



# Requests, Responses, and Schema

- [API Requests and Responses, on page 1](#)
- [Schema Object Definitions, on page 88](#)

## API Requests and Responses



**Note** Elements defined as mandatory ([1..x]) in CPS response schema is applicable only to success response and not to failure response.

### AddCredentialRequest

This API adds a credential to an existing subscriber.

#### Schema

```
<AddCredentialRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <credential> CredentialType </credential> [1]
</AddCredentialRequest>
```

#### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddCredentialRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <credential>
        <networkId>newUniqueIdentifier</networkId>
        <password>password</password>
        <type>type</type>
        <description>my Credential</description>
      </credential>
    </AddCredentialRequest>
```

```

    </se:Body>
  </se:Envelope>

```

## AddCredentialResponse

### Schema

```

<AddCredentialResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</AddCredentialResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddCredentialResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </AddCredentialResponse>
  </se:Body>
</se:Envelope>

```

## AddCredentialsRequest

This API adds multiple credentials to an existing subscriber.

### Schema

```

<AddCredentialsRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <credential> CredentialType </credential> [1..5]
</AddCredentialsRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddCredentialsRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <credential>
        <networkId>newUniqueIdentifier</networkId>
        <password>password</password>
      </credential>
      <credential>
        <networkId>anotherUniqueIdentifier</networkId>
      </credential>
    </AddCredentialsRequest>
  </se:Body>
</se:Envelope>

```

## AddCredentialsResponse

### Schema

```
<AddCredentialsResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</AddCredentialsResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddCredentialsResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </AddCredentialsResponse>
  </se:Body>
</se:Envelope>
```

## AddServiceRequest

This API adds service definitions and corresponding Balance entries for a given subscriber.

The new Balances follow existing Balance rules - meaning if new, then provision, but if the Balance already exists, then just add a new credit.

If the service code already exists for the Subscriber, the service is not added or changed.




---

### Note Not a Replace

If the service already exists for the Subscriber, the service is not updated or changed in any way.

---




---

### Note Automatic Balance Provisioning

This feature allows Service Options to be configured so that when the Service is applied to the Subscriber, Balance is automatically provisioned. (It also removes Balance when a Service is removed.) If the system is setup for auto-provisioning, do not include the Balance object when using this API. That is, only send in the Balance object when the system is not setup for auto-provisioning.

---

### Schema

```
<AddServiceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <service> ServiceType </service> [1..10]
  <balance> CreateBalanceType </balance> [0..10]
</AddServiceRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddServiceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
```

```

<service>
  <code>SERVICE_CODE</code>
  <enabled>true</enabled>
  <avp>
    <code>AVP_CODE</code>
    <value>AVP_VALUE</value>
  </avp>
  <schedule>
    <startDate>2011-01-01T00:00:00Z</startDate>
    <endDate>2012-01-01T00:00:00Z</endDate>
    <state>ON</state>
    <startTime>00:00</startTime>
    <endTime>23:59</endTime>
    <repeat>
      <dayOfMonth>*</dayOfMonth>
      <month>*</month>
      <dayOfWeek>?</dayOfWeek>
      <year>*</year>
    </repeat>
    <enabled>true</enabled>
  </schedule>
</service>
<balance>
  <code>DATA</code>
  <quotaCode>RECURRING</quotaCode>
  <startDate>2011-01-01T00:00:00Z</startDate>
  <expirationDate>2012-01-01T00:00:00Z</expirationDate>
  <initialAmount>500</initialAmount>
</balance>
</AddServiceRequest>
</se:Body>
</se:Envelope>

```

## AddServiceResponse

### Schema

```

<AddServiceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</AddServiceResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddServiceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </AddServiceResponse>
  </se:Body>
</se:Envelope>

```

## AddSubscriberSsidsRequest

This API adds SSID details to a subscriber in SPR SSID extension collection.

### Schema

```

<AddSubscriberSsidsRequest>
  <audit> AuditType </audit> [0..1] ?

```

```

    <networkId> xsd:string </networkId> [1] ?
    <subscriberSsid> SsidType </subscriberSsid> [1..40]
  </AddSubscriberSsidsRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddSubscriberSsidsRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>subscriber credential used for provisioning the subscriber to SPR</networkId>

