

ComputeNext Service provider (CNSP) 1.0 (working copy)

Introduction

ComputeNext marketplace enables cloud services to be provisioned from geographically diverse cloud providers operating on one or more cloud platforms. This document describes the API that a cloud service provider should support to enable fulfilment and life cycle management of the service catalog and consumed through the ComputeNext Cloud Brokerage Platform.

Terminology

Cloud Brokerage: Cloud Brokerage is the framework that enables the search, discovery, fulfilment, managing and tracking the efficient use of cloud services for a customer's service need.

Cloud Service Provider: A vendor that offers a service in the cloud computing paradigm with service delivered in IaaS, PaaS or SaaS models.

ComputeNext Service Provider (CSP): A cloud service provider onboarded to offer their services via appropriate APIs to the ComputeNext platform

ComputeNext Service Provider (CNSP) API: A REST interface specification to describe the services and its artifacts offered by a Cloud Service Provider

CNSP-compatible service endpoint: ComputeNext compatible endpoint defines the contract between the CNSP connector and the Service Provider.

ComputeNext Service provider (CNSP) API

In the current implementation, To on-board a service provider in ComputeNext marketplace a connector is to be available / developed. The connector will be the mediator between the ComputeNext system and the service provider platform. The connector understands the ComputeNext System API request , convert the request to service provider compatible so that the service provider understands and perform the appropriate action.

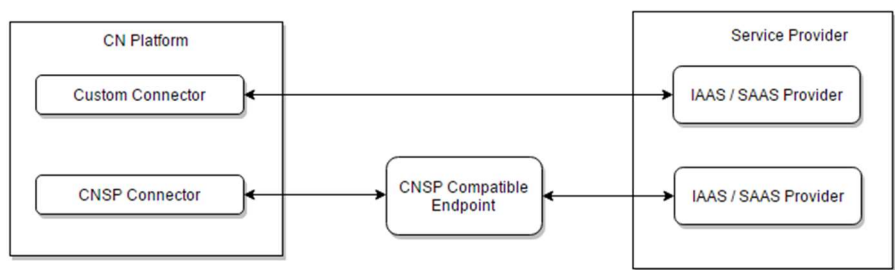
Once the connector for a service provider is developed, the offers supported by the SP need to be added to the system. In the current implementation this is done by adding the catalog details to JSON file, this JSON datas are pushed to the system.

With the current implementation there is no way a service provider can get them on-boarded to ComputeNext marketplace themselves without the help of ComputeNext team.

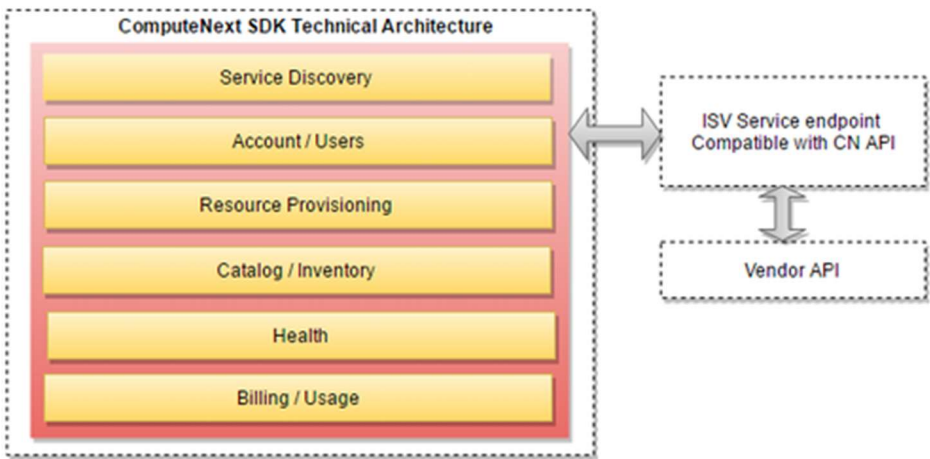
Computenext Service provider connector(CNSP) enable a way for service providers to get themselves on-boarded without the help of ComputeNext Team.

CNSP connector contains cloud abstracted API, wrapper around a large number of cloud service providers, abstracting the difference. The service providers will need to develops API endpoints(CNSP Compatible endpoints) in order to understand the API calls from CNSP connector.

