

The BigBitBus API Documentation

Revision v8.0

Date 08/07/2020

License: Copyright BigBitBus Inc. All rights reserved.

Table of Contents

- [BigBitBus API Version](#)
 - [1.1. Authorization](#)
 - [1.2. Applications](#)
 - [1.2.1. Create an application](#)
 - [1.2.2. Retrieving Applications](#)
 - [1.2.3. Obtaining a report on an Application](#)
 - [1.2.4. Querying Applications](#)
 - [1.2.5. Finding Application Alternatives on Another Provider](#)
 - [1.2.6. VM Optimization](#)
 - [1.2.7. Updating an Application](#)
 - [1.2.8. Deleting an Application](#)
 - [1.2.9. Deleting multiple Applications](#)
 - [1.2.10. Listing all Applications](#)
 - [1.2.10.1. Filtering and sorting Applications](#)
 - [1.3. Prices](#)
 - [1.3.1. Creating a new price for a service \(POST\)](#)
 - [1.3.2. Get price details about a pre-existing price \(GET\)](#)
 - [1.3.3. Updating the costperunit for a pre-existing price \(PUT\)](#)
 - [1.3.4. Deleting a price \(DELETE\)](#)
 - [1.3.5. Deleting multiple prices \(POST\)](#)
 - [1.3.6. Listing all prices \(GET\)](#)
 - [1.3.6.1. Filtering and sorting prices](#)
 - [1.4. ServiceTypes](#)
 - [1.4.1. Create a new service type \(POST\)](#)
 - [1.4.2. Retrieve a service type \(GET\)](#)
 - [1.4.3. Deleting a service type \(DELETE\)](#)
 - [1.4.4. Deleting multiple service types \(POST\)](#)
 - [1.4.5. List all service types \(GET\)](#)
 - [1.4.5.1. Filtering and sorting service types](#)
 - [1.4.6. Finding service types](#)
 - [1.4.6.1. Service type keyword search](#)
 - [1.4.6.2. Service type name autocomplete](#)
 - [1.4.6.3. Getting service type info](#)
 - [1.4.6.4. Compare two service types](#)
 - [1.5. ProviderDiscounts](#)
 - [1.5.1. Create a new discount \(POST\)](#)
 - [1.5.2. Retrieve a discount \(GET\)](#)
 - [1.5.3. Updating the discount amount for a pre-existing discount \(PUT\)](#)

- 1.5.4. Deleting a discount (DELETE)
- 1.5.5. Listing all discounts
 - 1.5.5.1. Filtering and sorting discounts
- 1.6. Providers
 - 1.6.1. Create a new provider (POST)
 - 1.6.2. Retrieve a provider (GET)
 - 1.6.3. Using providers
 - 1.6.3.1. Creating a service type with the provider
 - 1.6.3.2. Creating a price with the provider
 - 1.6.3.3. Reports and Optimization
 - 1.6.4. Delete a provider (DELETE)
 - 1.6.5. Deleting multiple providers (POST)
 - 1.6.6. Listing all providers (GET)
 - 1.6.6.1. Filtering and sorting providers
- 1.7. Shares
 - 1.7.1. Create a new share (POST)
 - 1.7.2. Retrieve a share (GET)
 - 1.7.3. Updating the share (PUT)
 - 1.7.4. Deleting a share (DELETE)
 - 1.7.5. Deleting multiple shares (POST)
 - 1.7.6. Listing all shares (GET)
 - 1.7.6.1. Filtering and sorting shares (GET)

Concepts

Cloud providers offer service types to users, who in turn compose applications from one or more service types. For example, a user will create a nosql database, consisting of multiple VMs, spread across different data centers (cloud regions). BigBitBus API allows users to store information about such applications in its database. This information is mapped to the cloud pricing and performance information (about different service types on offer by cloud providers).

Users create applications in the database (using the API). They can then query different BigBitBus API endpoints to get pricing information about the application, which analogous service types can be used to build a similar application on another cloud provider, comparing costs and performance, etc. Before we get into specifics of the API, we define a few terms which will be used throughout the documentation.

Definitions

- **Provider** A provider is either a public cloud provider, like AWS or Azure, or a in-house on-prem cloud.
- **ServiceType** A servicetype is an offering that a cloud provider offers. For example, AWS provides the "aws-t3.small" VM instance type.
- **Location** Location refers to cloud provider regions. For example, "us-east-2" and "ukwest" are regions where AWS and Azure provide services. ServiceTypes are unique to a location because cloud providers may not offer some servicetypes in some locations.
- **Application** An application describes a cluster or set of cloud services used together - for example, 30 virtual machines, spread across 3 data centers, that comprise a No-SQL database. An application is a set of one more more ServiceInstances. A ServiceInstance comprises of a ServiceType and arbitrary attributes that describe characteristics of how the ServiceType is used in the application. Here is an example of an Application, named "Enterprise_HR_SAAS_Application" that has two ServiceInstances. The first ServiceInstance uses the "aws-c5-xlarge" ServiceType and has several attributes - for example - the "quantity" is set to 8. The second ServiceInstance use the "aws-c5-2xlarge" ServiceType and has the quantity value 3. You can add any arbitrary <attr_id,value> pair to applications. Some of these attributes (`util`, `max_util`, and `quantity`) have special meaning and are used by the BigBitBus comparison and recommendation engine.

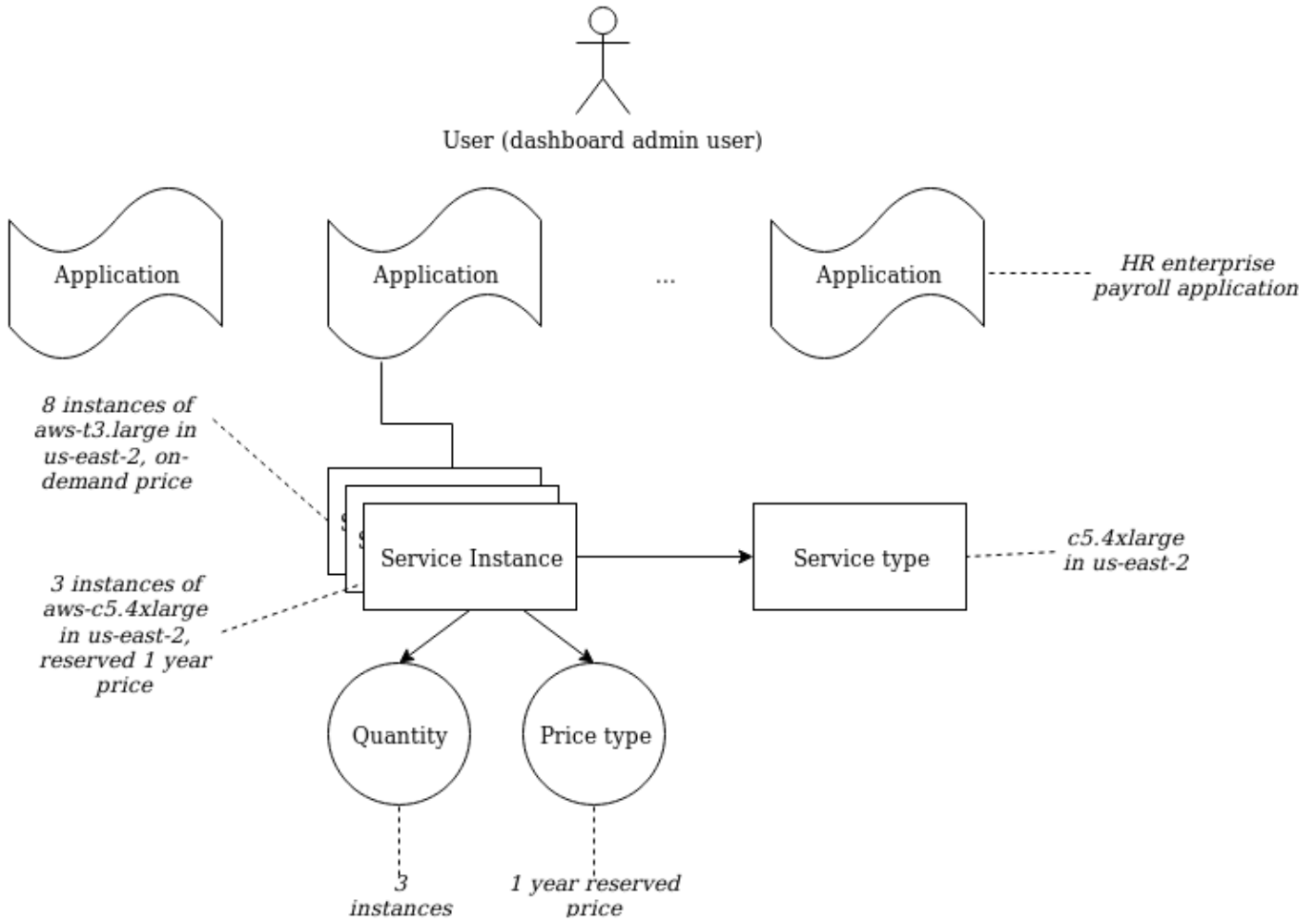
```
{
  "app_name": "Enterprise_HR_SAAS_Application",
  "services": [
    {
      "service_type": "aws-c5.xlarge",
      "location": "us-west-2",
      "price_type": "onDemandPrice",
      "provider": "aws",
      "attributes": [
        {
          "attr_id": "unit",
          "value": "instance"
        },
        {
```

```
    "attr_id": "util",
    "value": "40"
  },
  {
    "attr_id": "max_util",
    "value": "80"
  },
  {
    "attr_id": "quantity",
    "value": "8"
  }
]
},
{
  "service_type": "aws-c5.2xlarge",
  "location": "us-east-2",
  "price_type": "onDemandPrice",
  "provider": "aws",
  "attributes": [
    {
      "attr_id": "unit",
      "value": "instance"
    },
    {
      "attr_id": "quantity",
      "value": "3"
    }
  ]
}
]
```

- **PriceType** Cloud providers offer ServiceTypes at different prices to different customers. Users have the ability to create a custom price for any ServiceType or choose among the standard prices offered by the cloud provider - these are usually "onDemandPrice", "reserved1yearPrice", and "reserved3yearPrice".

Model

Next, lets visualize the data models being used under the hood.



In Fig.1 a user defines an application, which may comprise of multiple service instances. Each service instance is composed of one or more service type instance (quantity), along with properties like which price type is being used to pay for these services. The italicized text illustrates these concepts via examples. Users define applications via the API.

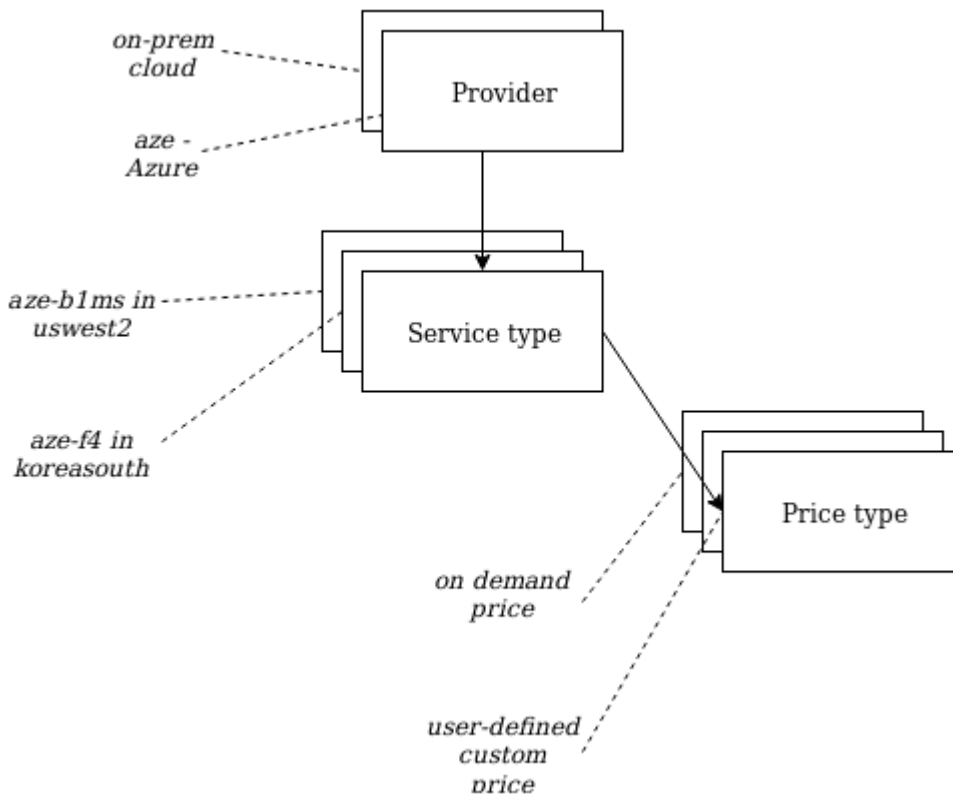


Fig. 2 shows providers that contain multiple service types, which in turn may have multiple price types associated with them. This data is loaded and updated in the BigBitBus database by the administrator.

BigBitBus API Version 1

This Document shows examples of making REST calls to the BigBitBus API version 1 (this is the current and only version as of now).

Many examples are used to illustrate the API below. All examples assume the BigBitBus API server is running at `http://127.0.0.1` and at port 8000.

1.1. Authorization

Use the authorization credentials (username, password) to get access to the API.

```
curl --request POST \  
  --url http://127.0.0.1:8000/api-token-auth/login/ \  
  --header 'Accept: */*' \  
  --header 'content-type: multipart/form-data' \  
  --form username=examplecorpuser \  
  --form password=securepassword
```

The command will return a Java Web Token (JWT) `token`, its string value is used in subsequent API requests.

```
{ "token": "eyJ0eXAiOiJKV1QiLCJhbhasdas3qJIUzI1NiJ9.eyJ1c2VyX2lkIjoyMywidXNlcm5hbWUiOiJleGFtcGxlY29ycHVzZXIiLCJleHAiOjE1NjkwOTU4NzEsImVtYWlsIjoiIn0.ow2iT6d99Bypdsux5wvzFpR6IqVU" }
```


1.2. Applications

1.2.1. Create an application

Send a **POST** request to create an application in the database.

```
curl -X POST \  
  http://127.0.0.1:8000/api/v1.0/calc/applications/ \  
  -H 'Accept: */*' \  
  -H 'Accept-Encoding: gzip, deflate' \  
  -H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imd1bm  
VyaWMiLCJleHAiOiJlNjkwOTQ5OTksImVtYWlsIjoiIn0.wLUVsUtZMRwLj6bvRK5dWAFmUTwYP  
NMbGpuHmmO9mXY' \  
  -d '{  
    "app_name": "Enterprise_HR_SAAS_Application",  
    "description": "Finance department application",  
    "metadata": "this can be any string up to 1024 characters long, for  
example, fookey=barvalue",  
    "services": [  
      {  
        "service_type": "aws-c5.xlarge",  
        "location": "us-west-2",  
        "price_type": "onDemandPrice",  
        "provider": "aws",  
        "attributes": [  
          {  
            "attr_id": "unit",  
            "value": "instance"  
          },  
          {  
            "attr_id": "util",  
            "value": "40"  
          },  
          {  
            "attr_id": "max_util",  
            "value": "80"  
          },  
          {  
            "attr_id": "quantity",  
            "value": "8"  
          }  
        ]  
      },  
      {  
        "service_type": "aws-c5.2xlarge",  
        "location": "us-east-2",  
        "price_type": "onDemandPrice",  
        "provider": "aws",  
        "attributes": [  
          {
```

```

        "attr_id": "unit",
        "value": "instance"
    },
    {
        "attr_id": "util",
        "value": "30"
    },
    {
        "attr_id": "max_util",
        "value": "80"
    },
    {
        "attr_id": "quantity",
        "value": "3"
    }
]
}
]
}'

```

The `metadata` and `description` keys are optional strings (each up to 1024 characters long). If not set, they default to empty string `""`.

