remove service
- **noChange** to keep service as it is.

Depending on the action, rules apply:

If the action is **add** ....

You **MUST** provide a

```
"serviceSpecification": {  
  "id": "1e3feeb0-8e36-46c6-862c-236d9c626439",
```

That the SDC uuid of the new service

You should provide a

```
"service": {  
  ...  
  "name": "Jean Service",
```

This service name will be use in AAI as instance name

If the action is **delete** or **noChange**

You **MUST** provide a

```
"service": {  
  "id": "e4688e5f-61a0-4f8b-ae02-a2fbde623bcb",
```

This is the idd (in AAI) of the service to be deleted or ...unchanged.

Please note that for this release the

```
"serviceState": "active",
```

Which is the requested service state is **not** taken into consideration.

We manage relationship between serviceOrder items – so if you have following structure:

```
{  
  "externalId": "LudONAP002",  
  ...,  
  "orderItem": [  
    {  
      "id": "1",  
      "action": "add",  
      "@type": null,  
      "@schemaLocation": null,  
      "@baseType": null,  
      "orderItemRelationship": [  
        {  
          "type": "reliesOn",  
          "id": "2"  
        }  
      ]  
    }  
  ]  
}
```



```

    ],
    "service": {
      "id": null,
      ...,
      "serviceSpecification": {
        "id": "98d95267-5e0f-4531-abf8-f14b90031dc5",
        ...
      }
    }
  },
  {
    "id": "2",
    "action": "add",
    "@type": null,
    "@schemaLocation": null,
    "@baseType": null,
    "service": {
      "id": null,
      ...,
      "serviceSpecification": {
        "id": "1e3feeb0-8e36-46c6-862c-236d9c626439",
        ...
      }
    }
  },
]
}

```

NBI will consider that the service described under orderItem “2” **MUST** be provisioned before the one described in the orderItem “1” because this order Item “1” has a relies on to the service described in order item “2”

It is possible to describe Service Characteristic – as an example:

```

"serviceCharacteristic": [
  {
    "name": "flavor",
    "valueType": null,
    "value": {
      "@type": null,
      "@schemaLocation": null,
      "serviceCharacteristicValue": "upgraded"
    }
  },
  {
    "name": "external_network_name",
    "valueType": null,
    "value": {
      "@type": null,
      "@schemaLocation": null,
      "serviceCharacteristicValue": "Jean Pontus Network"
    }
  }
]

```

For this release, SO only accept userParams based on name/value attributes structure for requested parameters: So we only manage characteristic description based as well on name/value.

## 5/ Find Service Order(s)

The operation GET serviceOrder without id allows querying External API to retrieve existing serviceOrder.

For this release, only a subset of criteria is available:

- externalId
- state
- description
- orderDate.gt (orderDate must be greater – *after* -than)
- orderDate.lt (orderDate must be lower-*before* - than)

Send a GET message to

```
https://{api_url}/nbi/api/v1/serviceOrder/?externalId=LudONAP001
```

The response will provide a list of full serviceOrder representation.

If no serviceOrder match the criteria, empty list is retrieved.

## 6/ Retrieve one Service Order

It is possible to use the GET operation to retrieve all information for one serviceOrder.

In order to do that, send a GET message to

```
https://{api_url}/nbi/api/v1/serviceOrder/5ac735fb67567dbdd7c516c7
```

**5ac735fb67567dbdd7c516c7** is the id of a serviceOrder (NBI provided it).

Wait for a response from the server with the following characteristics

- response Code 200-OK (sunny day)
- The body of the response includes one serviceOrder resource referring to its id.

```
{
  "id": "5ac735fb67567dbdd7c516c7",
  "href": "serviceOrder/5ac735fb67567dbdd7c516c7",
  "externalId": "LudONAP001",
  "priority": "1",
  "description": "Ludo first ONAP Order",
  "category": "Consumer",
  "state": "COMPLETED",
  "orderDate": "2018-04-06T08:55:23.934Z",
  "completionDateTime": "2018-04-06T08:55:28.704Z",
  "expectedCompletionDate": null,
  "requestedStartDate": "2018-02-28T13:33:37.299Z",
  "requestedCompletionDate": "2018-02-28T13:33:37.299Z",
  "startDate": null,
  "@baseType": null,
}
```

```

"@type": null,
"@schemaLocation": null,
"relatedParty": [
  {
    "id": "6490",
    "href": null,
    "role": "ONAPcustomer",
    "name": "Jean Pontus",
    "@referredType": "individual"
  }
],
"orderRelationship": null,
"orderItem": [
  {
    "id": "1",
    "action": "ADD",
    "state": "COMPLETED",
    "@type": null,
    "@schemaLocation": null,
    "@baseType": null,
    "orderItemRelationship": [],
    "service": {
      "id": "instanceId",
      "href": null,
      "name": null,
      "serviceState": "active",
      "@type": null,
      "@schemaLocation": null,
      "serviceCharacteristic": null,
      "serviceRelationship": null,
      "relatedParty": null,
      "serviceSpecification": {
        "id": "1e3feeb0-8e36-46c6-862c-236d9c626439",
        "href": null,
        "name": null,
        "version": null,
        "targetServiceSchema": null,
        "@type": null,
        "@schemaLocation": null,
        "@baseType": null
      }
    }
  }
]