Architecture



CNSP Connector



Below are services API that are implemented in CNSP API, the cloud service provider should develop the CNSP compatible API endpoints to listen from the below service APIs.

Category	API	Descriptions
----------	-----	--------------

Service discovery API	/apiv1/discovery	When a CSP registers their endpoint with ComputeNext marketplace, the service discovery API is used to obtain metadata information about the states and supported actions of the different types of services and their artifacts offered by the CSP to the Brokerage platform.
Account / Users Management API	/apiv1/account	API methods to create and manage account and user life cycle in the CSP's cloud services.
Resource provisioning	/apiv1/resource	Cloud Resource refers to services and artifacts offered by the CSP. The API methods enable the creation and lifecycle management of the cloud resources in a CSP.
Catalog / Inventory	/apiv1/catalog	API methods to get attributes and properties of the particular cloud service and artifacts offered by the CSP along with the service inventory made available to the marketplace. These attributes include human-readable informational data presented to the customer, parameters/options selectable by the customer to use variations of the service and metadata to configure the service during different lifecycle events (create, start, stop, upgrade, terminate, etc.)
Health	/apiv1/health	API methods to provide service endpoint health as well as health of service and artifacts offered via the Monitors the services availability and the resource endpoints
Billing / Usage	/apiv1/billing	Helps to get the billing and usage details from CSP.

Authentication

CNSP connector authenticates with the CNSP compatible API endpoint using HTTP basic authentication (the Authorization: header) on every request and will reject any request that do not contain a username and password. The CNSP compatible API endpoint is responsible for checking the username and password and returning a 401 Unauthorized message if credentials are invalid.

Request / Response

When SP endpoint receives a request from CNSP connector, it should synchronously perform the necessary action on the resource and respond to the request.

Request structure

Account / Users Management

Action	API	Sample
--------	-----	--------

Create Account	POST /account	<p>Request</p> <pre>{ "accountid": "[accountid]", "accountname": "[accountname]", "phone": "[companyphone]", "userinfo": { "firstname": "[firstname]", "lastname": "[lastname]", "email": "[email]", "password": "[autogenerated]", "role": "[role]", "phone": "[personalphone]" }, "address": { "addressline1": "[addressline1]", "addressline2": "[addressline1]", "city": "[city]", "state": "[state]", "postalcode": "[postalcode]", "country": "[country]" }, "additionalattributes": { "attributename": "[attributeval]", "attributename": "[attributeval]" } }</pre> <p>Success Response</p> <pre>{ "result": { "providerresponse": { "accountid": "[accountid]", "provideraccountid": "[provideraccountid]", "respcode": 200 }, "success": true, "message": "Account created successfully" } }</pre> <p>Error Response</p> <pre>{ "result": { "providerresponse": { "errormessage": "Invalid credentials provided.", "respcode": 401 }, "success": false, "message": "Authentication failed" } }</pre>
----------------	---------------	---

		<p>Or</p> <pre>{ "result":{ "providerresponse":{ "errorMessage":"User with email address already exists.", "respcode" :400 }, "success":false, "message":"Account creation failed" } }</pre>
--	--	--

Update Account	PUT /account	<p>Request</p> <pre>{ "accountname":"[accountname]", "phone":"[companyphone]", "address":{ "addressline1":"[addressline1]", "addressline2":"[addressline1]", "city":"[city]", "state":"[state]", "postalcode":"[postalcode]", "country":"[country]" }, "additionalattributes":{ "attributename":"[attributeval]", "attributename":"[attributeval]" }, "requestor":{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]", "userid":"[userid]" } }</pre> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]", "respcode":200 }, "success":true, "message":"Account updated successfully" } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "errormessage":"The account doesn't exist.", "respcode" :404 }, "success":false, "message":"Account updation failed" } }</pre>
----------------	--------------	--

Delete Account	DELETE /account	<p>Request</p> <pre>{ "requestor":{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]", "userid":"[userid]" } }</pre> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]", "respcode":200 }, "success":true, "message":"Account deleted successfully" } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "errormessage":"The account doesn't exist.", "respcode" :404 }, "success":false, "message":"Account deletion failed" } }</pre>
----------------	-----------------	--