Tables for `ServiceTypes` and `PriceTypes` are available by navigating to [http://127.0.0.1:8000/data/\[services,prices\]](http://127.0.0.1:8000/data/[services,prices]).

1.2.2. Retrieving Applications

`GET` requests can be used to read created applications:

1. Get a list of all applications belonging to the authorized user.

```

curl -X GET \
  http://127.0.0.1:8000/api/v1.0/calc/applications/ \
  -H 'Accept: */*' \
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwOTQ5OTksImVtYWlsIjoiIn0.wLUVsUtZMRwLj6bvRK5dWAFmUTwYP
NMbGpuHmM09mXY' \
  -H 'Content-Type: application/json'

```

which yields

```

{
  "count": 2,
  "next": null,
  "previous": null,
  "results": [
    {

```

```

    "app_name": "app1",
    "date_created": "2019-11-14T16:18:18.730938Z",
    "pk_hash":
"373d10dfd18ae01daca0dcad1448d6b62eca0b7ad66b3c439ba869fa",
    "services": [
      {
        "attributes": [],
        "service_type": "aze-basic_a0",
        "version_date": "2019-11-03T20:22:09.047029Z",
        "pk_hash":
"6baf5d6af555b4c4046d6f0a00b0ea6d50ed8629fa97515e70f3fce8",
        "location": "brazilsouth",
        "price_type": "onDemandPrice",
        "provider": "aze"
      }
    ],
    "description": ""
  },
  {
    "app_name": "app2",
    "date_created": "2019-11-20T17:15:42.879412Z",
    "pk_hash":
"be78422202906efd425a84bdbb796ef38d91cd068ddb9773cb1aea1d",
    "services": [
      {
        "attributes": [],
        "service_type": "aze-basic_a0",
        "version_date": "2019-11-03T20:22:10.225264Z",
        "pk_hash":
"6baf5d6af555b4c4046d6f0a00b0ea6d50ed8629fa97515e70f3fce8",
        "location": "us-east-2",
        "price_type": "reserved1yearPrice",
        "provider": "aze"
      }
    ],
    "description": ""
  }
]
}

```

The list results are paginated, and by default shows at most the first 10 applications. The response also contains URLs corresponding to the next and previous 'pages' if applicable, and a count of the total number of applications in the database.

The limit and offset of the results can be customized like so:

```

curl -X GET \
  'http://127.0.0.1:8000/api/v1.0/calc/applications/?limit=1&offset=0' \
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh

```

```
i5D3CpCiKQoFk' \
-H 'Content-Type: application/json'
```

which would yield:

```
{
  "count": 2,
  "next": "http://127.0.0.1:8000/api/v1.0/calc/applications/?
limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "app_name": "app1",
      "date_created": "2019-11-14T16:18:18.730938Z",
      "pk_hash":
"373d10dfd18ae01daca0dcad1448d6b62eca0b7ad66b3c439ba869fa",
      "services": [
        {
          "attributes": [],
          "service_type": "aze-basic_a0",
          "version_date": "2019-11-03T20:22:09.047029Z",
          "pk_hash":
"6baf5d6af555b4c4046d6f0a00b0ea6d50ed8629fa97515e70f3fce8",
          "location": "brazilsouth",
          "price_type": "onDemandPrice",
          "provider": "aze"
        }
      ],
      "description": ""
    }
  ]
}
```

2. Get the details of a specific application. This can be accomplished by passing the primary key `pk` or the `app_name` in the URL, like so

```
# Using the pk of the application
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/applications/pk/89a8e4463ed6ab13713d7ad
8c82b40e23d6dec023348d49625e74885/ \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlbnJkwOTQ5OTksImVtYWlsIjoiIn0.wLUVsUtZMRwLj6bvRK5dWAFmUTwYP
NMbGpuHmmO9mXY' \
-H 'Content-Type: application/json'
```

```
# Using the app_name of the application; this also means that two
applications belonging to the same user cannot have identical app_names.
curl -X GET \
http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/Enterprise_HR_SAA
S_Application/ \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwOTQ5OTksImVtYWlsIjoiIn0.wLUVsUtZMRwLj6bvRK5dWAFmUTwYP
NMbGpuHmmO9mXY' \
-H 'Content-Type: application/json' \
```

1.2.3. Obtaining a report on an Application

A report contains cost data for the application. As before, the application can be specified using a `pk` or the `app_name`.

```
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/reports/applications/app_name/Enterpris
e_HR_SAAS_Application/ \
-H 'Accept: */*' \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwOTQ5OTksImVtYWlsIjoiIn0.wLUVsUtZMRwLj6bvRK5dWAFmUTwYP
NMbGpuHmmO9mXY' \
-H 'Content-Type: application/json'

#
# OR
#
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/reports/applications/pk/89a8e4463ed6ab1
3713d7ad8c82b40e23d6dec023348d49625e74885/ \
-H 'Accept: */*' \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwOTQ5OTksImVtYWlsIjoiIn0.wLUVsUtZMRwLj6bvRK5dWAFmUTwYP
NMbGpuHmmO9mXY' \
-H 'Content-Type: application/json'
```

Here is an example of a report returned by the above API calls

```
{
  "application": {
    "app_name": "Enterprise_HR_SAAS_Application",
    "date_created": "2019-09-20T01:51:24.368595Z",
```

```
"pk_hash": "89a8e4463ed6ab13713d7ad8c82b40e23d6dec023348d49625e74885",
"services": [
  {
    "attributes": [
      {
        "attr_id": "unit",
        "value": "instance"
      },
      {
        "attr_id": "util",
        "value": "40"
      },
      {
        "attr_id": "max_util",
        "value": "80"
      },
      {
        "attr_id": "quantity",
        "value": "8"
      }
    ],
    "service_type": "aws-c5.xlarge",
    "date_created": "2019-09-20T01:51:24.416991Z",
    "pk_hash":
"881a3b377b41ff3c8d6084d79f8e1bfa42d9fc45f9642ef9eead88b8",
    "location": "us-west-2",
    "price_type": "onDemandPrice",
    "provider": "aws"
  },
  {
    "attributes": [
      {
        "attr_id": "unit",
        "value": "instance"
      },
      {
        "attr_id": "util",
        "value": "30"
      },
      {
        "attr_id": "max_util",
        "value": "80"
      },
      {
        "attr_id": "quantity",
        "value": "3"
      }
    ],
    "service_type": "aws-c5.2xlarge",
    "date_created": "2019-09-20T01:51:24.485717Z",
    "pk_hash":
"45ddf322bad20dae58a7cf736493e02a7d178448154daec065fa0398",
    "location": "us-east-2",
    "price_type": "onDemandPrice",
```

```

    "provider": "aws"
  }
],
"description": "Finance department application",
"metadata": "this can be any string up to 1024 characters long, for
example, fookey=barvalue"
},
"cost_components": [
  {
    "currentGen": "True",
    "numcpus": "4",
    "memory": "7.45",
    "quantity": "8",
    "util": "40",
    "max_util": "80",
    "service": "aws-c5.xlarge",
    "unit_cost": 0.17,
    "unit": "",
    "location": "us-west-2",
    "service_cost": 1.36,
    "price_type": "onDemandPrice"
  },
  {
    "currentGen": "True",
    "memory": "15.1",
    "numcpus": "8",
    "quantity": "3",
    "util": "30",
    "max_util": "80",
    "service": "aws-c5.2xlarge",
    "unit_cost": 0.34,
    "unit": "",
    "location": "us-east-2",
    "service_cost": 1.02,
    "price_type": "onDemandPrice"
  }
],
"total_cost": 2.38
}

```

1.2.4. Querying Applications

The application name and its description may have keywords that the user wants to query. This can be accomplished using an API call like so

```

curl -X GET \
  http://127.0.0.1:8000/api/v1.0/calc/applicationsearch/aqw Cassandra\
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh

```

```
i5D3CpCiKQoFk' \
-H 'Content-Type: application/json' \
```

1.2.4.1 Querying Applications using ApplicationList

The application may have specific attributes based on which user want to query. Following Filters can be applied to API Call.

Filters

Filter	Description
name_contains	Filter if application name contains the entered string
metadata_contains	Filter if application metadata contains entered string
min_time	Filter if application creation date is more recent than min_time
max_time	Filter if application creation date is earlier than max_time

Note: Times have to be specified in following format:

```
YYYY-MM-DD HH:MM[:ss[.uuuuuu]] [TZ]
# Some Valid DateTime Examples:
2019-12-15
2020-01-01
2019-12-12 9:55:57
)
```

We can now filter applications, like so

```
curl --location --request GET '127.0.0.1:8000/api/v1.0/calc/applications/?
name_contains=CI%20database&metadata_contains=Testing' \
--header 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJ1c2VybmFtZSI6Im1hY2
ludG9zaF93aG9sZXNhbGUiLCJleHAiOjE1ODE3MTQzNzUsImVtYWlsIjoieYWRtaW5AbWFjaGlud
G9zaC5jb20ifQ.a5bimnHANfv-rSU5xrt11zX7VqLTy0hefOqyYyaPjls' \
--header 'Content-Type: application/json'
```

It returns following Output:

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
```



```
    "app_name": "CI database",
    "date_created": "2019-12-11T03:27:27.842342Z",
    "pk_hash":
"14ee6590b4e342b91be4294e7dd64b39525ed8f0ae6fbbbe23ddaad5b",
    "services": [
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "15"
          },
          {
            "attr_id": "max_util",
            "value": "80"
          }
        ],
        "service_type": "aws-t3.2xlarge",
        "version_date": "2019-11-03T20:24:34.333836Z",
        "pk_hash":
"4bd0129bf078e5d6d1ad0966f4c2481194041e175cdcebc49673a988",
        "location": "ca-central-1",
        "price_type": "onDemandPrice",
        "provider": "aws",
        "quantity": 10,
        "description": ""
      },
      {
        "attributes": [],
        "service_type": "Custom_Organization_IT_HR_cost",
        "version_date": "2019-12-11T03:27:26Z",
        "pk_hash":
"4d87230cddb963c11a1c1a7a0989d5c89c6d5eb113463f811ce18e7c",
        "location": "anywhere",
        "price_type": "hr_overhead_cost",
        "provider": "org",
        "quantity": 8,
        "description": ""
      },
      {
        "attributes": [],
        "service_type":
"Custom_Organization_amortized_IT_cost",
        "version_date": "2019-12-11T03:27:26Z",
        "pk_hash":
"09c6fe490b524143121e5b22130a91c2ca01f7dab43c749ad508dd1b",
        "location": "anywhere",
        "price_type": "hr_overhead_cost",
        "provider": "org",
        "quantity": 8,
        "description": ""
      }
    ]
  }
}
```

```

        }
      ],
      "description": "Database for all IT CI numbers in the
organization",
      "metadata": "Testing It"
    }
  ]
}

```

Sort Criteria

The returned application list can (optionally) be pre-sorted based on the following values for the `sort` parameter in the url.

sort Parameter value	Meaning
date	Sort according to date created (ascending order)
-date	Sort according to date created (descending order)
name	Sort according to application name (ascending order)
-name	Sort according to application name (descending order)
service_instance	Sort according to date created (ascending order)
-service_instance	Sort according to date created (descending order)

For example:

```

curl --location --request GET '127.0.0.1:8000/api/v1.0/calc/applications/?
metadata_contains=Test&sort=-name' \
--header 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJlc2VybmFtZSI6Im1hY2
ludG9zaF93aG9sZXNhbnGUiLCJleHAiOjE1ODIxMzM5MjcsImVtYWlsIjojoiYWRtaW5AbWFjaGlud
G9zaC5jb20ifQ.RM01U6ELAHN8ha3raVRrbCZHGlOC3dLarkCYwRX4mII' \
--header 'Content-Type: application/json'

```

1.2.5. Finding Application Alternatives on Another Provider

We now use the BigBitBus API matching engine to find comparable services in other cloud providers in case the user wants to migrate to another cloud provider.

In our example we wish to look at moving the application into the Azure provider (aze) from the current AWS provider.

```

curl -X GET \
'http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/Enterprise_HR_SA

```

```
AS_Application/provider/aze/pricetype/onDemandPrice/?matchtype=attribute' \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json'
```

This yields

```
{
  "application": {
    "app_name": "Enterprise_HR_SAAS_Application",
    "date_created": "2020-01-27T16:53:48.552308Z",
    "pk_hash":
    "e7fcbe981109cac618a0e0c3bf96296c3e5b8daa80f8a60894ffaf32",
    "services": [
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "40"
          },
          {
            "attr_id": "max_util",
            "value": "80"
          },
          {
            "attr_id": "quantity",
            "value": "8"
          }
        ],
        "service_type": "aws-c5.xlarge",
        "version_date": "2019-11-03T20:23:48.916702Z",
        "pk_hash":
        "dfa94ee6827261a6b5b4d9e89e3e347df0441da02c44e0f585b9c11a",
        "location": "us-east-2",
        "price_type": "onDemandPrice",
        "provider": "aws",
        "quantity": 8,
        "description": ""
      },
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          }
        ]
      }
    ]
  }
}
```

```

        "attr_id": "util",
        "value": "30"
    },
    {
        "attr_id": "max_util",
        "value": "80"
    },
    {
        "attr_id": "quantity",
        "value": "3"
    }
],
"service_type": "aws-c5.2xlarge",
"version_date": "2019-11-03T20:23:49.534033Z",
"pk_hash":
"e825125f695dbd83293f054ffd802acb042efa067ce2976bf5a0db66",
"location": "us-east-2",
"price_type": "onDemandPrice",
"provider": "aws",
"quantity": 3,
"description": ""
}
],
"description": "",
"metadata": ""
},
"current_application": {
    "components": [
        {
            "numcpus": "4",
            "memory": "8",
            "currentGen": "True",
            "util": "40",
            "max_util": "80",
            "service_type": "aws-c5.xlarge",
            "unit_cost": 0.17,
            "quantity": 8,
            "service_cost": 1.36,
            "discount": 0,
            "unit": "",
            "location": "us-east-2",
            "price_type": "onDemandPrice",
            "price_version_date": "2019-11-03T20:23:48.920561Z"
        },
        {
            "currentGen": "True",
            "numcpus": "8",
            "memory": "16",
            "util": "30",
            "max_util": "80",
            "service_type": "aws-c5.2xlarge",
            "unit_cost": 0.34,
            "quantity": 3,
            "service_cost": 1.02,

```

