

Integrated Billing Solutions with HP CSA 4.00



Table of Contents

Introduction	2
Part 1. HP CSA Concepts	2
Part 2. Billable Service Conditions	4
Part 3. Billable Intervals	4
Part 4. How to Collect Information from HP CSA with REST Calls	5
Solution Developers' Responsibilities	5
How to test REST Communication	5
HP CSA Certificates	7
Part 5. Sample API Sequence and Responses	7
Data Collection Sequence	7
Step 1: Get a credential token for an authenticated user	8
Step 2: Retrieve the list of organizations	8
Step 3: For each organization, get lists of subscribers and other information	9
Step 4: For each subscriber, get subscription information	10
Part 6. Utilization Properties	12
Get Utilization Information for a Subscription	13
Part 7. Group Ownership	14
How to Determine Group Ownership and Original Subscriber	14
Part 8. HP CSA 4.00 Changes and Issues	15

Introduction

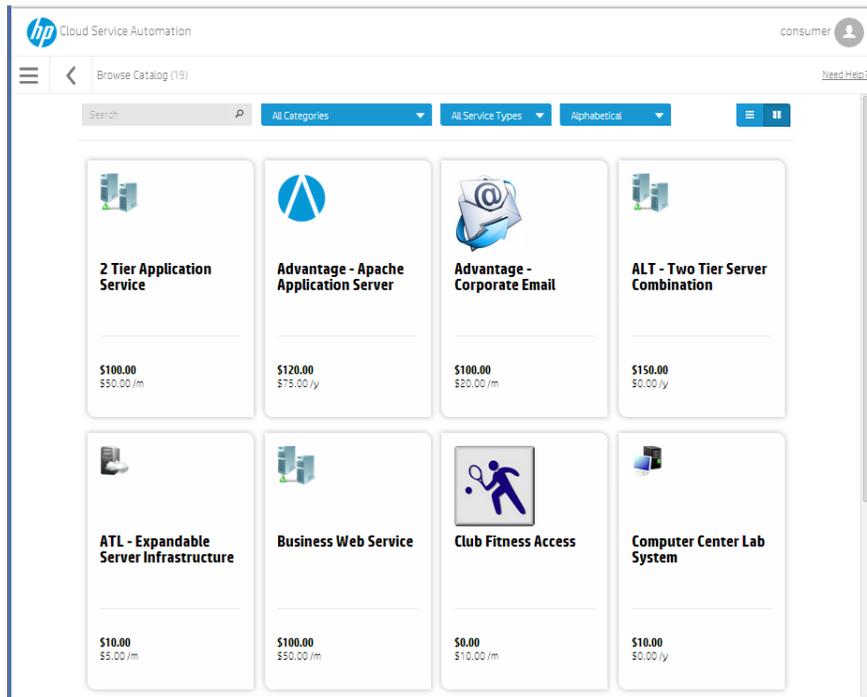
This document is intended to help solution developers create billing solutions using the subscription and resource information in HP Cloud Service Automation (HP CSA). See the *HP CSA Concepts Guide* for an overview of HP CSA concepts.

Chargeback data from HP CSA is used to support internal or external billing functions. Elements of the data can include infrastructure, application licensing, and network costs. The same information might be used for *showback* reports, allowing an IT department to analyze usage without actually cross-charging costs.

Part 1. HP CSA Concepts

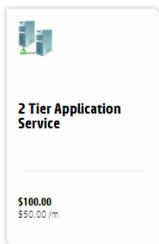
- **Services** HP CSA allows providers to define and offer catalogs of services they offer to their subscribers. These catalogs include service offerings such as basic infrastructure services, application services, or complex services with configuration management and performance monitoring.

Figure 1. Example Catalog of Services



- **Service Prices** Providers can assign prices to service offerings. The price information can include both initial (start-up) pricing and recurring costs. The prices can also be tied to the cost of subscriber-selected options, such as the number of servers or the amount of storage.

Figure 2. Base Pricing Information



- **Service Details** When a subscriber selects an item from a service catalog, they can specify detailed information for their order. This information includes:
 - A subscriber-specified name for the order. If a billing solution is to track individual subscriptions, this would be the field that could be recognized by a subscriber.

- When the service is scheduled to start and end. Billing solutions report the cost of HP CSA services from the time they are deployed until they expire or are cancelled. This is referred to as *subscription-based billing*. If no End Date is specified, the subscription will continue to run until it is cancelled.
- Group ownership. This option defines whether the subscription is to be owned (monitored, adjusted, or cancelled) by only the original subscriber, or any member of that subscriber's group.
- Subscriber-selectable options. A service provider may associate additional pricing information with service options.