Get an Account	GET /account/:provideraccountid	<p>Request Get accountid from URL param</p> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "accountinfo":{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]" "accountname":"[accountname]", "phone":"[companyphone]", "address":{ "addressline1":"[addressline1]", "addressline2":"[addressline1]", "city":"[city]", "state":"[state]", "postalcode":"[postalcode]", "country":"[country]" }, "additionalattributes":{ "attributename":"[attributeval]", "attributename":"[attributeval]" } }, "success":true, "message":"Account retrieved successfully" } } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "errormessage":"The user is not authorized.", "respcode" :500 }, "success":false, "message":"Account retrieval failed" } }</pre>
----------------	------------------------------------	---

Get All Accounts	GET /account	<p>Request</p> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode" :200, "accounts":[{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]" "accountname":"[accountname]", "phone":"[companyphone]", "address":{ "addressline1":"[addressline1]", "addressline2":"[addressline1]", "city":"[city]", "state":"[state]", "postalcode":"[postalcode]", "country":"[country]" }, "additionalattributes": { "attributename":"[attributeval]", "attributename":"[attributeval]" } }, { "accountid":"[accountid]", "provideraccountid":"[provideraccountid]" "accountname":"[accountname]", "accounttype":"[accounttype]", "phone":"[companyphone]", "address":{ "addressline1":"[addressline1]", "addressline2":"[addressline1]", "city":"[city]", "state":"[state]", "postalcode":"[postalcode]", "country":"[country]" }, "additionalattributes":{</pre>
------------------	--------------	---

		<pre> "attributename":"[attributeval]", "attributename":"[attributeval]" }] }, "success":true, "message":"Accounts retrieved successfully" } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "errormessage":"The user is not authorized.", "respcode" :500 }, "success":false, "message":"Account retrieval failed" } }</pre>
--	--	---

Create User	POST /account/user	<p>Request</p> <pre>{ "firstname":"[firstname]", "lastname":"[lastname]", "email":"[email]", "password":"[autogenerated]", "phone":"[personalphone]" "role":[role], "additionalattributes":{ "attributename":"[attributeval]", "attributename":"[attributeval]" }, "requestor":{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]" } }</pre> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":201, "userid":"[userid]", "provideruserid":"[provideruserid]" }, "success":true, "message":"User created successfully" } }</pre> <p>Error Response</p> <pre>{ "result": { "providerresponse":{ "respcode":400, "errormessage":"The email address already exists" }, "success":false, "message":"User creation failed" } }</pre>
-------------	--------------------	--

Update user	PUT /account/user	<p>Request</p> <pre>{ "firstname":"[firstname]", "lastname":"[lastname]", "phone":"[personalphone]", "role":"[role]", "additionalattributes":{ "attributename":"[attributeval]", "attributename":"[attributeval]" }, "requestor":{ "accountid":"[accountid]", "provideraccountid":"[provideraccountid]", "userid":"[userid]" } }</pre> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "userid":"[userid]", "provideruserid":"[provideruserid]" }, "respcode":200 }, "success":true, "message":"User updated successfully" }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "errormessage":"The user doesn't exist.", "respcode" :404 }, "success":false, "message":"User updation failed" } }</pre>
-------------	-------------------	--

Other action on user	PUT /account/user/resource	<p>Assign resource to user, un-assign resource to user</p> <p>Request</p> <pre>{ "action": "assign", "instanceinfo": { "providerinstanceid": "[providerinstanceid]" }, "requestor": { "accountid": "[accountid]", "userid": "[userid]", "provideraccountid": "[provideraccountid]" } }</pre> <p>//to revoke</p> <pre>{ "action": "revoke", "instanceinfo": { "providerinstanceid": "[providerinstanceid]" }, "requestor": { "accountid": "[accountid]", "userid": "[userid]", "provideraccountid": "[provideraccountid]" } }</pre> <p>//If any parameters based on provider ex: if suspending send seatid etc..</p> <p>Response</p> <pre>{ "result": { "providerresponse": { "respcode": 200, "providerinstanceid": "[providerinstanceid]" }, "success": true, "message": "Subscription entitled successfully" } }</pre> <p>Or</p> <pre>{ "result": { "providerresponse": { "respcode": 200, "providerinstanceid": "[providerinstanceid]" }, "success": true, "message": "The given subscription revoked successfully for the subscriber" } }</pre>
----------------------------	----------------------------	---