```
        "discount": 0,
        "unit": "",
        "location": "us-east-2",
        "price_type": "onDemandPrice",
        "price_version_date": "2019-11-03T20:23:49.537700Z"
    }
],
    "total_cost": 2.38
},
"alternative_option": {
    "components": [
        {
            "costph": 0.169,
            "price_type": "onDemandPrice",
            "discount": 30,
            "quantity": 8,
            "service_type": "aze-standard_f4s_v2",
            "location": "eastus2",
            "service_cost": 1.35,
            "numcpus": "4",
            "memory": "8",
            "instanceTypeCategory": "Compute optimized",
            "currentGen": "False",
            "gpus": "0",
            "ntwPerf": "1 Gbit/s",
            "predbogo": "N/A",
            "predcpu": "N/A",
            "tgzfilename": "N/A",
            "epoch-secs": "N/A",
            "minbogo": "N/A",
            "maxbogo": "N/A"
        },
        {
            "costph": 0.338,
            "price_type": "onDemandPrice",
            "discount": 30,
            "quantity": 3,
            "service_type": "aze-standard_f8s_v2",
            "location": "eastus2",
            "service_cost": 1.01,
            "ntwPerf": "1 Gbit/s",
            "instanceTypeCategory": "Compute optimized",
            "currentGen": "False",
            "gpus": "0",
            "numcpus": "8",
            "memory": "16",
            "predbogo": "N/A",
            "predcpu": "N/A",
            "tgzfilename": "N/A",
            "epoch-secs": "N/A",
            "minbogo": "N/A",
            "maxbogo": "N/A"
        }
    ]
},
```

```
    "total_cost": 2.36
  }
}
```

The matching engine has returned similar services in Azure which may be used to compose the same application there, should the user decide to migrate from AWS to Azure. Compare the `total_cost` - \$2.36 per hour in Azure with the cost report of Step 4 - \$2.38 in AWS. The output may also include some a "note" about other possible matching options and their hourly costs.

The matching engine compares service type attributes (e.g. numcpus and memory in the case of VMs) in order to make comparisons and present similar services from other providers.

Alternatively, using version 2 of the API, the server will return multiple alternatives for each service type in the application

Using the same example:

```
curl -X GET \
'http://127.0.0.1:8000/api/v2.0/calc/applications/app_name/Enterprise_HR_SAAS_Application/provider/aze/pricetype/onDemandPrice/?matchtype=attribute' \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6ImdlbmVyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGXCpGBRNendUy2Kw' \
-H 'Content-Type: application/json'
```

This yields

```
{
  "application": {
    "app_name": "Enterprise_HR_SAAS_Application",
    "date_created": "2020-01-27T16:53:48.552308Z",
    "pk_hash":
    "e7fcbe981109cac618a0e0c3bf96296c3e5b8daa80f8a60894ffaf32",
    "services": [
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "40"
          },
          {
            "attr_id": "max_util",
            "value": "80"
          }
        ]
      }
    ]
  }
}
```

```

    },
    {
      "attr_id": "quantity",
      "value": "8"
    }
  ],
  "service_type": "aws-c5.xlarge",
  "version_date": "2019-11-03T20:23:48.916702Z",
  "pk_hash":
"dfa94ee6827261a6b5b4d9e89e3e347df0441da02c44e0f585b9c11a",
  "location": "us-east-2",
  "price_type": "onDemandPrice",
  "provider": "aws",
  "quantity": 8,
  "description": ""
},
{
  "attributes": [
    {
      "attr_id": "unit",
      "value": "instance"
    },
    {
      "attr_id": "util",
      "value": "30"
    },
    {
      "attr_id": "max_util",
      "value": "80"
    },
    {
      "attr_id": "quantity",
      "value": "3"
    }
  ],
  "service_type": "aws-c5.2xlarge",
  "version_date": "2019-11-03T20:23:49.534033Z",
  "pk_hash":
"e825125f695dbd83293f054ffd802acb042efa067ce2976bf5a0db66",
  "location": "us-east-2",
  "price_type": "onDemandPrice",
  "provider": "aws",
  "quantity": 3,
  "description": ""
}
],
"description": "",
"metadata": ""
},
"current_application": {
  "components": [
    {
      "numcpus": "4",
      "memory": "8",

```

```
    "currentGen": "True",
    "util": "40",
    "max_util": "80",
    "service_type": "aws-c5.xlarge",
    "unit_cost": 0.17,
    "quantity": 8,
    "service_cost": 1.36,
    "discount": 0,
    "unit": "",
    "location": "us-east-2",
    "price_type": "onDemandPrice",
    "price_version_date": "2019-11-03T20:23:48.920561Z"
  },
  {
    "currentGen": "True",
    "numcpus": "8",
    "memory": "16",
    "util": "30",
    "max_util": "80",
    "service_type": "aws-c5.2xlarge",
    "unit_cost": 0.34,
    "quantity": 3,
    "service_cost": 1.02,
    "discount": 0,
    "unit": "",
    "location": "us-east-2",
    "price_type": "onDemandPrice",
    "price_version_date": "2019-11-03T20:23:49.537700Z"
  }
],
"total_cost": 2.38
},
"alternative_option": {
  "components": [
    [
      {
        "costph": 0.169,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 8,
        "service_type": "aze-standard_f4s_v2",
        "location": "eastus2",
        "service_cost": 1.35,
        "numcpus": "4",
        "memory": "8",
        "instanceTypeCategory": "Compute optimized",
        "currentGen": "False",
        "gpus": "0",
        "ntwPerf": "1 Gbit/s",
        "predbogo": "N/A",
        "predcpu": "N/A",
        "tgzfilename": "N/A",
        "epoch-secs": "N/A",
        "minbogo": "N/A",
```



```
        "maxbogo": "N/A"
    },
    {
        "costph": 0.176,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 8,
        "service_type": "aze-basic_a3",
        "location": "eastus2",
        "service_cost": 1.41,
        "numcpus": "4",
        "memory": "7",
        "instanceTypeCategory": "General purpose",
        "currentGen": "False",
        "gpus": "0",
        "ntwPerf": "1 Gbit/s",
        "predbogo": "N/A",
        "predcpu": "N/A",
        "tgzfilename": "N/A",
        "epoch-secs": "N/A",
        "minbogo": "N/A",
        "maxbogo": "N/A"
    },
    {
        "costph": 0.191,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 8,
        "service_type": "aze-standard_a4_v2",
        "location": "eastus2",
        "service_cost": 1.53,
        "numcpus": "4",
        "memory": "8",
        "instanceTypeCategory": "General purpose",
        "currentGen": "False",
        "gpus": "0",
        "ntwPerf": "1 Gbit/s",
        "predbogo": "2519",
        "predcpu": "2520",
        "tgzfilename": "/tmp/generic-cpu-data-aze-
Standard_A4_v2",
        "epoch-secs": "1548440756",
        "minbogo": "239.51127819548876",
        "maxbogo": "23951.127819548874"
    },
    {
        "costph": 0.199,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 8,
        "service_type": "aze-standard_f4",
        "location": "eastus2",
        "service_cost": 1.59,
        "numcpus": "4",
```

```
    "memory": "8",
    "instanceTypeCategory": "Compute optimized",
    "currentGen": "False",
    "gpus": "0",
    "ntwPerf": "1 Gbit/s",
    "predbogo": "547",
    "predcpu": "548",
    "tgzfilename": "/tmp/generic-cpu-data-aze-Standard_F4",
    "epoch-secs": "1548434862",
    "minbogo": "535.7888721804512",
    "maxbogo": "53578.88721804512"
  },
  {
    "costph": 0.24,
    "price_type": "onDemandPrice",
    "discount": 30,
    "quantity": 8,
    "service_type": "aze-standard_a3",
    "location": "eastus2",
    "service_cost": 1.92,
    "numcpus": "4",
    "memory": "7",
    "instanceTypeCategory": "General purpose",
    "currentGen": "False",
    "gpus": "0",
    "ntwPerf": "1 Gbit/s",
    "predbogo": "N/A",
    "predcpu": "N/A",
    "tgzfilename": "N/A",
    "epoch-secs": "N/A",
    "minbogo": "N/A",
    "maxbogo": "N/A"
  }
],
[
  {
    "costph": 0.338,
    "price_type": "onDemandPrice",
    "discount": 30,
    "quantity": 3,
    "service_type": "aze-standard_f8s_v2",
    "location": "eastus2",
    "service_cost": 1.01,
    "ntwPerf": "1 Gbit/s",
    "instanceTypeCategory": "Compute optimized",
    "currentGen": "False",
    "gpus": "0",
    "numcpus": "8",
    "memory": "16",
    "predbogo": "N/A",
    "predcpu": "N/A",
    "tgzfilename": "N/A",
    "epoch-secs": "N/A",
    "minbogo": "N/A",
```

```
        "maxbogo": "N/A"
    },
    {
        "costph": 0.352,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 3,
        "service_type": "aze-basic_a4",
        "location": "eastus2",
        "service_cost": 1.06,
        "ntwPerf": "1 Gbit/s",
        "instanceTypeCategory": "General purpose",
        "currentGen": "False",
        "gpu": "0",
        "numcpu": "8",
        "memory": "14",
        "predbogo": "N/A",
        "predcpu": "N/A",
        "tgzfilename": "N/A",
        "epoch-secs": "N/A",
        "minbogo": "N/A",
        "maxbogo": "N/A"
    },
    {
        "costph": 0.398,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 3,
        "service_type": "aze-standard_f8s",
        "location": "eastus2",
        "service_cost": 1.19,
        "ntwPerf": "1 Gbit/s",
        "instanceTypeCategory": "Compute optimized",
        "currentGen": "False",
        "gpu": "0",
        "numcpu": "8",
        "memory": "16",
        "predbogo": "261",
        "predcpu": "262",
        "tgzfilename": "/tmp/generic-cpu-data-aze-
Standard_F8s",
        "epoch-secs": "1548440715",
        "minbogo": "1067.5455639097747",
        "maxbogo": "106754.55639097748"
    },
    {
        "costph": 0.398,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 3,
        "service_type": "aze-standard_f8",
        "location": "eastus2",
        "service_cost": 1.19,
        "ntwPerf": "1 Gbit/s",
```

```

        "instanceTypeCategory": "Compute optimized",
        "currentGen": "False",
        "gpus": "0",
        "numcpus": "8",
        "memory": "16",
        "predbogo": "917",
        "predcpu": "918",
        "tgzfilename": "/tmp/generic-cpu-data-aze-Standard_F8",
        "epoch-secs": "1548434858",
        "minbogo": "1065.4139849624062",
        "maxbogo": "106541.39849624061"
    },
    {
        "costph": 0.4,
        "price_type": "onDemandPrice",
        "discount": 30,
        "quantity": 3,
        "service_type": "aze-standard_a8_v2",
        "location": "eastus2",
        "service_cost": 1.2,
        "ntwPerf": "1 Gbit/s",
        "instanceTypeCategory": "General purpose",
        "currentGen": "False",
        "gpus": "0",
        "numcpus": "8",
        "memory": "16",
        "predbogo": "1521",
        "predcpu": "1522",
        "tgzfilename": "/tmp/generic-cpu-data-aze-
Standard_A8_v2",
        "epoch-secs": "1548440767",
        "minbogo": "438.69669172932333",
        "maxbogo": "43869.669172932336"
    }
]
],
"total_cost": 2.36
}
}

```

In this case, the `total_cost` is calculated as the alternative with the lowest cost for each component.

1.2.6. VM Optimization

If the average CPU utilization of a VM is specified in the application (`util`) then there are a few use cases where the VMs being used by the application can be optimized.

1. Optimize the cost of running the service within the same provider by choosing smaller VMs in case CPU utilization is low:

```
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/Enterprise_HR_SAA
S_Application/provider/aws/pricetype/onDemandPrice/ \
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
  -H 'Content-Type: application/json'
```

This returns the following JSON string

```
{
  "application": {
    "app_name": "Enterprise_HR_SAAS_Application",
    "date_created": "2020-01-27T16:53:48.552308Z",
    "pk_hash":
    "e7fcbe981109cac618a0e0c3bf96296c3e5b8daa80f8a60894ffaf32",
    "services": [
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "40"
          },
          {
            "attr_id": "max_util",
            "value": "80"
          },
          {
            "attr_id": "quantity",
            "value": "8"
          }
        ],
        "service_type": "aws-c5.xlarge",
        "version_date": "2019-11-03T20:23:48.916702Z",
        "pk_hash":
        "dfa94ee6827261a6b5b4d9e89e3e347df0441da02c44e0f585b9c11a",
        "location": "us-east-2",
        "price_type": "onDemandPrice",
        "provider": "aws",
        "quantity": 8,
        "description": ""
      },
      {
        "attributes": [
          {
```

```

        "attr_id": "unit",
        "value": "instance"
    },
    {
        "attr_id": "util",
        "value": "30"
    },
    {
        "attr_id": "max_util",
        "value": "80"
    },
    {
        "attr_id": "quantity",
        "value": "3"
    }
],
"service_type": "aws-c5.2xlarge",
"version_date": "2019-11-03T20:23:49.534033Z",
"pk_hash":
"e825125f695dbd83293f054ffd802acb042efa067ce2976bf5a0db66",
"location": "us-east-2",
"price_type": "onDemandPrice",
"provider": "aws",
"quantity": 3,
"description": ""
}
],
"description": "",
"metadata": ""
},
"current_application": {
    "components": [
        {
            "numcpus": "4",
            "memory": "8",
            "currentGen": "True",
            "util": "40",
            "max_util": "80",
            "service_type": "aws-c5.xlarge",
            "unit_cost": 0.17,
            "quantity": 8,
            "service_cost": 1.36,
            "discount": 0,
            "unit": "",
            "location": "us-east-2",
            "price_type": "onDemandPrice",
            "price_version_date": "2019-11-03T20:23:48.920561Z"
        },
        {
            "currentGen": "True",
            "numcpus": "8",
            "memory": "16",
            "util": "30",
            "max_util": "80",

```

```

        "service_type": "aws-c5.2xlarge",
        "unit_cost": 0.34,
        "quantity": 3,
        "service_cost": 1.02,
        "discount": 0,
        "unit": "",
        "location": "us-east-2",
        "price_type": "onDemandPrice",
        "price_version_date": "2019-11-03T20:23:49.537700Z"
    },
    ],
    "total_cost": 2.38
},
"alternative_option": {
    "components": [
        {
            "service_type": "aws-t3.xlarge",
            "machine_cpu": 41.56805002183218,
            "numcpus": "4",
            "memory": "16",
            "costph": 0.166,
            "service_cost": 1.328,
            "price_version_date": "2019-11-04T15:42:57.155528Z",
            "description": "",
            "price_type": "onDemandPrice",
            "location": "us-east-2"
        },
        {
            "service_type": "aws-t3.xlarge",
            "machine_cpu": 65.54483646460177,
            "numcpus": "4",
            "memory": "16",
            "costph": 0.166,
            "service_cost": 0.498,
            "price_version_date": "2019-11-04T15:42:57.155528Z",
            "description": "",
            "price_type": "onDemandPrice",
            "location": "us-east-2"
        }
    ],
    "total_cost": 1.83
}
}

```

Alternatively, using version 2 of the API, the server returns multiple alternatives for each component. Using the same example:

```

curl -X GET \
http://127.0.0.1:8000/api/v2.0/calc/applications/app_name/Enterprise_HR_SAA
S_Application/provider/aws/pricetype/onDemandPrice/ \

```

```
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json'
```

```
{
  "application": {
    "app_name": "Enterprise_HR_SAAS_Application",
    "date_created": "2020-01-27T16:53:48.552308Z",
    "pk_hash":
    "e7fcbe981109cac618a0e0c3bf96296c3e5b8daa80f8a60894ffaf32",
    "services": [
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "40"
          },
          {
            "attr_id": "max_util",
            "value": "80"
          },
          {
            "attr_id": "quantity",
            "value": "8"
          }
        ],
        "service_type": "aws-c5.xlarge",
        "version_date": "2019-11-03T20:23:48.916702Z",
        "pk_hash":
        "dfa94ee6827261a6b5b4d9e89e3e347df0441da02c44e0f585b9c11a",
        "location": "us-east-2",
        "price_type": "onDemandPrice",
        "provider": "aws",
        "quantity": 8,
        "description": ""
      },
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "30"
          }
        ]
      }
    ]
  }
}
```



```

        },
        {
            "attr_id": "max_util",
            "value": "80"
        },
        {
            "attr_id": "quantity",
            "value": "3"
        }
    ],
    "service_type": "aws-c5.2xlarge",
    "version_date": "2019-11-03T20:23:49.534033Z",
    "pk_hash":
    "e825125f695dbd83293f054ffd802acb042efa067ce2976bf5a0db66",
    "location": "us-east-2",
    "price_type": "onDemandPrice",
    "provider": "aws",
    "quantity": 3,
    "description": ""
    }
],
"description": "",
"metadata": ""
},
"current_application": {
    "components": [
        {
            "numcpus": "4",
            "memory": "8",
            "currentGen": "True",
            "util": "40",
            "max_util": "80",
            "service_type": "aws-c5.xlarge",
            "unit_cost": 0.17,
            "quantity": 8,
            "service_cost": 1.36,
            "discount": 0,
            "unit": "",
            "location": "us-east-2",
            "price_type": "onDemandPrice",
            "price_version_date": "2019-11-03T20:23:48.920561Z"
        },
        {
            "currentGen": "True",
            "numcpus": "8",
            "memory": "16",
            "util": "30",
            "max_util": "80",
            "service_type": "aws-c5.2xlarge",
            "unit_cost": 0.34,
            "quantity": 3,
            "service_cost": 1.02,
            "discount": 0,
            "unit": "",

```

```
        "location": "us-east-2",
        "price_type": "onDemandPrice",
        "price_version_date": "2019-11-03T20:23:49.537700Z"
    }
],
"total_cost": 2.38
},
"alternative_option": {
    "components": [
        [
            {
                "service_type": "aws-t3.xlarge",
                "machine_cpu": 41.56805002183218,
                "numcpus": "4",
                "memory": "16",
                "costph": 0.166,
                "service_cost": 1.328,
                "price_version_date": "2019-11-04T15:42:57.155528Z",
                "description": "",
                "price_type": "onDemandPrice",
                "location": "us-east-2"
            },
            {
                "service_type": "aws-c5.xlarge",
                "machine_cpu": 39.7316068337841,
                "numcpus": "4",
                "memory": "8",
                "costph": 0.17,
                "service_cost": 1.36,
                "price_version_date": "2019-11-03T20:23:48.920561Z",
                "description": "",
                "price_type": "onDemandPrice",
                "location": "us-east-2"
            },
            {
                "service_type": "aws-t2.xlarge",
                "machine_cpu": 77.15851994825053,
                "numcpus": "4",
                "memory": "16",
                "costph": 0.186,
                "service_cost": 1.488,
                "price_version_date": "2019-11-04T15:42:55.527389Z",
                "description": "",
                "price_type": "onDemandPrice",
                "location": "us-east-2"
            },
            {
                "service_type": "aws-m5.xlarge",
                "machine_cpu": 42.00038383036275,
                "numcpus": "4",
                "memory": "16",
                "costph": 0.192,
                "service_cost": 1.536,
                "price_version_date": "2019-11-03T20:23:52.246131Z",
```

```
    "description": "",
    "price_type": "onDemandPrice",
    "location": "us-east-2"
  },
  {
    "service_type": "aws-m4.xlarge",
    "machine_cpu": 55.35800119854204,
    "numcpus": "4",
    "memory": "16",
    "costph": 0.2,
    "service_cost": 1.6,
    "price_version_date": "2019-11-03T20:23:51.047623Z",
    "description": "",
    "price_type": "onDemandPrice",
    "location": "us-east-2"
  }
],
[
  {
    "service_type": "aws-t3.xlarge",
    "machine_cpu": 65.54483646460177,
    "numcpus": "4",
    "memory": "16",
    "costph": 0.166,
    "service_cost": 0.498,
    "price_version_date": "2019-11-04T15:42:57.155528Z",
    "description": "",
    "price_type": "onDemandPrice",
    "location": "us-east-2"
  },
  {
    "service_type": "aws-c5.xlarge",
    "machine_cpu": 62.64911803725377,
    "numcpus": "4",
    "memory": "8",
    "costph": 0.17,
    "service_cost": 0.51,
    "price_version_date": "2019-11-03T20:23:48.920561Z",
    "description": "",
    "price_type": "onDemandPrice",
    "location": "us-east-2"
  },
  {
    "service_type": "aws-m5.xlarge",
    "machine_cpu": 66.22654390008098,
    "numcpus": "4",
    "memory": "16",
    "costph": 0.192,
    "service_cost": 0.5760000000000001,
    "price_version_date": "2019-11-03T20:23:52.246131Z",
    "description": "",
    "price_type": "onDemandPrice",
    "location": "us-east-2"
  },
],
```

```

        {
            "service_type": "aws-c5n.xlarge",
            "machine_cpu": 61.508921656147756,
            "numcpus": "4",
            "memory": "10.5",
            "costph": 0.216,
            "service_cost": 0.648,
            "price_version_date": "2019-11-03T20:23:50.335644z",
            "description": "",
            "price_type": "onDemandPrice",
            "location": "us-east-2"
        },
        {
            "service_type": "aws-c5.2xlarge",
            "machine_cpu": 29.74818892191024,
            "numcpus": "8",
            "memory": "16",
            "costph": 0.34,
            "service_cost": 1.02,
            "price_version_date": "2019-11-03T20:23:49.537700z",
            "description": "",
            "price_type": "onDemandPrice",
            "location": "us-east-2"
        }
    ],
    "total_cost": 1.83
}

```

The BigBitBus API server has performance information about 100s of VMs across different cloud providers. It finds other VMs which may be cheaper than the original service types. Note that the CPU utilization of the suggested VMs reported by the tool is an estimate based on CPU performance and utilization models. The `util` and `max_util` parameters set in the application are used as starting points for the calculation. The `util` parameter is the average utilization of the VM's CPU whereas the `max_util` is the maximum utilization of the suggested CPU that the user is willing to allow. For example if `max_util` is 80 then the tool will only consider VMs whose performance model indicates that less than 80% CPU will be utilized.

It is important to note that the utilization is an estimate; users are encouraged to verify the utilization for their specific application on the suggested options before deploying the application for production use.

2. A user may also want to optimize VMs when moving an application from one cloud to another. For example, consider the case when a user is considering moving from AWS to Azure and she wants to optimize costs by choosing smaller VMs, if low CPU utilization allows:

```

curl -X GET \
http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/Enterprise_HR_SAA
S_Application/provider/aze/pricetype/onDemandPrice/ \
-H 'Authorization: JWT

```

```

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json'

```

This returns the following alternate options for the application.

```

{
  "application": {
    "app_name": "Enterprise_HR_SAAS_Application",
    "date_created": "2019-09-20T01:51:24.368595Z",
    "pk_hash": "89a8e4463ed6ab13713d7ad8c82b40e23d6dec023348d49625e74885",
    "services": [
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "40"
          },
          {
            "attr_id": "max_util",
            "value": "80"
          },
          {
            "attr_id": "quantity",
            "value": "8"
          }
        ],
        "service_type": "aws-c5.xlarge",
        "date_created": "2019-09-20T01:51:24.416991Z",
        "pk_hash":
"881a3b377b41ff3c8d6084d79f8e1bfa42d9fc45f9642ef9eead88b8",
        "location": "us-west-2",
        "price_type": "onDemandPrice",
        "provider": "aws"
      },
      {
        "attributes": [
          {
            "attr_id": "unit",
            "value": "instance"
          },
          {
            "attr_id": "util",
            "value": "30"
          },
          {
            "attr_id": "max_util",

```

```
        "value": "80"
      },
      {
        "attr_id": "quantity",
        "value": "3"
      }
    ],
    "service_type": "aws-c5.2xlarge",
    "date_created": "2019-09-20T01:51:24.485717Z",
    "pk_hash":
"45ddf322bad20dae58a7cf736493e02a7d178448154daec065fa0398",
    "location": "us-east-2",
    "price_type": "onDemandPrice",
    "provider": "aws"
  }
],
"description": "Finance department application",
"metadata": "this can be any string up to 1024 characters long, for
example, fookey=barvalue",
},
"current_application": {
  "components": [
    {
      "currentGen": "True",
      "numcpus": "4",
      "memory": "7.45",
      "quantity": "8",
      "util": "40",
      "max_util": "80",
      "service_type": "aws-c5.xlarge",
      "unit_cost": 0.17,
      "unit": "",
      "location": "us-west-2",
      "service_cost": 1.36,
      "price_type": "onDemandPrice"
    },
    {
      "currentGen": "True",
      "memory": "15.1",
      "numcpus": "8",
      "quantity": "3",
      "util": "30",
      "max_util": "80",
      "service_type": "aws-c5.2xlarge",
      "unit_cost": 0.34,
      "unit": "",
      "location": "us-east-2",
      "service_cost": 1.02,
      "price_type": "onDemandPrice"
    }
  ],
  "total_cost": 2.38
},
"alternative_option": {
```

```

"components": [
  {
    "servicetype": "aze-standard_b4ms",
    "machine_cpu": 75.43015027854008,
    "numcpus": "4",
    "memory": "15.67",
    "costph": 0.166,
    "description": "aze-standard_b4ms",
    "price_type": "onDemandPrice",
    "location": "us-west-2",
    "service_cost": 1.328
  },
  {
    "servicetype": "aze-standard_f4",
    "machine_cpu": 69.46153379604617,
    "numcpus": "4",
    "memory": "7.79",
    "costph": 0.199,
    "description": "aze-standard_f4",
    "price_type": "onDemandPrice",
    "location": "us-east-2",
    "service_cost": 0.597
  }
],
"total_cost": 1.93
}
}

```

3. A user may want to also optimize by selecting another price_type. For example, in addition to moving to a smaller VM within AWS, a user may want to move from an `onDemandPrice` to `reservedYearPrice`. Cloud providers offer discounts for reserved instances.

```

curl -X GET \
http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/Enterprise_HR_SAA
S_Application/provider/aws/pricetype/reservedYearPrice/ \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json'

```

```

{
  "application": {
    "app_name": "Enterprise_HR_SAAS_Application",
    "date_created": "2019-09-20T01:51:24.368595Z",
    "pk_hash": "89a8e4463ed6ab13713d7ad8c82b40e23d6dec023348d49625e74885",
    "services": [

```

```
{
  "attributes": [
    {
      "attr_id": "unit",
      "value": "instance"
    },
    {
      "attr_id": "util",
      "value": "40"
    },
    {
      "attr_id": "max_util",
      "value": "80"
    },
    {
      "attr_id": "quantity",
      "value": "8"
    }
  ],
  "service_type": "aws-c5.xlarge",
  "date_created": "2019-09-20T01:51:24.416991Z",
  "pk_hash":
  "881a3b377b41ff3c8d6084d79f8e1bfa42d9fc45f9642ef9eead88b8",
  "location": "us-west-2",
  "price_type": "onDemandPrice",
  "provider": "aws"
},
{
  "attributes": [
    {
      "attr_id": "unit",
      "value": "instance"
    },
    {
      "attr_id": "util",
      "value": "30"
    },
    {
      "attr_id": "max_util",
      "value": "80"
    },
    {
      "attr_id": "quantity",
      "value": "3"
    }
  ],
  "service_type": "aws-c5.2xlarge",
  "date_created": "2019-09-20T01:51:24.485717Z",
  "pk_hash":
  "45ddf322bad20dae58a7cf736493e02a7d178448154daec065fa0398",
  "location": "us-east-2",
  "price_type": "onDemandPrice",
  "provider": "aws"
}
```



```
    ],
    "description": "Finance department application",
    "metadata": "this can be any string up to 1024 characters long, for
example, fookey=barvalue",
  },
  "current_application": {
    "components": [
      {
        "currentGen": "True",
        "numcpus": "4",
        "memory": "7.45",
        "quantity": "8",
        "util": "40",
        "max_util": "80",
        "service_type": "aws-c5.xlarge",
        "unit_cost": 0.17,
        "unit": "",
        "location": "us-west-2",
        "service_cost": 1.36,
        "price_type": "onDemandPrice"
      },
      {
        "currentGen": "True",
        "memory": "15.1",
        "numcpus": "8",
        "quantity": "3",
        "util": "30",
        "max_util": "80",
        "service_type": "aws-c5.2xlarge",
        "unit_cost": 0.34,
        "unit": "",
        "location": "us-east-2",
        "service_cost": 1.02,
        "price_type": "onDemandPrice"
      }
    ]
  },
  "total_cost": 2.38
},
"alternative_option": {
  "components": [
    {
      "servicetype": "aws-t3.xlarge",
      "machine_cpu": 41.56805002183218,
      "numcpus": "4",
      "memory": "15.48",
      "costph": 0.104,
      "description": "aws-t3.xlarge",
      "price_type": "reservedlyearPrice",
      "location": "us-west-2",
      "service_cost": 0.832
    },
    {
      "servicetype": "aws-t3.xlarge",
      "machine_cpu": 65.54483646460177,
```

```

        "numcpus": "4",
        "memory": "15.48",
        "costph": 0.104,
        "description": "aws-t3.xlarge",
        "price_type": "reserved1yearPrice",
        "location": "us-east-2",
        "service_cost": 0.312
    }
],
"total_cost": 1.14
}
}

```

1.2.7. Updating an Application

Applications can be updated using a **PUT** request. Currently all the service instances of an application need to be necessarily specified even if only one is changed. The name of an application cannot be changed; a new application should instead be created via **POST** request. The description of the application can be updated.

The following **PUT** request changes some of the service instances of the application. We can alternately use the **pk**, as illustrated in the **GET** method for a specific application above.:

```

curl -X PUT \
http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/Enterprise_HR_SAA
S_Application/ \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json' \
-d '{
  "app_name": "Enterprise_HR_SAAS_Application",
  "services": [
    {
      "service_type": "aws-c5.2xlarge",
      "quantity": 4,
      "location": "us-west-2",
      "price_type": "reserved1yearPrice",
      "provider": "aws",
      "attributes": [
        {
          "attr_id": "unit",
          "value": "instance"
        },
        {
          "attr_id": "util",
          "value": "20"
        }
      ]
    }
  ]
}

```

```

        "attr_id": "max_util",
        "value": "30"
      },
      {
        "attr_id": "quantity",
        "value": "4"
      }
    ]
  },
  {
    "service_type": "aws-c5.2xlarge",
    "quantity": 3,
    "location": "us-east-2",
    "price_type": "onDemandPrice",
    "provider": "aws",
    "attributes": [
      {
        "attr_id": "unit",
        "value": "instance"
      },
      {
        "attr_id": "util",
        "value": "30"
      },
      {
        "attr_id": "max_util",
        "value": "80"
      },
      {
        "attr_id": "quantity",
        "value": "3"
      }
    ]
  }
],
"description": "HR department application",
"metadata": "Updated the metadata!"
}'

```

Returns 200 OK if the update succeeds.

The updated application is returned.