Figure 3. Service Name, Description, and Start/Stop Dates

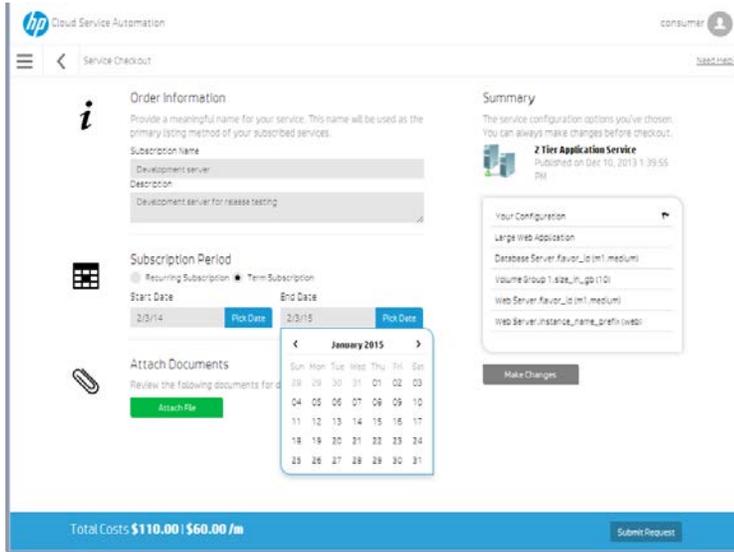
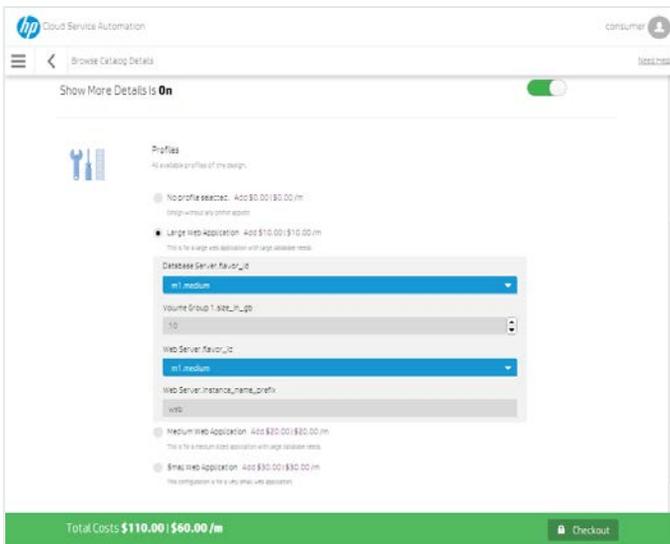


Figure 4. Group Ownership option



Figure 5. User-selectable options



- **Person** In HP CSA terminology a *person* is a subscriber to HP CSA services. This person is defined by an LDAP user, and can include other LDAP attributes such as organization and management relationships. HP CSA creates an internal representation of each person, including an ID number, membership in one or more groups, and a list of subscriptions. Billing solutions need to use this internal information to create billing records.

- **Organization** In HP CSA terminology an organization contains a set of persons. HP CSA service catalogs and pricing are assigned to specific organizations. Like HP CSA persons, organizations are defined in LDAP. For private HP CSA installations, organizations can include departments such as marketing, R&D, and payroll. For public HP CSA installations, organizations are independent customer organizations. Billing solutions also need to access the internal organization information.
- **Groups** HP CSA does not maintain any internal group relationships. Instead, it relies on directory services to identify members of groups referenced by HP CSA.

Part 2. Billable Service Conditions

Before costs can be associated with a service order, several criteria must be met:

- Any required approval steps must complete. If a request is denied explicitly or through a time-out, no service instance is created, and no records are made available.
- After a service request has been approved, a service instance is created. This service instance goes through the initializing, reserving, and deploying stages before the requested service is online. Status information on service instances is available during all deployment stages. A billing solution should only bill for a service after it has been fully deployed.

After a service reaches the deployed stage, valid billing information is made available.

Be aware that in the deployed state, the following service state changes can occur:

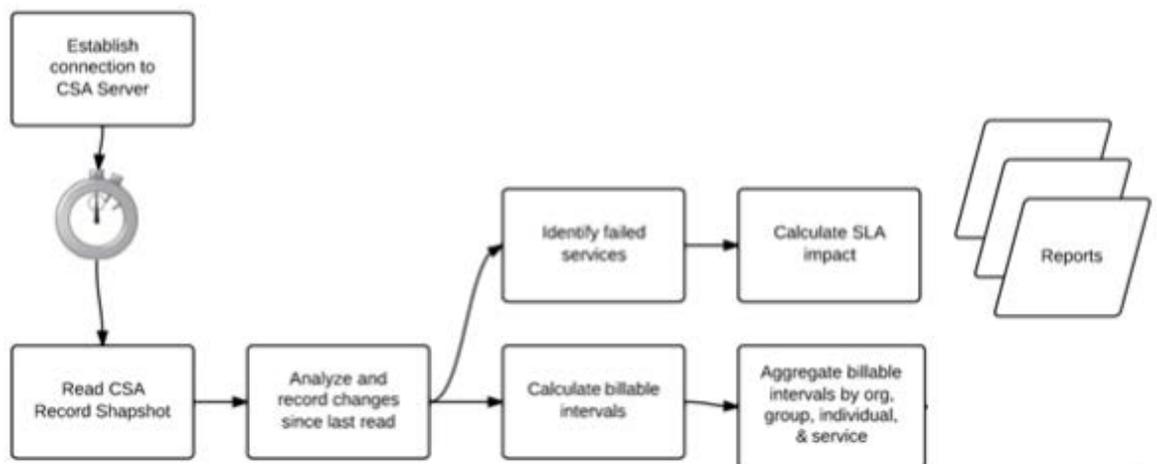
- The subscription can expire.
- The subscription can be cancelled by a subscriber or an operator.
- A subscriber may choose to “flex” (modify) the service instance by adding or deleting resources such as disk space, memory, or CPUs. These resources may have associated costs that need to be tracked and billed.