      <subscriberSsid>
        <ssidKey>UniqueIdentifier</ssidKey>
        <ssid>ssid</ssid>
        <accessType>accessType</accessType>
        <authType>Authentication Type</authType>
        <username>Username</username>
        <password>Password</password>
        <loginUrl>URL for login</loginUrl>
        <configUrl>details</configUrl>
        <verificationCertUrl>details</verificationCertUrl>
        <configMessage>details</configMessage>
        <portalFailMessage>details</portalFailMessage>
        <unmanagedVpnConnectPrompt>details</unmanagedVpnConnectPrompt>
        <unmanagedVpnDisconnectPrompt>details</unmanagedVpnDisconnectPrompt>
      </subscriberSsid>
      <subscriberSsid>
        <ssidKey>UniqueIdentifier</ssidKey>
        <ssid>ssid</ssid>
        ...
        <unmanagedVpnDisconnectPrompt>details</unmanagedVpnDisconnectPrompt>
      </subscriberSsid>
    </AddSubscriberSsidsRequest>
  </se:Body>
</se:Envelope>

```

## AddSubscriberSsidsResponse

### Schema

```

<AddSubscriberSsidsResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</AddSubscriberSsidsResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AddSubscriberSsidsResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </AddSubscriberSsidsResponse>
  </se:Body>
</se:Envelope>

```

## AuditRequest

This API adds an entry to the Audit History. See [Audit History](#) and [AuditType](#), on page 88 for more information.



### Note AuditRequests

As a consequence of the XSS defense changes to the API standard operation, any XML data sent in an AuditRequest must be properly escaped even if inside CDATA tags. For example, `&lt;ExampleRequest&gt;...&lt;/ExampleRequest&gt;`; See [Audit History](#) for more information.

### Schema

```
<AuditRequest>
  <audit> AuditType </audit> [0..1] ?
</AuditRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AuditRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <id>csrusername</id>
      <comment>comment</comment>
      <timestamp>2011-01-01T00:00:00Z</timestamp>
      <request>ExampleRequest</request>
      <dataid>subscriber@gmail.com</dataid>
      <data><![CDATA[<ExampleRequest>...</ExampleRequest>]]></data>
    </AuditRequest>
  </se:Body>
</se:Envelope>
```

## AuditResponse

### Schema

```
<AuditResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</AuditResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AuditResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </AuditResponse>
  </se:Body>
</se:Envelope>
```

## AuthenticateSubscriberRequest

This API validates a subscriber against the USuM database and returns a subscriber.

### Schema

```
<AuthenticateSubscriberRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <password> xsd:string </password> [0..1]
</AuthenticateSubscriberRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AuthenticateSubscriberRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <password>password</password>
    </AuthenticateSubscriberRequest>
  </se:Body>
</se:Envelope>
```

## AuthenticateSubscriberResponse

### Schema

```
<AuthenticateSubscriberResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <subscriber> SubscriberType </subscriber> [0..1]
</AuthenticateSubscriberResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <AuthenticateSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <subscriber>
        ...
      </subscriber>
    </AuthenticateSubscriberResponse>
  </se:Body>
</se:Envelope>
```

## ChangeBalanceSubscriberIdRequest

This API alters the chargingId value for a subscriber and the MsBM SubscriberId. This API replaces the old MsBM API ChangeSubscriberAccountId.

The chargingId (subscriber.getBillingInfo().getChargingId()) is used as a correlation id value between Balance and USuM. The default correlation is the USuM id (generated by the database). If a chargingId is set on the subscriber, the chargingId value becomes the correlation id.

This API must be used to add an existing subscriber to a shared Balance account. Set the shared flag == true.

When removing a subscriber from a shared Balance account, set the shared flag == true. The newBalanceSubscriberId value is used to create a new chargingId for the subscriber, and the new Balance

account will have no Balance. If <balance/> elements are included in the request, then the new account will be provisioned in the normal fashion. Because the new Balance account has no Balance in the shared scenario, the <extendCredit/> elements are not applicable.




---

**Note Update Scope**

Please note that the UpdateSubscriber API does not/cannot change the chargingId/Balance SubscriberId relationship. To make such a change you must use ChangeBalanceSubscriberId.

---




---

**Note Old Account is Deleted!**

When adding a subscriber to a shared Balance, the Balance record for the old account is permanently deleted from the database and all quota is lost, unless the quota is migrated manually using the ExtendCredit or CreateBalance elements.

---

**Schema**