```

{
  "app_name": "Enterprise_HR_SAAS_Application",
  "date_created": "2019-09-20T01:51:24.368595Z",
  "pk_hash": "89a8e4463ed6ab13713d7ad8c82b40e23d6dec023348d49625e74885",
  "services": [
    {
      "attributes": [
        {
          "attr_id": "unit",

```

```
        "value": "instance"
      },
      {
        "attr_id": "util",
        "value": "20"
      },
      {
        "attr_id": "max_util",
        "value": "30"
      },
      {
        "attr_id": "quantity",
        "value": "4"
      }
    ],
    "service_type": "aws-c5.2xlarge",
    "date_created": "2019-09-21T02:37:21.035897Z",
    "pk_hash":
"379fe4b040f72c9fc94d96f99e760b2627f2b2b5c76ecbd3931fb479",
    "location": "us-west-2",
    "price_type": "reserved1yearPrice",
    "provider": "aws"
  },
  {
    "attributes": [
      {
        "attr_id": "unit",
        "value": "instance"
      },
      {
        "attr_id": "util",
        "value": "30"
      },
      {
        "attr_id": "max_util",
        "value": "80"
      },
      {
        "attr_id": "quantity",
        "value": "3"
      }
    ],
    "service_type": "aws-c5.2xlarge",
    "date_created": "2019-09-21T02:37:21.154060Z",
    "pk_hash":
"45ddf322bad20dae58a7cf736493e02a7d178448154daec065fa0398",
    "location": "us-east-2",
    "price_type": "onDemandPrice",
    "provider": "aws"
  }
],
"description": "HR department application",
"metadata": "Updated the metadata!"
}
```

1.2.8. Deleting an Application

Applications may be deleted as follows; as with the `GET` method, `app_name` or `pk` can be used to construct the URL.

```
curl -X DELETE \  
  
http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/Enterprise_HR_SAA  
S_Application/ \  
-H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX  
CpGBRNendUy2Kw' \  
-H 'Content-Type: application/json' \  

```

This returns `204 No Content` on successful deletion.

1.2.9 Deleting multiple Applications

Multiple applications may be deleted in one request.

```
curl -X POST \  
http://127.0.0.1:8000/api/v1.0/calc/applications/delete/ \  
-H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX  
CpGBRNendUy2Kw' \  
-H 'Content-Type: application/json' \  
-d '[  
  "Enterprise_HR_SAAS_Application",  
  "Enterprise_HR_SAAS_Application2"  
]'
```

This returns `204 No Content` on successful deletion.

1.2.10 Listing all Applications

The user may retrieve a list of all of the applications belonging to them like so:

```
curl -X GET \  
http://127.0.0.1:8000/api/v1.0/calc/applications/ \  
-H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
```

```
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json' \
```

API results are paginated, specify the limit and offset of the results like so:

```
curl -X GET \
  http://127.0.0.1:8000/api/v1.0/calc/applications/?limit=5&offset=50 \
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
  -H 'Content-Type: application/json' \
```

Without specifying these parameters, the default limit is 10 and the default offset is 0.

1.2.10.1 Filtering and sorting Applications

The user may narrow down their application list by adding filter parameters. The following are available:

- `name_contains` - Get only applications whose name contains this substring
- `metadata_contains` - Get only applications whose metadata contains this substring
- `min_time` - Get only applications whose creation time is later than this time value
- `max_time` - Get only applications whose creation time is earlier than this time value

Note: Times have to be specified in following format:

```
YYYY-MM-DD HH:MM[:ss[.uuuuuu]] [TZ]
# Some Valid DateTime Examples:
2019-12-15
2020-01-01
2019-12-12 9:55:57
```

Furthermore, the user may specify a property to sort the results by setting the `sort` parameter to one of the following values:

- `date` - The date created
- `name` - The application's name
- `service_instances` - The number of service instances associated with the application

To reverse the sort order, prepend the character `-` to the value of `sort`

For example, display applications that were created after June 1st, 2020, who contain the substring 'MyOrganization' in their name, and sort them in decreasing order of the number of service instances associated with each:

```
curl -X GET \
  http://127.0.0.1:8000/api/v1.0/calc/applications/?
  name_contains=MyOrganization&min_time=2020-06-01&sort=-service_instances \
  -H 'Authorization: JWT
  eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
  VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
  CpGBRNendUy2Kw' \
  -H 'Content-Type: application/json' \
```

1.3. Prices

Prices for public cloud services are included in the database; however users can add custom prices.

1.3.1. Creating a new price for a service (POST)

Each service type can have multiple prices associated with it. For example, public cloud providers have `onDemandPrice`, `reserved1yearPrice` and `reserved3yearPrice`. Users can define their own price types, for example, if a user obtains special pricing from a provider. All prices are expressed as floating numbers - \$USD per hour.

```
curl -X POST \
  http://127.0.0.1:8000/api/v1.0/calc/price/ \
  -H 'Authorization: JWT
  eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
  VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
  CpGBRNendUy2Kw' \
  -H 'Content-Type: application/json' \
  -d '{
    "service_pk":
    "45d7822c23fe69282e5640092775050834c1b7ec987d92d2a206c3a8",
    "name": "examplecorp-aws-valued-customer-price",
    "costperunit": 0.001
  }'
```

The returned output is

```
{
  "service_name": "aws-t3.small",
  "name": "examplecorp-aws-valued-customer-price",
  "service_pk":
  "45d7822c23fe69282e5640092775050834c1b7ec987d92d2a206c3a8",
  "costperunit": 0.001,
  "location": "us-east-2"
}
```

This returns `201 created` on success.

Note that the `location` key is deciphered from the service location.

1.3.2. Get price details about a pre-existing price (GET)

Details about a pre-existing pricetype can be queried either by using the `pk` of the price or by using a combination of the `service_pk` and the `pricetype` (the name of the price like "onDemandPrice").

Using `pk`:

```
curl -X GET \  
  
http://127.0.0.1:8000/api/v1.0/calc/price/pk/789a89b8bcc4ce13552cfdd882cee4  
b130bbf1db0ad3d83b2a1600a9/ \  
  -H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOiE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX  
CpGBRNendUy2Kw' \  
  -H 'Content-Type: application/json' \  

```

...OR

```
curl -X GET \  
  
http://127.0.0.1:8000/api/v1.0/calc/price/service_pk/45d7822c23fe69282e5640  
092775050834c1b7ec987d92d2a206c3a8/pricetype/examplecorp-aws-valued-  
customer-price/ \  
  -H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOiE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX  
CpGBRNendUy2Kw' \  
  -H 'Content-Type: application/json' \  

```

Both the approaches yield the price detail

```
{  
  "service_name": "aws-t3.small",  
  "pk": "dc0c96587629b22e441020309b37227cb885632e2b09caba40cee5be",  
  "name": "examplecorp-aws-valued-customer-price",  
  "service_pk": "45d7822c23fe69282e5640092775050834c1b7ec987d92d2a206c3a8",  
  "costperunit": 0.001,  
  "location": "us-east-2"  
}
```

1.3.3. Updating the costperunit for a pre-existing price (PUT)

The `costperunit` value can be updated for any price (as long as it is "owned" - originally created - by the user). Again, the `service_pk` and `pricetype` or the `pk` of the price can be used as in the `GET` request above.

```
curl -X PUT \

http://127.0.0.1:8000/api/v1.0/calc/price/service_pk/45d7822c23fe69282e5640
092775050834c1b7ec987d92d2a206c3a8/pricetype/examplecorp-aws-valued-
customer-price/ \
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
  -H 'Content-Type: application/json' \
  -d '{
    "costperunit": 0.005
  }'
```

The modified price is returned

```
{
  "service_name": "aws-t3.small",
  "pk": "dc0c96587629b22e441020309b37227cb885632e2b09caba40cee5be",
  "name": "examplecorp-aws-valued-customer-price",
  "service_pk": "45d7822c23fe69282e5640092775050834c1b7ec987d92d2a206c3a8",
  "costperunit": 0.005,
  "location": "us-east-2"
}
```

A `200 OK` is returned on success.

1.3.4. Deleting a price (DELETE)

Similarly, a price can be deleted using the `service_pk` and `pricetype` or the `pk` of the price can be used as in the `GET` request above.

Using `service_pk` and `pricetype`

```
curl -X DELETE \

http://127.0.0.1:8000/api/v1.0/calc/price/service_pk/45d7822c23fe69282e5640
092775050834c1b7ec987d92d2a206c3a8/pricetype/examplecorp-aws-valued-
customer-price/ \
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
  -H 'Content-Type: application/json' \
```

...or using the price `pk`

```
curl -X DELETE \  
  
http://127.0.0.1:8000/api/v1.0/calc/price/pk/dc0c96587629b22e441020309b3722  
7cb885632e2b09caba40cee5be/ \  
-H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX  
CpGBRNendUy2Kw' \  
-H 'Content-Type: application/json' \  

```

A 204 No Content is returned.

1.3.5 Deleting multiple prices (POST)

Multiple prices may be deleted in one request using the `pks` of the prices.

```
curl -X POST \  
http://127.0.0.1:8000/api/v1.0/calc/price/delete/ \  
-H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX  
CpGBRNendUy2Kw' \  
-H 'Content-Type: application/json' \  
-d '[  
  "dc0c96587629b22e441020309b37227cb885632e2b09caba40cee5be",  
  "7cb885632e2b09caba40cee5bedc0c96587629b22e441020309b3722",  
  ]'  

```

A 204 No Content is returned.

1.3.6. Listing all prices (GET)

The user may retrieve a list of all generic or user-owned prices like so:

```
curl -X GET \  
http://127.0.0.1:8000/api/v1.0/calc/price/ \  
-H 'Authorization: JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm  
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX  
CpGBRNendUy2Kw' \  
-H 'Content-Type: application/json' \  

```

API results are paginated, specify the limit and offset of the results as follows:

```
curl -X GET \
  http://127.0.0.1:8000/api/v1.0/calc/price/?limit=50&offset=10 \
  -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
  -H 'Content-Type: application/json' \
```

Without specifying these parameters, the default limit is 10 and the default offset is 0

1.3.6.1. Filtering and sorting prices

The user may further specify their price list by adding filter parameters. The following are available:

- `name_contains` - Get only prices whose name contains this substring
- `location_contains` - Get only prices whose location contains this substring
- `service_name_contains` - Get only prices whose associated service type's name contains this substring
- `service_pk` - Get only prices whose associated service is the one specified by this pk value
- `min_cost` - Get only prices whose cost-per-unit is greater than or equal to this value
- `max_cost` - Get only prices whose cost-per-unit is less than or equal to this value
- `min_time` - Get only prices who were created later than this time
- `max_time` - Get only prices who were created earlier than this time
- `user_defined` - Get only prices who are user-defined if this value is 'yes', or not user-defined if this value is 'no'

Note: Times have to be specified in following format:

```
YYYY-MM-DD HH:MM[:ss[.uuuuuu]][TZ]
# Some Valid DateTime Examples:
2019-12-15
2020-01-01
2019-12-12 9:55:57
```

Alternatively, the following API endpoints are available.

To get all prices belonging to a service type:

```
```bash
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/price/service_pk/ffe6434785a8cca79c3837
aec689e8294715c7330f778324d6a1a274/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
```

## To get all prices of a particular price type

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/price/pricetype/examplecorp-aws-
valued-customer-reduced-price/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlE1NjY2OTU5ImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json'
```

## To get all prices belonging to a service type AND of a particular price type:

```
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/price/service_pk/ffc653c1505713081527f1
a39894d5fe16cebd8f81593c6622165cc7/pricetype/reservedlyearPrice/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlE1NjY2OTU5ImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json'
```

For all three endpoints, results are paginated, and on success return a **200 OK**.

Furthermore, the user may specify a property to sort the results by, achieved by setting the `sort` parameter to one of the following values:

- `name` - The price's name
- `location` - The location name
- `service` - The associated service type's name
- `date` - The creation date
- `cost` - The cost-per-unit
- `userdefined` - Whether or not the price is user-defined (non-user defined first, then user-defined)

To reverse the sort order, prepend the character `-` to the value of `sort`

For example, get prices that were created before June 1st, 2020, whose location is in the US, and sort them by location name:

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/price/?max_time=2020-06-
01&location_contains=us&sort=location \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlE1NjY2OTU5ImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
```

```
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json' \
```

## 1.4. ServiceTypes

In addition to the public cloud service types, additional service types can also be added. For example, a service type to capture the "server admin" component of an application.

### 1.4.1. Create a new service type (POST)

Create new service type like so

```
curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/servicetype/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkyMTAyOTUsImVtYWlsIjoieIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
 -d '{
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "60"
 },
 {
 "attr_id": "unit",
 "value": "instance"
 },
 {
 "attr_id": "util",
 "value": "30"
 },
 {
 "attr_id": "max_util",
 "value": "90"
 }
],
 "name": "my_misc_service_1",
 "location": "eastus2",
 "type": "compute",
 "unit": "undefined",
 "provider": "alt"
 }'
```

The new service type is returned as a JSON string.

```
{
 "name": "my_misc_service_1",
```

```
"provider": "alt",
"location": "eastus2",
"type": "compute",
"unit": "undefined",
"attributes": [
 {
 "attr_id": "quantity",
 "value": "60"
 },
 {
 "attr_id": "unit",
 "value": "instance"
 },
 {
 "attr_id": "util",
 "value": "30"
 },
 {
 "attr_id": "max_util",
 "value": "90"
 }
],
"pk_hash": "e72e11be451152e0758d18248d700e05c36e0b0a4a1929f612fc6ea6",
"user_defined": true
}
```

The `user_defined` field indicates that the servicetype has been created by an end-user (and is not a "standard" servicetype that BigBitBus shipped. This boolean will be useful when deciding if a delete request should be honoured or not.

A 201 `Created` code is returned.

#### 1.4.2. Retrieve a service type (GET)

We can use the `pk` or a combination of (provider, service name and location) to get details of a certain servicetype.

```
curl -X GET \
http://127.0.0.1:8000/api/v1.0/calc/servicetype/pk/e72e11be451152e0758d1824
8d700e05c36e0b0a4a1929f612fc6ea6 \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJE1NjkkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json' \
```

or

```
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/servicetype/provider/alt/name/my_misc_s
ervice_1/location/eastus2 \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json'
```

## yields

```
{
 "name": "my_misc_service_1",
 "provider": "alt",
 "location": "eastus2",
 "type": "compute",
 "unit": "undefined",
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "60"
 },
 {
 "attr_id": "unit",
 "value": "instance"
 },
 {
 "attr_id": "util",
 "value": "30"
 },
 {
 "attr_id": "max_util",
 "value": "90"
 }
],
 "pk_hash": "e72e11be451152e0758d18248d700e05c36e0b0a4a1929f612fc6ea6",
 "user_defined": true
}
```

A 200 OK code is returned.