Part 3. Billable Intervals

When a HP CSA service instance reaches the Deployed state, a billing solution can collect an initial snapshot of the service instance information. The billing solution must then periodically collect subsequent snapshots of the service instance. As long as the service continues to run, a billing solution can calculate the cost of those intervals and add them to the customer billing report.

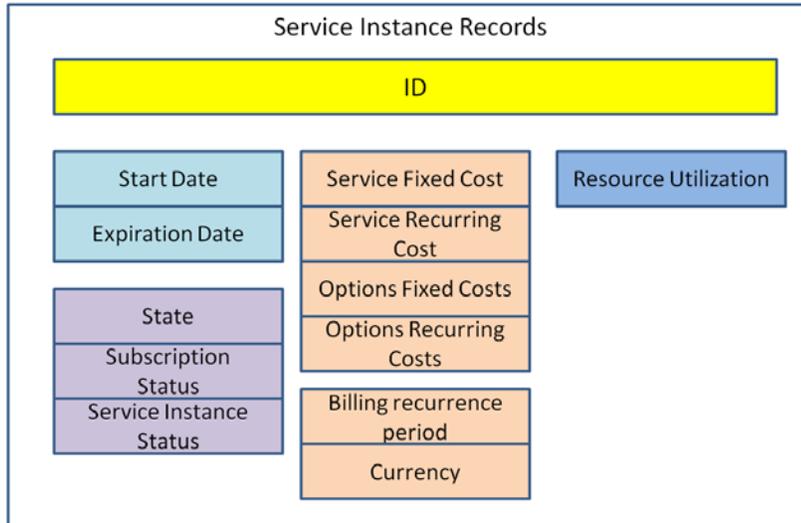
This high-level flowchart shows how service records can be used to build chargeback reports.

Figure 6. Example of how to create billable intervals



The service costs, currency, and billing interval are specified within the service instance. Figure 7 shows many of the billing-related fields that are contained in a service instance. Part 4 of this document describes how to collect this information with API calls.

Figure 7. Information in a Service Instance



If a service expires, is cancelled, or fails, the billing solution should stop accruing charges against that service.

Part 4. How to Collect Information from HP CSA with REST Calls

HP CSA exposes usage and billing information in the service instances through a set of REST (**RE**presentational **State Transfer**) APIs. Data can be returned in either XML or JSON format. The format of the returned data is specified by the user making the REST API call in the Accept Content-Type header setting. The examples in this paper use the XML format.

Solution Developers' Responsibilities

Solution developers communicate with HP CSA over http (or https) and parse the data structures returned by HP CSA. The default https port for communication with HP CSA is port 8444. A developer can modify this port as needed in the HP CSA JBoss configuration.