		<pre>} } Error Response { "result":{ "providerresponse":{ "respcode":404, "errorMessage":"The subscriber already has a seat for the subscription." }, "success":false, "message":"Entitling of subscription failed" } } Or { "result":{ "providerresponse":{ "respcode":403, "errorMessage":"The user is not authorized to perform this operation." }, "success":false, "message":"Revoking of the given subscription for the subscriber failed" } } }</pre>
--	--	--

Delete User	DELETE /account/user	<p>Request</p> <pre>{ "requestor":{ "userid":"[userid]" } }</pre> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "userid":"[userid]" }, "success":true, "message":"User profile deleted successfully" } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "errorMessage":"The user doesn't exist.", "respcode" :404 }, "success":false, "message":"User profile deletion failed" } }</pre>
-------------	----------------------	--

Get a user	GET /account/user/:provideruserid	<p>Request Get userid from URL param</p> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "userinfo":{ "provideruserid":"[provideruserid]", "firstname":"[firstname]", "lastname":"[lastname]", "email":"[email]", "phone":"[personalphone]" "role":[role], "additionalattributes":{ "attributename":"[attributeval]", "attributename":"[attributeval]" } }, "success":true, "message":"User retrieved successfully" } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":403, "errormessage":"The user is not authorized to perform this operation." }, "success":false, "message":"The user is not authorized to perform this operation." } }</pre>
------------	--------------------------------------	---

Get All user	GET /account/user/:provideraccountid	<p>Request</p> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "users":[{ "provideruserid":"[provideruserid]", "firstname":"[firstname]", "lastname":"[lastname]", "email":"[email]", "phone":"[personalphone]" "role":[role], "additionalattributes":{ "attributename":"[attributeval]", "attributename":"[attributeval]" } }, { "provideruserid":"[provideruserid]", "firstname":"[firstname]", "lastname":"[lastname]", "email":"[email]", "username":"[username]", "phone":"[personalphone]" "role":[role], "additionalattributes":{ "attributename":"[attributeval]", "attributename":"[attributeval]" } }] }, "success":true, "message":"Users retrieved successfully" }</pre>
--------------	--------------------------------------	---

		<pre>} Error Response { "result": { "providerresponse":{ "respcode":403, "errormessage":"The user is not authorized to perform this operation." }, "success":false, "message":"The user is not authorized to perform this operation." } }</pre>
--	--	--

Resource Provisioning

Action	API	Sample
--------	-----	--------

Create a Resource	POST /resource	<div><div>Request</div><div><pre>{ "requestid":"[requestid]", "action":"create", "resource":{ "type":"saas" }, "parameters":{ "sku":"[sku]", "licenseQuantity":[numberoflicense], "customoption1":{"title":"customoption1title"}, "customoption2":{"title":"customoption2title"}, "additionalparameters":{ "parametername":"[parameterinterval]", "parametername":"[parameterinterval]" } }, "requestor":{ "accountid":"[accountid]", "userid":"[userid]", "provideraccountid":"[provideraccountid]", "accountname":"[accountname]" } }</pre></div><div>Success Response</div><div><pre>{ "result":{ "providerresponse":{ "providerinstanceid":"[providerinstanceid]", "username":"[username]", "password":"[password]", "[respparam1]":{ "label":"respparam1label", "value":"respparam1value" }, "[respparam2]":{ "label":"respparam2label", "value":"respparam2value" }, "status":"active", "respcode":200 }, "success":true, "message":"Resource created successfully", } }</pre></div><div>Error Response</div><div><pre>{ "result":{ "providerresponse":{ "respcode":403, "errormessage":"The user is not authorized" }, "success":false,</pre></div></div>
-------------------	----------------	---