### 1.4.3. Deleting a service type (DELETE)

A user can delete a service type if the service type is user-defined (e.g. created by the user via a POST request earlier).

```
curl -X DELETE \

http://127.0.0.1:8000/api/v1.0/calc/servicetype/pk/e72e11be451152e0758d1824
8d700e05c36e0b0a4a1929f612fc6ea6/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json'
```

or

```
curl -X DELETE \

http://127.0.0.1:8000/api/v1.0/calc/servicetype/provider/alt/name/my_misc_s
ervice_1/location/eastus2/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json'
```

A 204 No Content is returned.

#### 1.4.4 Deleting multiple service types (POST)

A user can delete multiple service types if they are all user-defined.

```
curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/servicetype/delete/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
 -d '[
 "e72e11be451152e0758d18248d700e05c36e0b0a4a1929f612fc6ea6",
 "b971a4a221ad29cfb3b0558e37689dfb460d590f108f45f7ea7dc751"
]'
```

#### 1.4.5 Listing all service types (GET)

The user may retrieve a list of all generic or user-owned service types like so:



```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/servicetype/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \

```

API results are paginated, specify the limit and offset of the results as follows:

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/servicetype/?limit=5&offset=50 \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \

```

Without specifying these parameters, the default limit is 10 and the default offset is 0.

#### 1.4.5.1 Filtering and sorting service types

The user may narrow down their service type list by adding filter parameters. The following are available:

- `name_contains` - Get only service types whose name contains this substring
- `location_contains` - Get only service types whose location contains this substring
- `user_defined` - Get only service types whose are user-defined is the value is 'yes', or not user-defined if the value is 'no'
- `min_time` - Get only service types whose were last updated later than this time value
- `max_time` - Get only service types whose were last updated earlier than this time value
- `min_memory` - Get only service types whose memory capacity is greater than or equal then this value
- `max_memory` - Get only service types whose memory capacity is less than or equal then this value
- `min_cpus` - Get only service types whose number of CPUs is greater than or equal then this value
- `max_cpus` - Get only service types whose number of CPUs is less than or equal then this value

**Note:** Times have to be specified in following format:

```
YYYY-MM-DD HH:MM[:ss[.uuuuuu]][TZ]
Some Valid DateTime Examples:
2019-12-15
2020-01-01
2019-12-12 9:55:57
```

Furthermore, the user may specify a property to sort the results by setting the `sort` paramter to one of the following values:

- `date` - The date updated
- `name` - The service type's name
- `location` - The service type's location
- `userdefined` - Whether or not the service type was created by the user (non-user defined first, then user-defined)
- `memory` - The amount of memory the service type has
- `cpus` - The numer of CPUs the service type has

To reverse the sort order, prepend the character `-` to the value of `sort`

For example, display service types that were created after June 1st, 2020, whose location is in the US, and sort them in decreasing order of the number of CPUs that each has:

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/servicetype/?
 location_contains=us&min_time=2020-06-01&sort=-cpus \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
```

## 1.4.6. Finding service types

### 1.4.6.1. Service type keyword search

The `servicessearch` endpoint can be used for keyword search. A list of space-separated terms are sent to the API server, as shown in these examples.

Search for "aws-t3.small" in "us-east"

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/servicessearch/aws-t3.small%20us-east/
 \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQoFk' \
 -H 'Content-Type: application/json'
```

Search for "aze" "f4" "korea" for Azure VMs with "f4" in their name in a Korean region.

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/servicessearch/aze%20f4%20korea/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
```

```
i5D3CpCiKQoFk' \
-H 'Content-Type: application/json'
```

### 1.4.6.2. Service type name autocomplete

The `serviceautocomplete` endpoint can be used for to autocomplete a partial name, demonstrated as follows:

Search for services prefixed by "aws-t3"

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/serviceautocomplete/aws-t3/ \
 -H 'Authorization: JWT
eyJ0eXAIoiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMIeHAIoiE1NjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQoFk' \
 -H 'Content-Type: application/json'
```

which returns

```
[
 "aws-t3.2xlarge",
 "aws-t3a.2xlarge",
 "aws-t3a.large",
 "aws-t3a.medium",
 "aws-t3a.micro",
 "aws-t3a.nano",
 "aws-t3a.small",
 "aws-t3a.xlarge",
 "aws-t3.large",
 "aws-t3.medium",
 "aws-t3.micro",
 "aws-t3.nano",
 "aws-t3.small",
 "aws-t3.xlarge"
]
```

### 1.4.6.3 Getting service type info

The `servicetypeinfo` endpoint can be used for to find locations, prices and providers for a service type

Search for information on "aws-t3.2xlarge"

```
curl --location --request GET
'127.0.0.1:8000/api/v1.0/calc/servicetypeinfo/aws-t3.micro/' \
--header 'Authorization: JWT'
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t1b190eXB1IjoieWNjZXNzIiwiaXhwIjoxNTg4NjkyNDU5LCJqdGkiOiI2YjAzYzBjYTQ3ZTc0ZjMwOWUxZmY0NmZlYzc2MjI4YiIsInVzZXJfaWQiOiJ9.KZEcZthAKC58yhZSKdczIs8bpNWuQq7N1ChrjISt-QQ' \
-H 'Content-Type: application/json'
```

which returns

```
"locations": [
 "ap-east-1",
 "ap-northeast-1",
 "ap-northeast-2",
 "ap-south-1",
 "ap-southeast-1",
 "ap-southeast-2",
 "ca-central-1",
 "eu-central-1",
 "eu-north-1",
 "eu-west-1",
 "eu-west-3",
 "me-south-1",
 "sa-east-1",
 "us-east-1",
 "us-east-2",
 "us-west-1",
 "us-west-2"
],
"prices": [
 "onDemandPrice",
 "reserved3yearPrice",
 "reserved1yearPrice"
],
"providers": [
 "aws"
],
"locationpricemap": {
 "ap-east-1": [
 "onDemandPrice",
 "reserved1yearPrice",
 "reserved3yearPrice"
],
 "ap-northeast-1": [
 "onDemandPrice"
],
 "ap-northeast-2": [
 "onDemandPrice",
 "reserved1yearPrice",
 "reserved3yearPrice"
],
 ...
}
```

The `prices` key is the union of all the prices available at the list of `locations`. The `locationpricemap` shows which price type is available at a location. This is useful because not all price types may be available at all locations. For example, the `ap-northeast-1` location in the above example only has `onDemandPrice`.

#### 1.4.6.4. Compare two service types

This API returns a data structure that makes it easy to compare the fields of two different service types.

```
curl -X GET \

http://127.0.0.1:8000/api/v2.0/calc/servicetype/pk1/8e108e383d32cc10be23ecf0d8a3dea50ce3b7246b5fe3ea03d9cdb0/pk2/ca540b71715a20ec928c537406c702b2692cd505c3541a3c83a83e89/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6ImdlbmVyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bhI5D3CpCiKQqoFk' \
 -H 'Content-Type: application/json' \
```

or

```
curl -X GET \

http://127.0.0.1:8000/api/v2.0/calc/servicetype/provider1/alt/name1/my_misc_service_1/location1/eastus2/provider2/alt/name2/my_misc_service_2/location2/eastus2/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6ImdlbmVyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bhI5D3CpCiKQqoFk' \
 -H 'Content-Type: application/json' \
```

yields

```
{
 "unit": {
 "1": "undefined",
 "2": "undefined"
 },
 "provider": {
 "1": "alt",
 "2": "alt"
 },
 "type": {
```

```
 "1": "compute",
 "2": "compute"
 },
 "location": {
 "1": "eastus2",
 "2": "eastus2"
 },
 "pk_hash": {
 "1": "8e108e383d32cc10be23ecf0d8a3dea50ce3b7246b5fe3ea03d9cdb0",
 "2": "ca540b71715a20ec928c537406c702b2692cd505c3541a3c83a83e89"
 },
 "attributes": [
 {
 "attr_id": "quantity",
 "value": {
 "1": "60",
 "2": "45"
 }
 },
 {
 "attr_id": "unit",
 "value": {
 "1": "instance",
 "2": "instance"
 }
 },
 {
 "attr_id": "util",
 "value": {
 "1": "30",
 "2": "15"
 }
 },
 {
 "attr_id": "max_util",
 "value": {
 "1": "90",
 "2": "85"
 }
 },
 {
 "attr_id": "foo",
 "value": {
 "1": "bar",
 "2": "bar"
 }
 }
],
 "name": {
 "1": "my_misc_service_1",
 "2": "my_misc_service_2"
 },
 "version_date": {
 "1": "2020-06-05T15:11:15.742498Z",
```

```
 "2": "2020-06-05T15:11:20.204243Z"
 },
 "user_defined": {
 "1": true,
 "2": true
 },
 "enabled": {
 "1": true,
 "2": true
 },
 "description": {
 "1": "",
 "2": ""
 }
}
```

In both cases, a 200 OK code is returned.

## 1.5. ProviderDiscounts

Users may add discounts they receive from using certain providers. For example, a 30% price discount for machines that use amazon web services (AWS)

### 1.5.1. Create a new discount (POST)

Create new discount like so

```
curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/discount/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjksMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
 -d '{
 "provider": "aws",
 "discount": "30"
 }'
```

The new discount is returned as a JSON string.

```
{
 "pk_hash": "b42385356271f4a9e3945abe2c3daff54d41af4ad0977394805437ec",
 "provider": {
 "name": "aws",
 "owner": "generic"
 },
 "discount": "30"
}
```

Note that discount percentages must be expressed as an integer between 0 and 100, and further that only one discount may exist for each provider. In the case that there is both a generic-owned and user-owned provider of the same name, the user-owned one will be used.

A 201 `Created` code is returned.

### 1.5.2. Retrieve a discount (GET)

We can use the pk to get details of a discount.

```
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/discount/pk/e72e11be451152e0758d18248d7
00e05c36e0b0a4a1929f612fc6ea6/ \
 -H 'Authorization: JWT'
```



```

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImdlbmVyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json' \

```

## yields

```

{
 "pk_hash": "b42385356271f4a9e3945abe2c3daff54d41af4ad0977394805437ec",
 "provider": {
 "name": "aws",
 "owner": "generic"
 },
 "discount": "30"
}

```

A 200 OK code is returned.

### 1.5.3. Updating the discount amount for a pre-existing discount (PUT)

the `discount` value can be updated for any provider discount(as long as it is "owned" - originally created - by the user). The `pk` of the discount can be used as in the `get` request above.

```

curl -x put \

http://127.0.0.1:8000/api/v1.0/calc/discount/pk/b42385356271f4a9e3945abe2c3
daff54d41af4ad0977394805437ec/ \
-h 'authorization: jwt
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImdlbmVyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-h 'content-type: application/json' \
-d '{
 "provider": "aws",
 "discount": "40"
}'

```

the modified discount is returned

```

{
 "pk_hash": "b42385356271f4a9e3945abe2c3daff54d41af4ad0977394805437ec",
 "provider": {
 "name": "aws",
 "owner": "generic"
 },

```

```
"discount": "30"
}
```

a 200 `ok` is returned on success.

#### 1.5.4. Deleting a discount (DELETE)

A user can delete a discount for any provider.

```
curl -X DELETE \

http://127.0.0.1:8000/api/v1.0/calc/discount/pk/e72e11be451152e0758d18248d7
00e05c36e0b0a4a1929f612fc6ea6/ \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json'
```

A 204 `No Content` is returned.

#### 1.5.5. Listing all discounts

This API call returns all discounts belonging to a user.

```
curl -X GET \

http://127.0.0.1:8000/api/v1.0/calc/discount/ \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOjE1NjkxODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQQoFk' \
-H 'Content-Type: application/json' \
```

Limit the size of the response using pagination, like so:

```
curl -X GET \

'http://127.0.0.1:8000/api/v1.0/calc/discount/?limit=5&offset=4' \
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOjE1NjkxODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQQoFk' \
-H 'Content-Type: application/json' \
```

## 1.5.5. Listing all discounts

The user may retrieve a list of all discounts belonging to the user like so:

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/discount/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOjE1NjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQoFk' \
 -H 'Content-Type: application/json' \

```

API results are paginated, specify the limit and offset of the results as follows:

```
curl -X GET \
 'http://127.0.0.1:8000/api/v1.0/calc/discount/?limit=5&offset=4' \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOjE1NjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQoFk' \
 -H 'Content-Type: application/json' \

```

Without specifying these parameters, the default limit is 10, and the default offset is 0

### 1.5.5.1. Filtering and sorting discounts

The user may further specialize their list of provider discounts using filter parameters. The following are available:

- `provider` - Get only provider discounts for the provider whose short name is this value
- `min_discount` - Get only discounts whose percentage value is above or equal this value
- `max_discount` - Get only discounts whose percentage value is below or equal this value

Furthermore, the user may specify a property to sort the results by, achieved by setting the `sort` parameter one of the following values:

- `provider_name` - The associated provider's full name
- `provider_short_name` - The associated provider's short name

To reverse the sort order, prepend the character `-` to the value of `sort`

For example, display discounts for the provider 'alt' greater than 15%, and sorted by the percent value

```
curl -X GET \
 'http://127.0.0.1:8000/api/v1.0/calc/discount/?
provider_name=alt&min_discount=15&sort=provider_name' \

```

```
-H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQQoFk' \
-H 'Content-Type: application/json' \
```

## 1.6. Providers

Users can create custom providers. This may be useful in order to add servetypes corresponding to a private datacenter or cloud and adding customized costs to their applications. For example, a user may create a "provider" called `overheads` and add "servicetypes" for `hr_admin_overhead` and `network_cost_per_server`. Lets work through an example below.

### 1.6.1. Create a new provider (POST)

First, create the provider

```
curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/provider/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImJic2
VydmIjZWFjY291bnQiLCJleHAiOiJlNzQ5NjIyMzIsImVtYWlsIjoiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULE' \
 -H 'Content-Type: application/json' \
 -d '{
 "name": "overheads",
 "description": "Provider to capture custom overheads for any
application"
 }'
```

We note that the backend auto-assigns a 3-letter `short_name` to the new provider corresponding to the first 3 letters of the `name` (lowercased if needed). If we were to create a second provider with the same prefix (e.g. `overtheheads`) then the backend would choose a random 3-letter `short_name` and assign it to the second provider.

Optionally, a `short_name` may be explicitly specified as follows:

```
curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/provider/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImJic2
VydmIjZWFjY291bnQiLCJleHAiOiJlNzQ5NjIyMzIsImVtYWlsIjoiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULE' \
 -H 'Content-Type: application/json' \
 -d '{
 "name": "overheads",
 "short_name": "ove",
 "description": "Provider to capture custom overheads for any
```

```
application"
}'
```

Yielding the same successful response as above if the name or short name is not already taken.