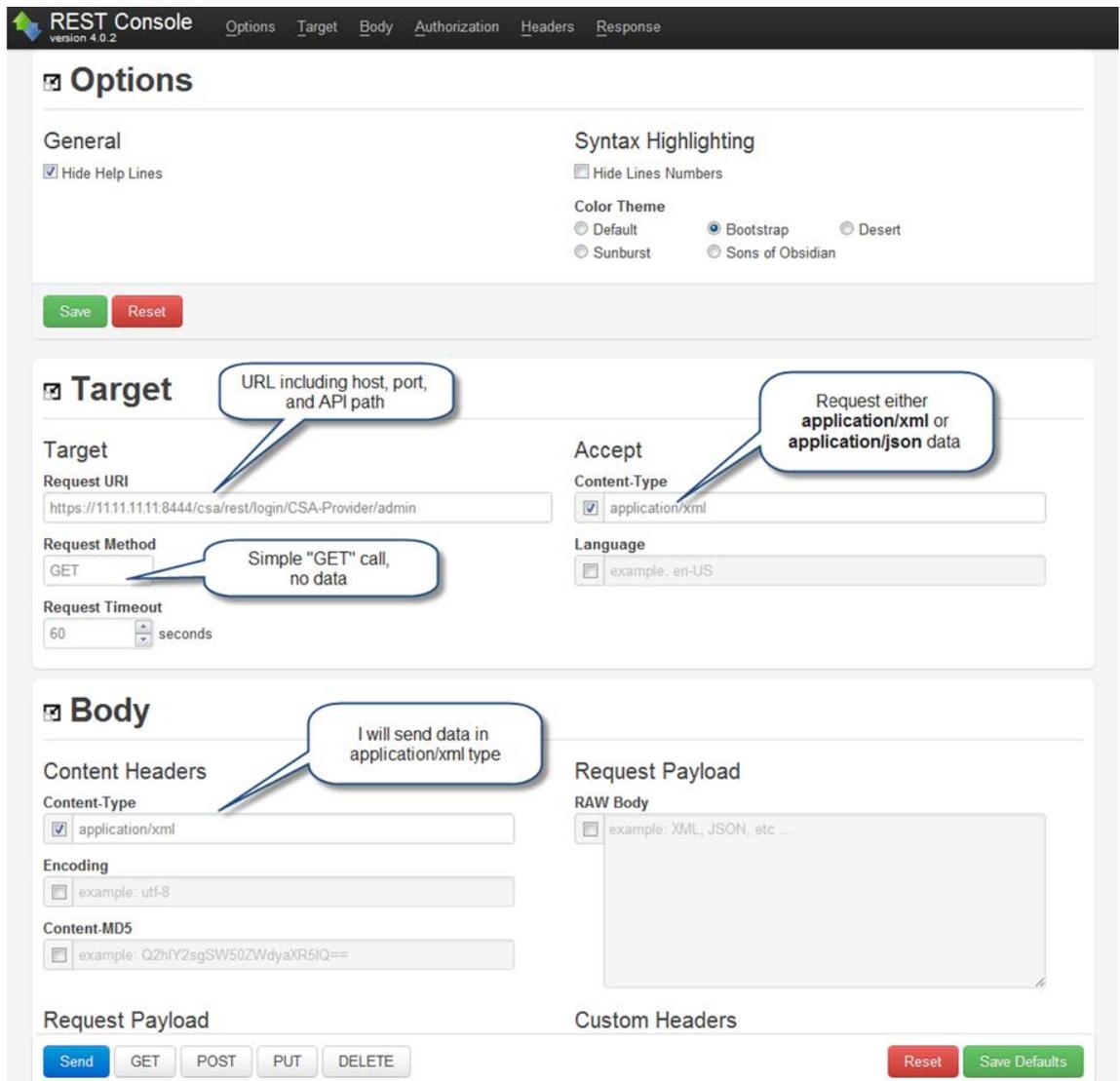
The developer must configure authentication (also called authorization) and headers as needed. Common headers include `application-Type: application/xml` and `accept: application/xml`.

How to test REST Communication

A convenient way to test communication with HP CSA is to use a REST Console such as those provided in Google™ Chrome and other browsers. You can add this Google Chrome plug-in as follows:

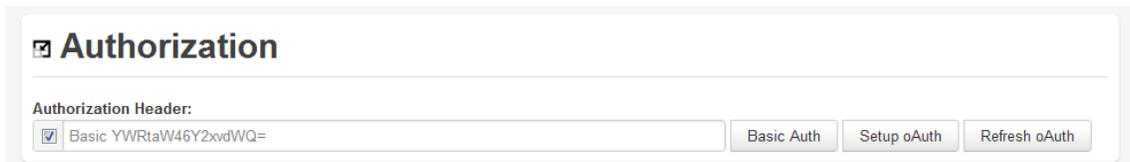
- Start Google Chrome.
- Open **Chrome Web Store**.
- Use search box to search for **REST Console**.
- Select **Add to Chrome**.

Figure 8. Example Configuration of the REST Console Plug-in



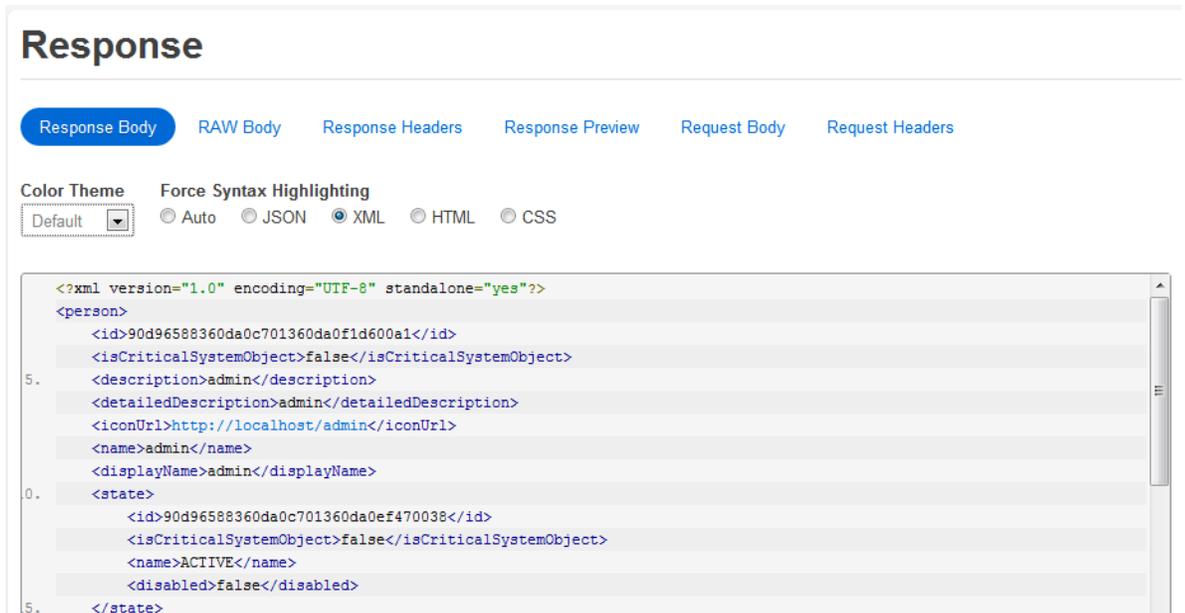
Configure the *basic* authorization in the Authorization tab using a username and password for administrative user or other user with appropriate permissions on your HP CSA 4.00 system.