		<pre>"message":"The user is not authorized to perform the operation" } }</pre>
Update resource details (Number of licenses)	PUT /resource	<pre>Request { "requestId":"[requestid]", "action":"update", "resource":{ "type":"saas" }, "instanceinfo":{ "providerinstanceid":"[providerinstanceid]" }, "parameters":{ "license":[numberoflicense], "customparam1":"[customparamval1]", "customparam2":"[customparamval2]" }, "additionalparameters":{ "parametername":"[parameterval]", "parametername":"[parameterval]" }, "requestor":{ "accountid":"[accountid]", "userid":"[userid]", "provideraccountid":"[provideraccountid]" } } Success Response { "result":{ "providerresponse":{ "providerinstanceid":"[providerinstanceid]", "respcode" :200 }, "success":true, "message":"Resource details updated successfully" } } Error Response { "result":{ "providerresponse":{ "errormessage":"The user is not authorized to perform the operation", "respcode":500 }, "success":false, "message":"Resource details updation failed" } }</pre>

<p>Additional Action on Resource</p>	<p>PUT /resource</p>	<p>Request</p> <pre>{ "requestId":"[requestid]", "action":"update.suspend", "resource":{ "type":"saas" }, "instanceinfo":{ "providerinstanceid":"[providerinstanceid]" }, "additionalparameters":{ "parametername":"[parametername]", "parametername":"[parametername]" } "requestor":{ "accountid":"[accountid]", "userid":"[userid]", "provideraccountid":"[provideraccountid]" } }</pre> <p>//or</p> <pre>{ "requestId":"[requestid]", "action":"update.reactivate", "resource":{ "type":"saas" }, "instanceinfo":{ "providerinstanceid":"[providerinstanceid]" }, "additionalparameters":{ "parametername":"[parametername]", "parametername":"[parametername]" } "requestor":{ "accountid":"[accountid]", "userid":"[userid]", "provideraccountid":"[provideraccountid]" } }</pre> <p>Success Response</p> <pre>{ "result": { "providerresponse":{ "respcode":204, "providerinstanceid":"[providerinstanceid]" }, "success":true, "message":"Resource suspended successfully" } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{</pre>
--------------------------------------	----------------------	--

		<pre>"respcode":404, "errormessage":"Resource not found", "providerinstanceid":"[providerinstanceid]" }, "success":false, "message":"Suspension of resource failed" } }</pre>
Get Resource/instance details	GET /resource/:providerinstanceid	<p>Request Get providerinstanceidfrom request parameters</p> <p>Success Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "resourceinfo":{ "resource":{ "type":"saas" }, "parameters":{ "providerinstanceid":"[providerinstanceid]", "in "license":[numberoflicens "cu "cu "status":"[pending/active/suspended/cancel "startdate":"[subscriptionstartdate]", "en }, "additional "pa "pa } }, "success":true, "message":"Resource details retr } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":403, "errormessage":"The user is not authorized }, "success":false, "message":"The user is not authorized to perf } }</pre>

Get All Resouces	GET /resource/:provideraccountid	<p>Request Get acccoutid from request parameters</p> <p>Success Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "resources":[{ "resource":{ "type":"saas" }, "parameters":{ "providerinstanceid":"[providerinstanceid]" "in "license":[numberoflicens }, "additional "pa "pa } }, { "resource":{ "type":"saas" }, "parameters":{ "providerinstanceid":"[providerinstanceid]" "in "license":[numberoflicens }, "additional "pa "pa } }] }, "success":true, "message":"Resources retrieved suc } }</pre> <p>Error Response</p> <pre>{ "result": { "providerresponse":{ "respcode":403, "errormessage":"The user is not authorized }, "success":false, "message":"The user is not authorized to perf } }</pre>
------------------	-------------------------------------	--

Delete a resource	DELETE /resource	<p>Request</p> <pre>{ "requestId":"[requestid]", "action":"delete", "instanceinfo":{ "providerinstanceid":"[providerinstanceid]" } "requestor":{ "provideraccountid":"[provideraccountid]", "provideruserid":"[provideruserid]" } }</pre> <p>Success Response</p> <pre>{ "result":{ "providerresponse": { "providerinstanceid":"[providerinstanceid]", "respcode" :200 }, "success":true, "message":"Resource deleted successfully" } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "errormessage":"The user is not a valid user", "respcode" :500 }, "success":false, "message":"Resource deletion failed" } }</pre>
-------------------	------------------	---