```
<ChangeBalanceSubscriberIdRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <newBalanceSubscriberId> xsd:string </newBalanceSubscriberId> [1]
  <shared> xsd:boolean (default = false) </shared> [0..1]
  <removeOriginalAccount> xsd:boolean (default = false) </removeOriginalAccount> [0..1] ?
  <balance> CreateBalanceType </balance> [0..10]
  <extendCredit> ExtendCreditType </extendCredit> [0..10]
</ChangeBalanceSubscriberIdRequest>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeBalanceSubscriberIdRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <newBalanceSubscriberId>chargingId</newBalanceSubscriberId>
    </ChangeBalanceSubscriberIdRequest>
  </se:Body>
</se:Envelope>
```

## ChangeBalanceSubscriberIdResponse

**Schema**

```
<ChangeBalanceSubscriberIdResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <returnCredit> ReturnCreditType </returnCredit> [0..10]
</ChangeBalanceSubscriberIdResponse>
```

**Example**



```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeBalanceSubscriberIdResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ChangeBalanceSubscriberIdResponse>
  </se:Body>
</se:Envelope>

```

## ChangeBillCycleRequest

This API changes the bill cycle day for the Balance account for a given subscriber.

### Schema

```

<ChangeBillCycleRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <accountBalanceCode> xsd:string </accountBalanceCode> [1]
  <recurringQuotaCode> xsd:string </recurringQuotaCode> [1]
  <newBillCycleDay> xsd:integer (1 >= value <= 31) </newBillCycleDay> [1] ?
</ChangeBillCycleRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeBillCycleRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <newBillCycleDay>15</newBillCycleDay>
      <accountBalanceCode>DATA</accountBalanceCode>
      <recurringQuotaCode>RECURRING</recurringQuotaCode>
    </ChangeBillCycleRequest>
  </se:Body>
</se:Envelope>

```

## ChangeBillCycleResponse

### Schema

```

<ChangeBillCycleResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</ChangeBillCycleResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeBillCycleResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ChangeBillCycleResponse>
  </se:Body>
</se:Envelope>

```

## ChangeCredentialPasswordRequest

This API changes the password for a subscriber's credential.

### Schema

```
<ChangeCredentialPasswordRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <oldPassword> xsd:string </oldPassword> [1]
  <newPassword> xsd:string </newPassword> [1]
</ChangeCredentialPasswordRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeCredentialPasswordRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <oldPassword>password</oldPassword>
      <newPassword>newpassword</newPassword>
    </ChangeCredentialPasswordRequest>
  </se:Body>
</se:Envelope>
```

## ChangeCredentialPasswordResponse

### Schema

```
<ChangeCredentialPasswordResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</ChangeCredentialPasswordResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeCredentialPasswordResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ChangeCredentialPasswordResponse>
  </se:Body>
</se:Envelope>
```

## ChangeCredentialUsernameRequest

This API changes the networkId (username) for a subscriber's credential.

### Schema

```
<ChangeCredentialUsernameRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <oldNetworkId> xsd:string </oldNetworkId> [1]
```

```
</ChangeCredentialUsernameRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeCredentialUsernameRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>newUniqueIdentifier</networkId>
      <oldNetworkId>uniqueIdentifier</oldNetworkId>
    </ChangeCredentialUsernameRequest>
  </se:Body>
</se:Envelope>
```

## ChangeCredentialUsernameResponse

### Schema

```
<ChangeCredentialUsernameResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</ChangeCredentialUsernameResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeCredentialUsernameResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ChangeCredentialUsernameResponse>
  </se:Body>
</se:Envelope>
```

## ChangeRecurringRefreshDayRequest

This API changes the refresh day for the monthly recurring quota accounting, which is the billing date. This API replaces the old MsBM API ChangeRecurringRefreshDayOfTheMonth.

For a monthly refreshing quota, this would change the refresh day of the month to the specified value and the quota would always refresh on that day of the month.

For a non-month based refresh period, the value passed in will just denote the next day of the month the quota will refresh. After that date the quota will continue to refresh on the frequency set up in reference data, i.e. every 10 days, every 2 weeks, etc. The time element of the new datetime will be zeroed out.



### Note What This API Does NOT Do!

This API was designed/tested for monthly recurring quotas only. The behavior with non-monthly refresh periods is eccentric and not guaranteed to behave as desired.

### Schema

```

<ChangeRecurringRefreshDayRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <balanceCode> xsd:string </balanceCode> [1]
  <quotaCode> xsd:string </quotaCode> [1]
  <newDayOfTheMonth> xsd:integer (1 >= value <= 28) </newDayOfTheMonth> [1] ?
  <resetCreditEndDates> xsd:boolean </resetCreditEndDates> [0..1] ?
</ChangeRecurringRefreshDayRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeRecurringRefreshDayRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <balanceCode>DATA</balanceCode>
      <quotaCode>Recurring</quotaCode>
      <newDayOfTheMonth>15</newDayOfTheMonth>
      <resetCreditEndDates>true</resetCreditEndDates>
    </ChangeRecurringRefreshDayRequest>
  </se:Body>
</se:Envelope>

```

## ChangeRecurringRefreshDayResponse

### Schema