Figure 9. Sample authorization



If the plug-in is configured correctly, when you place a request you will see output in the Response tab that looks similar to this:

Figure 10. Example Output in the Response Tab



HP CSA Certificates

When a user logs into HP CSA through a web browser, HP CSA presents a certificate to “prove” to the user that it is the HP CSA site that it claims to be. By default, HP CSA does not offer a certificate that has been validated by a trusted authority. At this point, the user has the choice of accepting the offered certificate or ending the session.

Figure 11. Example Security Certificate Message



The site's security certificate is not trusted!

You attempted to reach **16.78.67.34**, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Google Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications.

You should not proceed, **especially** if you have never seen this warning before for this site.

▶ [Help me understand](#)

Your API software will need to know how to handle the certificate “handshake” situation and how to proceed appropriately—perhaps by tracking trusted sites and certificates.

Part 5. Sample API Sequence and Responses

Data Collection Sequence

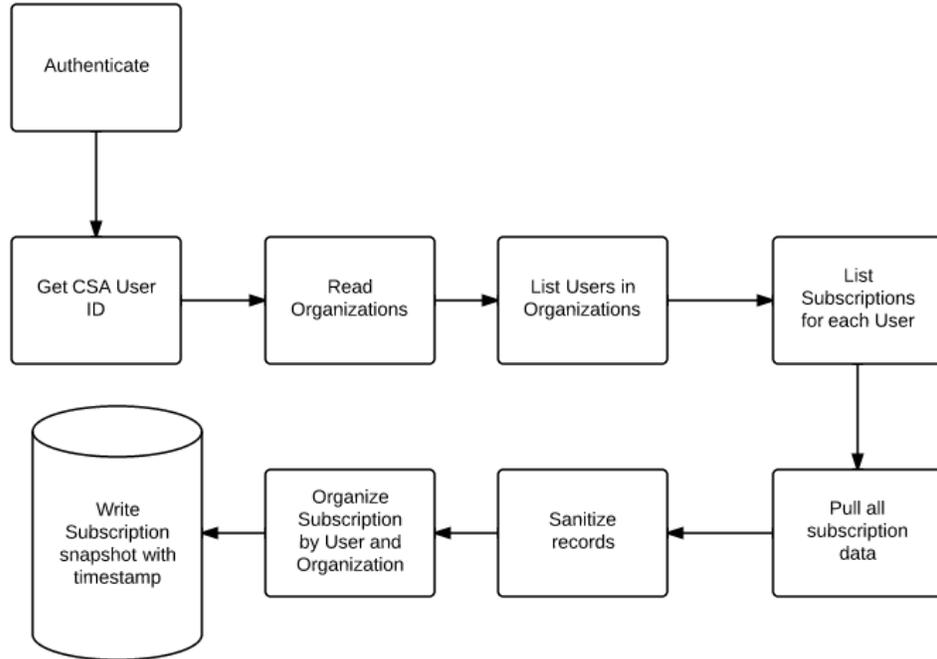
Figure 12 describes one possible sequence for HP CSA billing information data collection. Developers can choose to execute queries of different breadth or depth based on reporting requirements. For the details of specific REST API calls and options, see the *HP Cloud Service Automation API Reference*.

Example sequence:

1. Get a credential token for administrative user or other user with appropriate permissions.
2. Retrieve the list of organizations configured in this HP CSA installation.
3. For each organization, get a list of subscribers as well as other information the billing solution developer might be interested in.

- For each subscriber, retrieve information on their subscriptions including pricing, options, start and stop dates, and more.

Figure 12. Sample REST API sequence



In the following steps, please note that XML responses will be abbreviated in most examples, highlighting the content of most interest for that example. In practice, you will receive substantially more XML content returned by the REST.

Step 1: Get a credential token for an authenticated user

REST Syntax:

```
GET http://<host>:<port>/csa/rest/login/<organization_name>/<user_name>
```

Where organization_name and user_name are the names configured for the organization and user in LDAP.