Catalog

Action	API	Sample
--------	-----	--------

Retrieve all offerings	GET /catalog This endpoint will pull all resource types	<p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "resources":[{ "resource":{ "type":"saas", "fulfillmenttype":"[fulfillmenttype]", "vendor":"[vendor]", "region":"[region]", "isaccountrequired":true/false }, "parameters":{ "sku":"[productsku]", "name":"[product name]", "description":"[description with htmltags]", "shortdescription":"[short description]", "category":"[category]", "setproductasnew":{"fromdate":"[fromdate]","todate":"[todate]","metatitle":"[metatitle]","metakeywords":"[metakeywords]","metadescription":"[metadescription]","saasurl":"[saasurl]","faqurl":"[faqurl]","minpurchasequantity":"[minpurchasequantity]","maxpurchasequantity":"[maxpurchasequantity]","autorenew":true/false,"plantype":"[palntype]","isenabled":true/false }, "additionalparameters":{ "parametername":"[parameterinterval]","parametername":"[parameterinterval]","chargeattributes":[{"chargeid":"[chargeid]","chargename":"Users","description":"Per user charge details"}],"subscription serviceterm":{"costprice":{"billingperiodfrequency":1,"billingperiodunit":"month","chargetype":"[chargetype]","tierpricing":[{"sequencenumber":1,"startingvalue":0,"endingvalue":null,</pre>
------------------------	--	--

		<pre> "chargeamount":"[cost price]", "chargeamountunit":"/mo", "currencycode":"usd", "isdefault":1, "isenabled":1 }] }, "sellprice":{ "billingperiodfrequency":1, "billingperiodunit":"month", "chargetype":"[chargetype]", "tierpricing":[{ "sequencenumber":1, "startingvalue":0, "endingvalue":null, "chargeamount":"[sell price]", "chargeamountunit":"/mo", "currencycode":"usd", "isdefault":1, "isenabled":1 }] }, "erpprice":{ "billingperiodfrequency":1, "billingperiodunit":"month", "chargetype":"[chargetype]", "tierpricing":[{ "sequencenumber":1, "startingvalue":0, "endingvalue":null, "chargeamount":"[erp price]", "chargeamountunit":"/mo", "currencycode":"usd", "isdefault":1, "isenabled":1 }] } }] }, "success":true, "message":"Catalog retrieved successfully" } } Error Response { "result":{ "providerresponse":{ "respcode":403, "errorMessage":"The user is not authorized." }, "success":false, "message":"The user is not authorized. Unable to retrieve catalogue" } }</pre>
--	--	---

		}
--	--	---

Billing

Action	API	Sample
--------	-----	--------

Usage feed by accountid	GET /billing/:provideraccountid	<p>Request accountid</p> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "accountid":"[accountid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00", "usagefeed":[{ "subscriptionid":"[subscriptionid]", "instanceid":"[instanceid]", "chargeid":"[chargeid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00", "object_type":"UsageRecord", "meter_name":"Users", "meter_region":"[region]", "meter_category":"[category]", "unit":"/mo", "info_fields":{"fieldname":"fieldvalue"}, "quantity":[quantity] }, { "subscriptionid":"[subscriptionid]", "instanceid":"[instanceid]", "chargeid":"[chargeid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00",</pre>
-------------------------	---------------------------------	---

		<pre> "object_type": "UsageRecord", "meter_name": "Users", "meter_region": "[region]", "meter_category": "[category]", "unit": "/mo", "info_fields": {"fieldname": "fieldvalue"}, "quantity": [quantity] }], "success": true, "message": "Usage data retrieved successfully" } } Error Response { "result": { "providerresponse": { "respcode": 403, "errormessage": "The user is not authorized." }, "success": false, "message": "The user is not authorized." } }</pre>
--	--	---