## 1.6.2. Retrieve a provider (GET)

We can check that this `overheads` provider has been correctly created

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/provider/name/ove \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJlc2VybmFtZSI6ImJic2
VydmljZWJyY291bnQiLCJleHAiOjE1NzQ5NjIyMzIsImVtYWlsIjoiiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULE' \
 -H 'Content-Type: application/json'
```

```
{
 "name": "overheads",
 "short_name": "ove",
 "owner": "bbserviceaccount",
 "description": "Provider to capture custom overheads for any
application",
 "user_defined": true,
 "enabled": true
}
```

In the endpoint, after 'name', the user may specify either the provider's short name or full name. In ambiguous cases, where the name may suggest more than one provider, the following order of precedence is used to decide which one to return:

1. The provider that is user-owned and whose short name matches the provided 'name'
2. The provider that is user-owned and whose name matches the provided 'name'
3. The provider that is generic and whose short name matches the provided 'name'
4. The provider that is generic and whose name matches the provided 'name'

## 1.6.3. Using providers

### 1.6.3.1. Creating a service type with the provider

Next, lets create a service type called `hr_admin_overhead`

```
curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/servicetype/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJlc2VybmFtZSI6ImJic2
VydmljZWJyY291bnQiLCJleHAiOjE1NzQ5NjIyMzIsImVtYWlsIjoiiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULE'
```

```
VydmljZWFjY291bnQiLCJleHAiOjE1NzQ5NjIyMzIsImVtYWlsIjoiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULe' \
-H 'Content-Type: application/json' \
-d '{
 "attributes": [
],
 "name": "hr_admin_overhead",
 "location": "anywhere",
 "type": "other",
 "unit": "per server",
 "provider": "ove",
 "description": "This servicetype embodies the human resource admin cost
overhead per server"
}'
```

### The POST request returns

```
{
 "name": "hr_admin_overhead",
 "provider": "overheads",
 "location": "anywhere",
 "type": "other",
 "unit": "per server",
 "attributes": [],
 "pk_hash": "512c4c227fc31be9291f0c8e392dc930f1b1f9e0b20776df81e5a986",
 "description": "This servicetype embodies the human resource admin cost
overhead per server",
 "version_date": "2019-11-28T16:12:07.011819Z",
 "user_defined": true
}
```

#### 1.6.3.2. Creating a price with the provider

Next, lets add a custom price to the `hr_admin_overhead`. In this example the organization runs 1,000 servers with an IT team comprising of 15 people. Each team-member costs \$100,000 on average per year. So the HR admin overhead per server, per hour is

$(100000 * 15) / 1000 = \$1500$  per server, per year; or about \$0.17 per server, per hour.

So lets post a custom price to the `hr_admin_overhead` service.

```
curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/price/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJlc2VybmFtZSI6ImJic2
VydmljZWFjY291bnQiLCJleHAiOjE1NzQ5NjIyMzIsImVtYWlsIjoiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULe' \
 -H 'Content-Type: application/json' \
 -d '{
```

```

 "service_name": "hr_admin_overhead",
 "name": "hr_overhead_cost",
 "service_pk":
"512c4c227fc31be9291f0c8e392dc930f1b1f9e0b20776df81e5a986",
 "costperunit": 0.17,
 "location": "anywhere"
}'

```

Now we are ready to add the `hr_admin_overhead` to an application. In this case lets assume the application is running in AWS. The VMs will be selected from the Azure `aze` provider and the `hr_admin_overhead` will be selected from the `overheads` provider we just created. Like so

### 1.6.3.3. Reports and Optimization

```

curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/applications/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiLCJlc2VybmFtZSI6ImJic2
VydmIjZWFjY291bnQiLCJleHAiOiE1NzUwNDkxNTcsImVtYWlsIjoiIn0.PhWO2M7kcA280dCM2
6NjsGYGDOK4dnGFLY01KSz-SyM' \
 -H 'Content-Type: application/json' \
 -d '{
 "app_name": "User-application-in-AZURE, with overhead added",
 "description": "This is an application the customer currently runs in
AZURE public cloud and has hr admin overheads",
 "services": [
 {
 "service_type": "aze-standard_f2",
 "location": "canadacentral",
 "price_type": "onDemandPrice",
 "provider": "aze",
 "quantity": 8,
 "attributes": [
 {
 "attr_id": "quantity",
 "value": 8
 }
]
 },
 {
 "service_type": "aze-standard_b8ms",
 "location": "eastus2",
 "price_type": "onDemandPrice",
 "provider": "aze",
 "quantity": 9,
 "attributes": [
 {
 "attr_id": "quantity",
 "value": 9
 }
]
 }
]
 }'

```

```

 },
 {
 "service_type": "hr_admin_overhead",
 "location": "anywhere",
 "price_type": "hr_overhead_cost",
 "provider": "ove",
 "quantity": 17,
 "attributes": [
 {
 "attr_id": "quantity",
 "value": 17
 }
]
 }
]
}
'

```

Note the quantity has been set to 0.17 - since we calculated earlier that each server has a \$0.17 hr admin overhead per hour.

Next, lets got a cost report on the application

```

curl -X GET \
 'http://127.0.0.1:8000/api/v1.0/calc/reports/applications/app_name/User-application-in-AZURE,%20with%20overhead%20added/' \
 -H 'Authorization: JWT eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJlc2VybmFtZSI6ImJic2VydmVjZWFjY291bnQiLCJleHAiOjE1NzUwNDkxNTcsImVtYWlsIjoiiIn0.PhWO2M7kcA280dCM26NjsGYGDOK4dnGFLY01KSz-SyM' \
 -H 'Content-Type: application/json' \

```

This returns the following json

```

{
 "application": {
 "app_name": "User-application-in-AZURE, with overhead added",
 "date_created": "2019-11-28T17:50:33.122670Z",
 "pk_hash": "9a2564fe4d32cec895cb0f1b5ae87c0550165baf40a088e25054d4cd",
 "services": [
 {
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "8"
 }
],
 "service_type": "aze-standard_f2",
 "version_date": "2019-11-03T20:21:41.645718Z",

```



```
 "pk_hash":
 "f39943de1852e7746abfcd03dc8fec23cdc3666096b114cda98fcaf7",
 "location": "canadacentral",
 "price_type": "onDemandPrice",
 "provider": "aze"
 },
 {
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "9"
 }
],
 "service_type": "aze-standard_b8ms",
 "version_date": "2019-11-03T20:22:29.255322Z",
 "pk_hash":
 "ed44b2fc3989778c09aa8dbe171708bb4a9bbadd4afa3c4e90644881",
 "location": "eastus2",
 "price_type": "onDemandPrice",
 "provider": "aze"
 },
 {
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "17"
 }
],
 "service_type": "hr_admin_overhead",
 "version_date": "2019-11-28T17:50:23.796307Z",
 "pk_hash":
 "72456047a3bac812aa2269a352aaab0699a6fa3be61dc0331a1f8a49",
 "location": "anywhere",
 "price_type": "hr_overhead_cost",
 "provider": "overheads"
 }
],
"description": "This is an application the customer currently runs in
AZURE public cloud and has hr admin overheads"
},
"cost_components": [
 {
 "numcpus": "2",
 "memory": "4",
 "currentGen": "False",
 "quantity": "8",
 "service": "aze-standard_f2",
 "unit_cost": 0.104,
 "discount": 0,
 "unit": "",
 "location": "canadacentral",
 "service_cost": 0.83,
 "price_type": "onDemandPrice",
 "price_version_date": "2019-11-04T15:44:28.703193Z"
```

```

 },
 {
 "numcpus": "8",
 "memory": "32",
 "currentGen": "False",
 "quantity": "9",
 "service": "aze-standard_b8ms",
 "unit_cost": 0.333,
 "discount": 0,
 "unit": "",
 "location": "eastus2",
 "service_cost": 3.0,
 "price_type": "onDemandPrice",
 "price_version_date": "2019-11-03T20:22:29.258705Z"
 },
 {
 "quantity": "17",
 "service": "hr_admin_overhead",
 "unit_cost": 0.17,
 "discount": 0,
 "unit": "per server",
 "location": "anywhere",
 "service_cost": 2.89,
 "price_type": "hr_overhead_cost",
 "price_version_date": "2019-11-28T17:50:26.170275Z"
 }
],
 "total_cost": 6.720000000000001
}

```

The servicetype `hr_admin_overhead`, with quantity 17, is now a component of the total cost in the report.

We can also run the optimization or matching to another cloud provider as usual. In this case the admin cost is carried over as it is constant per server, no matter where the server is hosted.

For example, here we query porting the application to AWS.

```

curl -X GET \
 'http://127.0.0.1:8000/api/v1.0/calc/applications/app_name/User-application-in-AZURE,%20with%20overhead%20added/provider/aws/pricetype/reserved1yearPrice' \
 -H 'Authorization: JWT eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJlc2VybmFtZSI6ImJic2VydmljZWVjY291bnQiLCJleHAiOjE1NzUwNDkxNTcsImVtYWlsIjoiiIn0.PhWO2M7kcA280dCM26NjsGYGDOK4dnGFLY01KSz-SyM' \
 -H 'Content-Type: application/json' \

```

And this yields

```
{
 "application": {
 "app_name": "User-application-in-AZURE, with overhead added",
 "date_created": "2019-11-28T17:50:33.122670Z",
 "pk_hash": "9a2564fe4d32cec895cb0f1b5ae87c0550165baf40a088e25054d4cd",
 "services": [
 {
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "8"
 }
],
 "service_type": "aze-standard_f2",
 "version_date": "2019-11-03T20:21:41.645718Z",
 "pk_hash":
"f39943de1852e7746abfcd03dc8fec23cdc3666096b114cda98fcaf7",
 "location": "canadacentral",
 "price_type": "onDemandPrice",
 "provider": "aze"
 },
 {
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "9"
 }
],
 "service_type": "aze-standard_b8ms",
 "version_date": "2019-11-03T20:22:29.255322Z",
 "pk_hash":
"ed44b2fc3989778c09aa8dbe171708bb4a9bbadd4afa3c4e90644881",
 "location": "eastus2",
 "price_type": "onDemandPrice",
 "provider": "aze"
 },
 {
 "attributes": [
 {
 "attr_id": "quantity",
 "value": "17"
 }
],
 "service_type": "hr_admin_overhead",
 "version_date": "2019-11-28T17:50:23.796307Z",
 "pk_hash":
"72456047a3bac812aa2269a352aaab0699a6fa3be61dc0331a1f8a49",
 "location": "anywhere",
 "price_type": "hr_overhead_cost",
 "provider": "overheads"
 }
]
 },
}
```

```
 "description": "This is an application the customer currently runs in
AZURE public cloud and has hr admin overheads"
 },
 "current_application": {
 "components": [
 {
 "numcpus": "2",
 "memory": "4",
 "currentGen": "False",
 "quantity": "8",
 "service_type": "aze-standard_f2",
 "unit_cost": 0.104,
 "discount": 0,
 "unit": "",
 "location": "canadacentral",
 "service_cost": 0.83,
 "price_type": "onDemandPrice",
 "price_version_date": "2019-11-04T15:44:28.703193Z"
 },
 {
 "numcpus": "8",
 "memory": "32",
 "currentGen": "False",
 "quantity": "9",
 "service_type": "aze-standard_b8ms",
 "unit_cost": 0.333,
 "discount": 0,
 "unit": "",
 "location": "eastus2",
 "service_cost": 3.0,
 "price_type": "onDemandPrice",
 "price_version_date": "2019-11-03T20:22:29.258705Z"
 },
 {
 "quantity": "17",
 "service": "hr_admin_overhead",
 "unit_cost": 0.17,
 "discount": 0,
 "unit": "per server",
 "location": "anywhere",
 "service_cost": 2.89,
 "price_type": "hr_overhead_cost",
 "price_version_date": "2019-11-28T17:50:26.170275Z"
 }
],
 "total_cost": 6.720000000000001
 },
 "alternative_option": {
 "components": [
 {
 "costph": 0.0263,
 "discount": 0,
 "quantity": "8",
 "servicetype": "aws-t3a.medium",
```

```
 "price_type": "reservedyearPrice",
 "location": "ca-central-1",
 "service_cost": 0.21,
 "gpus": "0",
 "numcpus": "2",
 "memory": "4",
 "ntwPerf": "Low to Moderate",
 "instanceTypeCategory": "General purpose",
 "currentGen": "True",
 "predbogo": "N/A",
 "predcpu": "N/A",
 "tgzfilename": "N/A",
 "epoch-secs": "N/A",
 "minbogo": "N/A",
 "maxbogo": "N/A",
 "warning": "alternative for service aze-standard_f2 in
canadacentral with reservedyearPrice could not be found based on
performance, response may be unchanged or contain alternative based on
attributes instead."
 },
 {
 "costph": 0.1886,
 "discount": 0,
 "quantity": "9",
 "servicetype": "aws-t3a.2xlarge",
 "price_type": "reservedyearPrice",
 "location": "us-east-2",
 "service_cost": 1.7,
 "gpus": "0",
 "numcpus": "8",
 "memory": "32",
 "ntwPerf": "Moderate",
 "instanceTypeCategory": "General purpose",
 "currentGen": "True",
 "predbogo": "N/A",
 "predcpu": "N/A",
 "tgzfilename": "N/A",
 "epoch-secs": "N/A",
 "minbogo": "N/A",
 "maxbogo": "N/A",
 "warning": "alternative for service aze-standard_b8ms in eastus2
with reservedyearPrice could not be found based on performance, response
may be unchanged or contain alternative based on attributes instead."
 },
 {
 "costph": 0.17,
 "error": "alternative for service hr_admin_overhead in anywhere
with reservedyearPrice not found, partial data/costs returned",
 "discount": 0,
 "quantity": "17",
 "servicetype": "hr_admin_overhead",
 "price_type": "reservedyearPrice",
 "location": "anywhere",
 "service_cost": 2.89,
```

```

 "warning": "alternative for service hr_admin_overhead in anywhere
with reservedYearPrice could not be found based on performance, response
may be unchanged or contain alternative based on attributes instead."
 }
],
 "total_cost": 4.8
}
}

```

#### 1.6.4. Delete a provider (DELETE)

Custom Providers, such as the `overheads` provider we created above, can be deleted too, but only after all applications with `servicetypes` from the provider have been deleted. Otherwise the API will not allow custom provider deletion.