Response Format:

The first field of the response is the user’s ID. We’ll refer to this as *adminId* since we will be working with individual subscriber IDs shortly. Save adminId; you’ll use it in subsequent calls.

```
<person>
  <id>90d96588360da0c701360da0f1d600a1</id>
  ...
  <organization>
    <id>402895e336d17e380136d17e7b470001</id>
    ...
  </organization>
</person>
```

Step 2: Retrieve the list of organizations

REST Syntax:

```
GET http://<host>:<port>/csa/rest/organization?userIdentifier=<adminId>
```

Response Format:

```
<OrganizationList>
  <count>n</count>
  <limit>0</limit>
  <organization>...</organization>
  <organization>...</organization>
  ...
</OrganizationList>
```

Details of the Organization Element:

```
<organization>
  <id><organization ID></id>
  <description><organization description></description>
  <name>...</name>
  <displayName><organization name for displaying></displayName>
  <partyType>...</partyType>
  <accountNumber>...<accountNumber>
  <businessRole>...</businessRole>
</organization>
```

Save the organization ID elements for subsequent operations. You might also want related fields such as the *description* and *display name* for each organization.

Step 3: For each organization, get lists of subscribers and other information

REST Syntax:

```
GET https://<host>:<port>/csa/rest/organization/<organization_id>?scope=
baseplusone&detail=basic&userIdentifier=<adminId>
```

Response Format:

The response includes a number of elements you might be interested in.

- Organization information
- Catalogs made available to the organization by the administrator
- Account number for the organization
- Groups that have been created for the organization by the administrator
- Persons belonging to the organization

```
<Organization>
  <id><organization ID included in call></id>
  <description><organization description></description>
  <detailedDescription><detailed organization description></detailedDescription>
  <name><internal organization name></name>
  <displayName><organization name for displaying></displayName>
  ...
  <catalog>
    <id><catalog ID></id>
    ...
    <name><internal catalog name></name>
    <displayName><catalog name for displaying></displayName>
  </catalog>
  ...
  <accountNumber>
    <organization account number>
  </accountNumber>
  ...
  <person>
    <id><subscriber ID></id>
    ...
    <name><internal subscriber name></name>
    <displayName><subscriber display name></displayName>
  </person>
```

```

...
<group>
  <id><group ID></id>
  ...
  <name><internal group name></name>
  <displayName><group name for displaying></displayName>
</group>
</Organization>

```

Save each subscriber's ID (for other API calls) and subscribers' names. You can keep the catalog information or collect it on a subscriber-by-subscriber basis (Step 4).

Step 4: For each subscriber, get subscription information

Use the subscriber names (from Step 3) to get the subscription details for each subscriber.

There are many fields of interest returned by this call. Some of the ServiceSubscription fields related to billing contain information such as the following:

- **Subscription ID** Use this ID if you need to periodically examine the state of a single subscription.
- **Subscription Description, Name, Display Name, Catalog, Category Display Name** Use the display name and description if you need to present the right labels and information to a customer.
- **IdOfThisServiceInstance** This field is not directly used in billing, but HP CSA personnel can use it to diagnose problems that might have occurred in a specific service instance.
- **SubscriptionStatus, ServiceInstanceState** Use to identify the status of a subscription or service instance. If a service has failed, it might not make sense to continue to bill for it.
- **Subscription StartDate, StopDate, RecurrencePeriod** Use to decide when to start, stop, and provide incremental billing.
- **FixedPrice, UnitPrice, Currency, PriceMultiplier** Use to see price-related information for a subscription.

Note that there might be two *totalPrice* entries in the response: one for the initial establishment of the service which represents the sum of all SETUP costs, and one which represents the sum of all RECURRING costs incurred at the interval specified by the recurrencePeriod. These sums represent the base cost of the service offering to create this subscription plus any additional cost associated with each of the selected service offering's options. You can identify which totalPrice you're viewing by the xsi type or the totalPrice.priceCategory.name field.

One special field of the subscription data relates to group ownership. This is discussed in Part 7.

REST Syntax:

```
GET https://<host>:<port>/csa/rest/user/mysubscription?userIdentifier=<adminID>&
requestor=<user_name>&scope=view &creationStartDate=<date>&creationEndDate=<date>
```

Where *<user_name>* is the user whose subscription information will be returned.

Response Format (a list of service subscriptions, each with the following content):

```

<ServiceSubscription>
  <id>
    90cec3a03a93ef89013a93faa4cd0040
  </id>
  ...
  <description>
    <service description>
  </description>
  ...
  <name>
    <internal subscription name>
  </name>
  <displayName>
    <subscription name for displaying>
  </displayName>
  <catalogItem>
    ...
  </catalogItem>
  <state>

```

```

...
<name>
  <state of the subscription, for example, ACTIVE>
</name>
<displayName>
  <state name for displaying>
</displayName>
...
</state>
<serviceInstance>
  <id>
    90cec3a03a93ef89013a93faaa2b00a8
  </id>
  <disabled>
    false
  </disabled>
  <serviceInstanceState>
    <name>
      <state of this service instance, for example: CANCELLED>
    </name>
    <displayName>
      <state for displaying, for example, Cancelled>
    </displayName>
  </serviceInstanceState>
</serviceInstance>
<subscriptionStatus>
  <name>
    <status of the subscription, for example, ACTIVE or EXPIRED>
  </name>
  <displayName>
    <status for displaying>
  </displayName>
  ...
</subscriptionStatus>
<initiatingServiceRequest>
  SEE PART 7
</initiatingServiceRequest>
<startDate>
  <start date and time for the subscription, for example: 2012-10-
24T12:12:00-06:00>
</startDate>
<endDate>
  <end date and time for the subscription>
</endDate>
<submittedOn>
  ...
</submittedOn>
<totalPrice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="recurrentPricingVO">
  <priceCategory>
    ...
    <name>
      RECURRING
    </name>
    <displayName>
      ...
    </displayName>
  </priceCategory>
  <fixedPrice>
    <Price with decimal; remember that there's also a currency type.
Example: 150.0>
  </fixedPrice>
  <unitPrice>
    <Price per unit with decimal, example: 0.0>
  </unitPrice>