```

**Note:**

If the id specified did not correspond to an existing serviceOrder, a HTTP 404 code is retrieved.

## 7/ Find instantiated Service(s)

Send a GET message to

```
https://{api_url}/nbi/api/v1/service/
```

The GET (by list) has following parameter:

id	id of the service instance (inventory)
serviceSpecification.id	id of the service specification (catalog)

serviceSpecification.name	name of the service specification (catalog)
relatedParty.id	if not filled we use 'generic' customer

The API will not send a 404 error if this id did not exist but instead will send an empty list.

The GET response will be:

```
[
  {
    "id": "e4688e5f-61a0-4f8b-ae02-a2fbde623bcb",
    "name": "NewFreeRadius-service-instance-01",
    "serviceSpecification": {
      "name": "vFW",
      "id": "98d95267-5e0f-4531-abf8-f14b90031dc5"
    },
    "relatedParty": {
      "role": "ONAPcustomer",
      "id": "6490"
    }
  }
]
```

Only this subset of data are retrieved

Few remarks to understand the filtering:

1. If a request is send without any parameter, we'll retrieve the list of service-instance for the 'generic' customer:
2. If only customer parameter is filled (relatedParty.id + role= relatedParty'ONAPcustomer') we'll retrieve the list of service-instance for this customer (corresponding to serviceSpecification.name
3. If serviceSpecification.id or name is filled we'll retrieve the list of Service instance (from this service specification) – We'll use the customer id if provided (with Role='ONAPcustomer) or generic if no customer id provided.

## 8/ Retrieve one Service instantiated

It is also possible to use the GET operation to retrieve all information for one service.

In order to do that, send a GET message to

```
https://{api_url}/nbi/api/v1/service/e4688e5f-61a0-4f8b-ae02-a2fbde623bcb
```

**e4688e5f-61a0-4f8b-ae02-a2fbde623bcb** is the id of a service (it corresponds to the AAI service uuid).

Wait for a response from the server with the following characteristics

- response Code 200-OK (sunny day)
- The body of the response includes one service resource referring to its id.

```
{
  "id": "e4688e5f-61a0-4f8b-ae02-a2fbde623bcb",
  "name": "NewFreeRadius-service-instance-01",
```

```
"serviceSpecification": {
  "id": "98d95267-5e0f-4531-abf8-f14b90031dc5",
  "invariantUUID": "709d157b-52fb-4250-976e-7133dff5c347",
  "@type": "ONAPservice"
},
"supportingResource": [
  {
    "id": "cb80fbb6-9aa7-4ac5-9541-e14f45de533e",
    "name": "NewFreeRadius-VNF-instance-01",
    "status": "PREPROV",
    "modelInvariantId": "f5993703-977f-4346-a1c9-c1884f8cfd8d",
    "modelVersionId": "902438f7-1e4c-492d-b7cc-8650e13b8aeb",
    "@referredType": "ONAP resource"
  },
  {
    "id": "cb80fbb6-9aa7-4ac5-9541-e14f45de533e",
    "name": "NewFreeRadius-VNF-instance-01",
    "status": "PREPROV",
    "modelInvariantId": "f5993703-977f-4346-a1c9-c1884f8cfd8d",
    "modelVersionId": "902438f7-1e4c-492d-b7cc-8650e13b8aeb",
    "@referredType": "ONAP resource"
  }
],
"@type": "serviceONAP",
"hasStarted": "yes",
"type": "service-instance",
"relatedParty": {
  "role": "ONAPcustomer",
  "id": "6490"
}
}
```