```

curl -X DELETE \
 http://127.0.0.1:8000/api/v1.0/calc/provider/name/ove/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiLCJlc2VybmFtZSI6ImJic2
VydmljZWZjY291bnQiLCJleHAiOiJleHAiOjE1NzUwNDkxNTcsImVtYWlsIjoiIn0.PhWO2M7kcA280dCM2
6NjsGYGDOK4dnGFLY01KSz-SyM' \
 -H 'Content-Type: application/json' \

```

This DELETE request returns `204 No Content`.

#### 1.6.5. Deleting multiple providers (POST)

Multiple providers may be deleted in one request

```

curl -X POST \
 http://127.0.0.1:8000/api/v1.0/calc/provider/delete/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJleHAiOjE1NjkyMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
 -d '[
 'ove',
 'aCustomProvider'
]'

```

Providers may be referenced by name or short name. In ambiguous cases, the same rules of precedence apply that were documented in section 1.6.2.

This DELETE request returns `204 No Content`.

#### 1.6.6. Listing all providers (GET)

The user may retrieve a list of all generic and user-owned providers like so:

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/provider/ \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6ImJic2
VydmljZWFjY291bnQiLCJleHAiOjE1NzQ5NjIyMzIsImVtYWlsIjoiiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULe' \
 -H 'Content-Type: application/json'
```

API results are paginated, specify the limit and offset of hte results like so:

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/provider/?limit=50&offset=10 \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6ImJic2
VydmljZWFjY291bnQiLCJleHAiOjE1NzQ5NjIyMzIsImVtYWlsIjoiiIn0.6ZmIczIXvers4SFRs
OowZqBtMqpD9rHwzChH_shOULe' \
 -H 'Content-Type: application/json'
```

Without specifying these parameters, the default limit is 10 and the default offset is 0.

### 1.6.6.1. Filtering and sorting providers

The user may narrow down their provider list by adding filter parameters. The following are available:

- `name` - Get only providers whose name starts with this value
- `short_name` - Get only providers whose short name starts with this value
- `userdefined` - Get only providers who belong to the user if this value is 'yes', or get providers who are generic if this value is 'no'

Furthermore, the user may specify a property to sort the results by setting the `sort` paramter to one of the following values:

- `name` - The name
- `short_name` - The short name
- `userdefined` - Whether or not the provider is user-defined (non-userdefined providers go first, then user-defined ones)

To reverse the sort order, prepend the character `-` to the value of `sort`

For example, to display all generic providers whose short name starts with 'a', sorting by full name, do

```
curl -X GET \
 http://127.0.0.1:8000/api/v1.0/calc/provider/?
short_name=a&user_defined=no&sort=name \
 -H 'Authorization: JWT'
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
-H 'Content-Type: application/json' \

```



## 1.7. Shares

Users may share the searches and reports for items in the dashboard, creating a static file that can be public or private and whose link can be shared with others. For example, list of service types filtered by location and number of CPUs.

### 1.7.1. Create a new share (POST)

Create new share like so

```
curl -X POST \
 'http://127.0.0.1:8000/api/v2.0/share/?
 limit=20&offset=0&location_contains=eastus2&min_cpus=2&max_cpus=4' \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
 -d '{
 "name": "My servicetype share",
 "description": "A list of servicetypes in eastus2 with between 2 and 4
CPU cores",
 "author_affiliation": "BigBitBus",
 "metadata": "CPU, ServiceTypes",
 "expires": "30",
 "share_type": "services",
 "public": true
 }'
```

The `share_type` field specifies what the share will contain. The following values are available:

- `applications` - Application list
- `services` - ServiceType list
- `prices` - Price list
- `providers` - Provider list
- `discounts` - Discount list
- `report` - Application report
- `vmreport` - Application optimization
- `comparison` - ServiceType comparison

For any of the 'list' share types, filter and sort specifiers can be added to the url parameters in the exact same way as is done in the list's respective api endpoint, as seen above.

Furthermore, for a type whose respective endpoint requires specifying extra values in the url itself, these can added to this endpoint's url as parameters instead. For example, to share a comparison of two service types (documented in 1.4.5.5), a user may do:

```
curl -X POST \
 'http://127.0.0.1:8000/api/v2.0/share/?
 limit=20&offset=0&provider1=alt&name1=my_misc_service_1&location1=eastus2&
 rovider2=alt&name2=my_misc_service_2&location2=eastus2' \
 -H 'Authorization: JWT
 eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
 VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
 CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
 -d '{
 "name": "My servicetype comparison",
 "description": "A comparison of two servicetypes",
 "author_affiliation": "BigBitBus",
 "metadata": "Compare, ServiceTypes",
 "expires": "30",
 "share_type": "comparison",
 "public": true
 }'
```

The new share is returned as a JSON string, which includes the URL to the share file itself

```
{
 "id": 93,
 "name": "My servicetype comparison",
 "description": "A comparison of two servicetypes",
 "author_affiliation": "BigBitBus",
 "metadata": "Compare, ServiceTypes",
 "owner": "bbserviceaccount",
 "creation_timestamp": "2020-06-02T16:41:43.689918Z",
 "expiration_timestamp": "2020-07-02T16:41:43.689918Z",
 "share_type": "services",
 "file": "https://s3.us-west-2.amazonaws.com/com.bigbitbus.shares-
 test/public/93.html",
 "public": true
}
```

And a 201 Created code is returned.

### 1.7.2. Retrieve a share (GET)

We can use the pk to get the details of a share.

```
curl -X GET \
 'http://127.0.0.1:8000/api/v2.0/share/pk/93/' \
 -H 'Authorization: JWT
 eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imdlbm
 VyaWMiLCJleHAiOjE1NjkxMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
 CpGBRNendUy2Kw' \
```

```
-H 'Content-Type: application/json' \
```

yields

```
{
 "id": 93,
 "name": "My servicetype share",
 "description": "A list of servicetypes in eastus2 with between 2 and 4
CPU cores",
 "author_affiliation": "BigBitBus",
 "metadata": "CPU, ServiceTypes",
 "owner": "bbserviceaccount",
 "creation_timestamp": "2020-06-02T16:41:43.689918Z",
 "expiration_timestamp": "2020-07-02T16:41:43.689918Z",
 "share_type": "services",
 "file": "https://s3.us-west-2.amazonaws.com/com.bigbitbus.shares-
test/public/93.html",
 "public": true
}
```

A 200 OK code is returned.

### 1.7.3. Updating the share (PUT)

the `public` and `expiration_timestamp` values can be updated for a share owned by the user. The `pk` of the share can be used as in the `get` request above.

```
curl -x put \
 'http://127.0.0.1:8000/api/v2.0/share/pk/93/' \
 -h 'authorization: jwt
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiE1NjkxMTAyOTUsImVtYWlsIjoiiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -h 'content-type: application/json' \
 -d '{
 "name": "Someone else's servicetype share",
 "description": "A list of servicetypes in eastus2 with between 2 and 4
CPU cores",
 "author_affiliation": "BigBitBus",
 "metadata": "CPU, ServiceTypes",
 "expires": "25",
 "share_type": "services",
 "public": false
}'
```

the modified share is returned

```
{
 "id": 93,
 "name": "Someone else's servicetype share",
 "description": "A list of servicetypes in eastus2 with between 2 and 4
CPU cores",
 "author_affiliation": "BigBitBus",
 "metadata": "CPU, ServiceTypes",
 "owner": "bb-serviceaccount",
 "creation_timestamp": "2020-06-02T16:41:43.689918Z",
 "expiration_timestamp": "2020-06-27T16:41:43.689918Z",
 "share_type": "services",
 "file": "https://s3.us-west-2.amazonaws.com/com.bigbitbus.shares-
test/private/93.html",
 "public": false
}
```

a 200 `ok` is returned on success.

Note that if the value of `public` is changed, then the url will also change.

#### 1.7.4. Deleting a share (DELETE)

A user can delete a share it owns. The `pk` of the share can be used as in the `put` and `get` requests above.

```
curl -X DELETE \
 'http://127.0.0.1:8000/api/v2.0/share/pk/93/' \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json'
```

A 204 `No Content` is returned.

#### 1.7.5 Deleting multiple shares (POST)

Multiple shares may be deleted in one request using the `pks` of the shares.

```
curl -X POST \
 'http://127.0.0.1:8000/api/v2.0/share/delete/' \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJlNjkwMTAyOTUsImVtYWlsIjoiIn0.GvZaozS0NQ3e3B4MrGDzNIIP32nGX
CpGBRNendUy2Kw' \
 -H 'Content-Type: application/json' \
 -d '[
 "93",
```

```
"94",
]'
```

A 204 No Content is returned.

### 1.7.6. Listing all shares (GET)

The user can retrieve a list of all shares that they own like so:

```
curl -X GET \
 'http://127.0.0.1:8000/api/v2.0/share/' \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJE1NjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQoFk' \
 -H 'Content-Type: application/json' \
```

By default, the first 10 shares are returned from the call. Specify the limit and offset of the response using pagination, like so:

```
curl -X GET \
 'http://127.0.0.1:8000/api/v2.0/share/?limit=2&offset=0' \
 -H 'Authorization: JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJlc2VybmFtZSI6Imdlbm
VyaWMiLCJleHAiOiJE1NjkyODkzMDAsImVtYWlsIjoiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bh
i5D3CpCiKQoFk' \
 -H 'Content-Type: application/json' \
```

yields

```
{
 "count": 91,
 "next": "http://127.0.0.1:8000/api/v2.0/share/?limit=2&offset=2",
 "previous": "null",
 "results": [
 {
 "id": 93,
 "name": "Someone elses servicetype share",
 "description": "A list of servicetypes in eastus2 with between
2 and 4 CPU cores",
 "author_affiliation": "BigBitBus",
 "metadata": "CPU, ServiceTypes",
 "owner": "bbserviceaccount",
 "creation_timestamp": "2020-06-02T16:41:43.689918Z",
 "expiration_timestamp": "2020-06-27T16:41:43.689918Z",
 "share_type": "services",
```

```
 "file": "https://s3.us-west-
2.amazonaws.com/com.bigbitbus.shares-test/private/93.html",
 "public": false
 },
 {
 "id": 94,
 "name": "Another servicetype share",
 "description": "A list of servicetypes in eastus2 with between
4 and 8 CPU cores",
 "author_affiliation": "BigBitBus",
 "metadata": "CPU, ServiceTypes",
 "owner": "bbserviceaccount",
 "creation_timestamp": "2020-06-02T16:43:55.813865Z",
 "expiration_timestamp": "2020-07-02T16:43:55.813865Z",
 "share_type": "services",
 "file": "https://s3.us-west-
2.amazonaws.com/com.bigbitbus.shares-test/private/94.html",
 "public": false
 }
]
```

In both cases, a 200 OK code is returned.

### 1.7.6.1. Filtering and sorting shares

The user may further specify their list by adding filter parameters. The following are available:

- `name_contains` - Get only shares whose name contains this substring
- `desc_contains` - Get only shares whose description contains this substring
- `aa_contains` - Get only shares whose `author_affiliation` field contains this substring
- `metadata_contains` - Get only shares whose metadata contains this substring
- `share_type` - Get only shares whose share type is this value
- `public` - Get only shares who are publicly shared if this value is 'true', or not publicly shared if this value is 'false'
- `min_create_time` - Get only shares who were created later than this time
- `max_create_time` - Get only shares who were created earlier than this time
- `min_expire_time` - Get only shares who will expire later than this time
- `max_expire_time` - Get only shares who will expire earlier than this time

**Note:** Times have to be specified in following format:

```
YYYY-MM-DD HH:MM[:ss[.uuuuuu]][TZ]
Some Valid DateTime Examples:
2019-12-15
2020-01-01
2019-12-12 9:55:57
```

Furthermore, the user may specify a property to sort the results by, achieved by setting the `sort` parameter to one of the following values:

- `name` - The name
- `desc` - The description
- `author_affiliation` - The `author_affiliation` field
- `metadata` - The `metadata` field
- `create_time` - The creation time
- `expire_time` - The expiration time
- `public` - Whether or not the share is public (non-public first, then public)

To reverse the sort order, prepend the character `-` to the value of `sort`

For example, display shares that were created after June 1st, 2020, who are not public, and sort them in reverse alphabetical order by name

```
curl -X GET \
 'http://127.0.0.1:8000/api/v2.0/share/?min_create_time=2019-06-01&public=false&sort=-name' \
 -H 'Authorization: JWT eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImdlbmVyaWMiLCJleHAiOiJlNjkyODkzMDAsImVtYWlsIjoiiIn0.ESU0xx9I8MVuIhvN4eD02szSzz3bhi5D3CpCiKQQoFk' \
 -H 'Content-Type: application/json' \

```

## Table of Contents

- [BigBitBus API Version 1](#)
  - [1.1. Authorization](#)
  - [1.2. Applications](#)
    - [1.2.1. Create an application](#)
    - [1.2.2. Retrieving Applications](#)
    - [1.2.3. Obtaining a report on an Application](#)
    - [1.2.4. Querying Applications](#)
    - [1.2.5. Finding Application Alternatives on Another Provider](#)
    - [1.2.6. VM Optimization](#)
    - [1.2.7. Updating an Application](#)
    - [1.2.8. Deleting an Application](#)
    - [1.2.9. Deleting multiple Applications](#)
    - [1.2.10. Listing all Applications](#)
      - [1.2.10.1. Filtering and sorting Applications](#)
  - [1.3. Prices](#)
    - [1.3.1. Creating a new price for a service \(POST\)](#)
    - [1.3.2. Get price details about a pre-existing price \(GET\)](#)
    - [1.3.3. Updating the costperunit for a pre-existing price \(PUT\)](#)
    - [1.3.4. Deleting a price \(DELETE\)](#)
    - [1.3.5. Deleting multiple prices \(POST\)](#)
    - [1.3.6. Listing all prices \(GET\)](#)
      - [1.3.6.1. Filtering and sorting prices](#)
  - [1.4. ServiceTypes](#)
    - [1.4.1. Create a new service type \(POST\)](#)
    - [1.4.2. Retrieve a service type \(GET\)](#)
    - [1.4.3. Deleting a service type \(DELETE\)](#)
    - [1.4.4. Deleting multiple service types \(POST\)](#)
    - [1.4.5. List all service types \(GET\)](#)
      - [1.4.5.1. Filtering and sorting service types](#)
    - [1.4.6. Finding service types](#)
      - [1.4.6.1. Service type keyword search](#)
      - [1.4.6.2. Service type name autocomplete](#)
      - [1.4.6.3. Getting service type info](#)
      - [1.4.6.4. Compare two service types](#)
  - [1.5. ProviderDiscounts](#)
    - [1.5.1. Create a new discount \(POST\)](#)
    - [1.5.2. Retrieve a discount \(GET\)](#)
    - [1.5.3. Updating the discount amount for a pre-existing discount \(PUT\)](#)
    - [1.5.4. Deleting a discount \(DELETE\)](#)
    - [1.5.5. Listing all discounts](#)
      - [1.5.5.1. Filtering and sorting discounts](#)
  - [1.6. Providers](#)
    - [1.6.1. Create a new provider \(POST\)](#)



- 1.6.2. Retrieve a provider (GET)
- 1.6.3. Using providers
  - 1.6.3.1. Creating a service type with the provider
  - 1.6.3.2. Creating a price with the provider
  - 1.6.3.3. Reports and Optimization
- 1.6.4. Delete a provider (DELETE)
- 1.6.5. Deleting multiple providers (POST)
- 1.6.6. Listing all providers (GET)
  - 1.6.6.1. Filtering and sorting providers
- 1.7. Shares
  - 1.7.1. Create a new share (POST)
  - 1.7.2. Retrieve a share (GET)
  - 1.7.3. Updating the share (PUT)
  - 1.7.4. Deleting a share (DELETE)
  - 1.7.5. Deleting multiple shares (POST)
  - 1.7.6. Listing all shares (GET)
    - 1.7.6.1. Filtering and sorting shares (GET)