```

```

        </totalPrice>
        <totalPrice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="initialPricingVO">
            <priceCategory>
                ...
                <name>
                    SETUP
                </name>
                <displayName>
                    ...
                </displayName>
            </priceCategory>
            <fixedPrice>1200.0</fixedPrice>
            <unitPrice>0.0</unitPrice>
        </totalPrice>
        # pricingModel contains the currency and billing period.
        <pricingModel>
            ...
            <currency>
                ...
                <name>
                    <currency name, for example: USD>
                </name>
                <displayName>
                    <currency name for displaying>
                </displayName>
            </currency>
            <priceAdjustmentMultiplier>1.0</priceAdjustmentMultiplier>
            <recurrencePeriod>
                ...
                <name>
                    <recurrence period name, for example: MONTH>
                    # Something to consider: will the monthly cost be the
                    # same in February as in July though the months are
                    # different in length?
                </name>
                <displayName>
                    <recurrence period name for display, for example: Monthly>
                </displayName>
            </recurrencePeriod>
        </pricingModel>
    </ServiceSubscription>

```

Part 6. Utilization Properties

During the process of service design, certain parameters such as memory, CPU, storage, and power may be classified as “Utilization Properties.” The primary purpose of these properties is to select appropriate resource offerings and deduct those resources from the resource pools. Most of the fields returned by the API are related to the resource binding process; but the resourceType, unit, and usage fields might be used in billing reports.

Notes:

- “Utilization” specifies resources needed to start a process. This is not always the same as “usage,” which is the metered consumption of resources.
- There is no direct way to associate Utilization with pricing information. Pricing for reservation-based usage must be specified in the service and option pricing screens.

Get Utilization Information for a Subscription

REST Syntax:

GET http://<host>:<port>/csa/rest/utilization/<subscriptionId>&userIdentifier=<adminId>

Response Format (a list utilization resource):

```
<UtilizationList>
  <utilization>
    <resourceBindingId>...</resourceBindingId>
    <resourcePool>...</resourcePool>
    <resourceProvider>...</resourceProvider>
    <resourceType>
      <id>...</id>
      <isCriticalSystemObject>...</isCriticalSystemObject>
      <description>Power capacity as retrieved from a power
        management/budget interface</description>
      <name>POWER</name>
      <displayName>Power</displayName>
      <disabled>...</disabled>
    <serviceComponentId>...</serviceComponentId>

    <serviceInstanceId>
      <unit>
        <id>...</id>
        <isCriticalSystemObject>...</isCriticalSystemObject>
        <description>Kilowatts</id> THESE ARE SPECIAL UTILIZATION PROPERTIES
        <name>KW</name> AND CANNOT BE DEFINED BY DESIGNER.
        <displayName>KW</displayName> MUST BE INTEGER.
      </unit>

      <usage>
        123
      </usage>
    </serviceInstanceId>
  </utilization>...</utilization>
</UtilizationList>
```

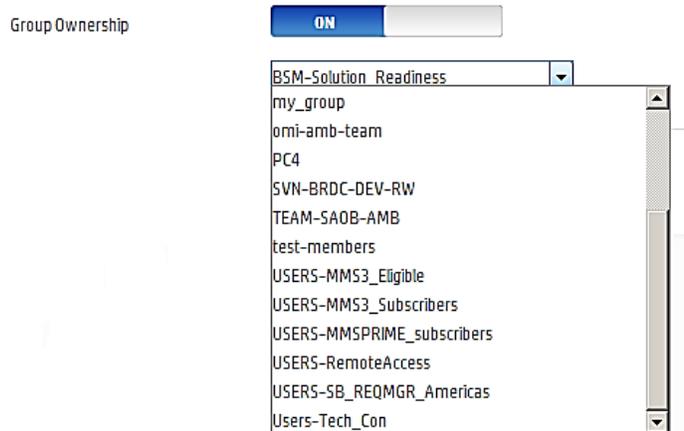
Defined Utilization Properties

The following Utilization Properties are defined in HP CSA.