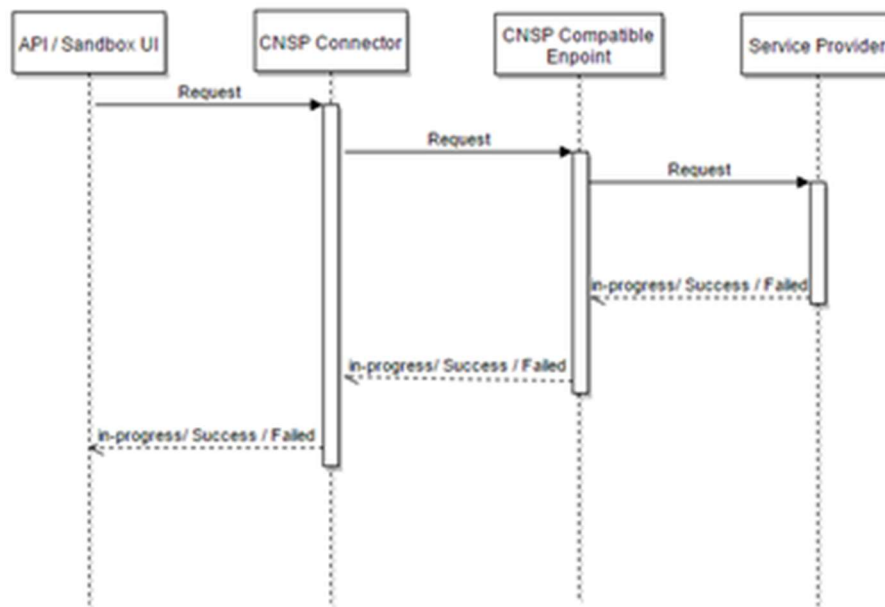
Usage feed by subscriptionid	GET /billing/:subscriptionid	<p>Request subscriptionid</p> <p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "accountid":"[accountid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00", "usagefeed":[{ "subscriptionid":"[subscriptionid]", "instanceid":"[instanceid]", "chargeid":"[chargeid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00", "object_type":"UsageRecord", "meter_name":"Users", "meter_region":"[region]", "meter_category":"[category]", "unit":"/mo", "info_fields":{"fieldname":"fieldvalue"}, "quantity":[quantity] }, { "subscriptionid":"[subscriptionid]", "instanceid":"[instanceid]", "chargeid":"[chargeid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00",</pre>
------------------------------------	---------------------------------	--

		<pre> "object_type": "UsageRecord", "meter_name": "Users", "meter_region": "[region]", "meter_category": "[category]", "unit": "/mo", "info_fields": {"fieldname": "fieldvalue"}, "quantity": [quantity] }], }, "success": true, "message": "Usage data retrieved successfully" }</pre> <p>Error Response</p> <pre>{ "result": { "providerresponse": { "respcode": 403, "errorcode": "BZSSS1452E", "errormessage": "The user is not authorized." }, "success": false, "message": "The user is not authorized." } }</pre>
--	--	---

Usage feed	GET /billing/:providerinstanceid	<p>Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":200, "accountid":"[accountid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00", "usagefeed":[{ "instanceid":"[instanceid]", "chargeid":"[chargeid]", "usage_start_time":"2016-08-01T00:00:00+00:00", "usage_end_time":"2016-09-01T00:00:00+00:00", "object_type":"UsageRecord", "meter_name":"Users", "meter_region":"[region]", "meter_category":"[category]", "unit":"/mo", "info_fields":{"fieldname":"fieldvalue"}, "quantity":[quantity] }], "success":true, "message":"Usage data retrieved successfully" } } }</pre> <p>Error Response</p> <pre>{ "result":{ "providerresponse":{ "respcode":403, "errormessage":"The user is not authorized." }, "success":false, "message":"The user is not authorized." } }</pre>
------------	-------------------------------------	--

		}
		}

The service provider can build a prototype application to practice using the operation and test the code from the sandbox environment. The Sandbox environment has few API exposed, through which the Service provider can simulate API request to the CNSP compatible endpoint through the CNSP connector before going to the live environment.



Steps for On-boarding a vendor via CNSP Connector ([Vendor Onboaring via CNSP \(v1.0\)](#))

1. Admin to create a vendor account and admin user (With default role "vendor")
2. Vendor login to admin portal

3. Create endpoint
4. Import products via API (or) Add products by selecting the endpoint via which the product should be provisioned
5. Products will be displayed in marketplace
6. Customers can add to Workload/Order and provision
7. Support subscription management operations from marketplace via Endpoint – Suspend, Reactivate, Delete