- CPU (Number of CPUs)
- IPv4 Addresses (Number of Addresses)
- IPv6 Address (Number of Addresses)
- License (Number of Keys)
- Memory (MB)
- Physical Server (Number of Servers)
- Power (KW)
- Storage (GB)
- Subnet (Number of Subnets)
- VLAN (Number of VLANs)
- Virtual Server (Number of Servers)

Part 7. Group Ownership

Figure 13. Group Ownership in HP CSA Marketplace Portal



When ordering a service in HP CSA 4.00, a subscriber can specify that it is to be co-owned by other subscribers. For some customers, this is an important feature, but it requires some additional processing for the chargeback solution. Here are some points to keep in mind:

- The subscription will be listed for *every* subscriber in the specified group. This means that the billing solution can never assume that a subscription and subscription costs are unique to a single person. The subscription ID will be the same in each user’s list of subscriptions.
- A subscription can be assigned to a group that may include non-HP CSA users. Any member of any LDAP group shared with the original subscriber is a potential “owner.”
- If a subscription is assigned to a group, the only valid members of that group are people who have actually logged in to HP CSA. This is because HP CSA relies on LDAP to maintain Group/Subscriber relationships. After a user’s initial login, the user/LDAP information is cached. At that point, an integrated solution can list user and subscription information. This implies that the integrated solution needs to periodically check the list of valid HP CSA subscribers.

How to Determine Group Ownership and Original Subscriber

In Step 4, we retrieved subscription information such as the name, start date, and price information. There are a few additional fields that will be useful in understanding group ownership.

The `<ownerGroup>` field will appear in a subscription record if it is shared, and will contain the name of the selected LDAP group. If you choose to bill based on a group account, this is the field you need.

If group ownership has not been specified, the field will not exist.

An example of this field would be:

```
<ownerGroup>Test team 1</ownerGroup>
```

If you want to bill for a subscription based on the original requester, this will take a few more steps.

Find the ID of the `<catalog>` that was used to generate the subscription request. In the subscription details, look for the `/ServiceSubscription/catalogItem/catalog/id` entry. An example of this would be:

```
<ServiceSubscription>
  <id>
    90cec3a03a93ef89013a93faa4cd0040
  </id>
  ...
  <catalogItem>
    <catalog>
      <id>catalogId</id>
      ...
    </catalog>
  </catalogItem>
</ServiceSubscription>
```

Save the *catalogId* value.

Next, get the ID of the <initiatingServiceRequest>. This includes information from the original request submission. Look for the value of /ServiceSubscription/initiatingRequest/ID. An example of this would be:

```
<ServiceSubscription>
  <id>
    90cec3a03a93ef89013a93faa4cd0040
  </id>
  ...
  <initiatingServiceRequest>
    <id>
      <id>initiatingRequestId</id>
```

Now use the *catalogId* and *initiatingRequestId* with the /csa/rest/catalog call to get the desired subscriber information.

REST Syntax:

```
GET http://<host>:<port>/csa/rest/catalog/<catalogId>/request/
<initiatingRequestId>&userIdentifier=<subscriberId>
```

Response Format

```
<ServiceRequest>
  ...
  <createdBy>
    <id>subscriberId</id>
    ...
    <name>subscriberName</name>
    <displayName>subscriberDisplayName</displayName>
    ...
  </createdBy>
  ...
</ServiceRequest>
```

Part 8. HP CSA 4.00 Changes and Issues

- The lifecycleExecutionStatus field may or may not exist in a given record. If the requested service was based on a Sequenced (lifecycle actions & Operations Orchestration) design, then the field will exist. If the service was based on a Topology design, the field will not exist. You may need to update your parser to handle this condition.
- HP CSA 4.00 introduces a new method for designing services. The *Topology Designer* uses a “prescriptive” model for specifying designs, and the HP CSA 4.00 Topology plug-in service works with HP Cloud OS and OpenStack to realize the subscription request. You may wish to collect data describing which subscriptions are based on Sequenced designs, and which are based on Topology designs. In HP CSA subscription records, this is known as an “experience type”. The syntax to get the experience type is:

```
/csa/rest/catalog/%s/instance/%s?userIdentifier=%s
```

To make this call, you’re going to need both the catalog ID and the instance ID. The location of these fields is shown above in Step 4. The response will be in the path /ServiceInstance/serviceDesignerExperienceType, and will be either SEQUENCE_BASED or TOPOLOGY_BASED.

- Beware of very large XML responses. There is a defect in the initial release of HP CSA 4.00: XML responses may include user-specified attachments and service offering attachments. These can be large text documents, diagrams, and zip files. The HP CSA 4.01 patch removes these unwanted attachments, and XML size and performance return to normal. Be sure to install this patch!

For more information

To access other toolkits to design and extend services running on HP CloudSystem, go to hp.com/go/csdevelopers.

For more information on HP CloudSystem, visit hp.com/go/cloudsystem.

HP software product manuals and documentation for the following products can be found at <http://h20230.www2.hp.com/selfsolve/manuals>. You will need an HP Passport to sign in and gain access.

- HP Cloud Service Automation
- HP ArcSight
- HP Operations Orchestration
- HP Server Automation
- HP SiteScope
- HP Universal CMDB

To help us improve our documents, please send feedback to CSAdocs@hp.com.

Learn more at
hp.com/go/csa

Sign up for updates
hp.com/go/getupdated



Share with colleagues



Rate this document

© Copyright 2013-2014 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

February 2014