```

<ChangeRecurringRefreshDayResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</ChangeRecurringRefreshDayResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeRecurringRefreshDayResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ChangeRecurringRefreshDayResponse>
  </se:Body>
</se:Envelope>

```

## ChangeSubscriberAvpsRequest

This API changes a subscriber's or subaccount's AVPs. The order of operation is delete, modify, add (new). The modify avps only modify the value.

### Schema

```

<ChangeSubscriberAvpsRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <deletedAvp> AvpType </deletedAvp> [0..100]
  <modifiedAvp> AvpType </modifiedAvp> [0..100]
  <newAvp> AvpType </newAvp> [0..100]

```

```
</ChangeSubscriberAvpsRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeSubscriberAvpsRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <deletedAvp>
        <code>AVP_CODE</code>
        <value>VALUE</code>
      </deletedAvp>
      <modifiedAvp>
        <code>AVP_CODE</code>
        <value>VALUE</code>
        <newValue>NEW_VALUE</newValue>
      </modifiedAvp>
      <newAvp>
        <code>AVP_CODE</code>
        <value>VALUE</code>
      </newAvp>
    </ChangeSubscriberAvpsRequest>
  </se:Body>
</se:Envelope>
```

## ChangeSubscriberAvpsResponse

### Schema

```
<ChangeSubscriberAvpsResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</ChangeSubscriberAvpsResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeSubscriberAvpsResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ChangeSubscriberAvpsResponse>
  </se:Body>
</se:Envelope>
```

## ChangeSubscriberStatusRequest

This API changes the subscriber's and all subaccounts' status. The changeAll boolean flag default == true which maintains backwards compatibility. When changeAll == false, the API will change the status for the specified subscriber only (sub account or parent account)

### Schema

```
<ChangeSubscriberStatusRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
```

```

    <status> StatusType </status> [1]
    <changeAll> xsd:boolean </changeAll> [0..1]
  </ChangeSubscriberStatusRequest>

```

**Example**

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeSubscriberStatusRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <status>ACTIVE</status>
    </ChangeSubscriberStatusRequest>
  </se:Body>
</se:Envelope>

```

## ChangeSubscriberStatusResponse

**Schema**

```

<ChangeSubscriberStatusResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</ChangeSubscriberStatusResponse>

```

**Example**

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ChangeSubscriberStatusResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ChangeSubscriberStatusResponse>
  </se:Body>
</se:Envelope>

```

## CreateBalanceRequest

This API provisions balance/quota for a given subscriber. This API replaces the old MsBM API ProvisionSubscriber.

If the balance already exists, the API will just credit the existing balance unless the quota is recurring, then it throws an error.

By default, CreateBalance will set lastRecurringRefresh to the current datetime and WILL NOT zero out the time to midnight.

**Note** CreateSubscriberRequest and UpdateSubscriberRequest

CreateSubscriber and UpdateSubscriber can also provision initial balance for a subscriber.

**Note** Removed Elements

The nextRefreshDate and avp elements in a threshold have been removed because they are not used by the Balance module.

**Schema**

```
<CreateBalanceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <balance> CreateBalanceType </balance> [1..10]
</CreateBalanceRequest>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateBalanceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <balance>
        <code>DATA</code>
        <quotaCode>RECURRING</quotaCode>
        <startDate>2011-01-01T00:00:00Z</startDate>
        <expirationDate>2012-01-01T00:00:00Z</expirationDate>
        <initialAmount>500</initialAmount>
        <lastRecurringRefresh>2011-01-01T00:00:00Z</lastRecurringRefresh>
      </balance>
    </CreateBalanceRequest>
  </se:Body>
</se:Envelope>
```

## CreateBalanceResponse

**Schema**

```
<CreateBalanceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</CreateBalanceResponse>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateBalanceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </CreateBalanceResponse>
  </se:Body>
</se:Envelope>
```

## CreateSubscriberRequest

This API creates a subscriber in the USuM database. It also allows provisioning initial quota/balance using the CreateBalance element.

### Schema

```
<CreateSubscriberRequest>
  <audit> AuditType </audit> [0..1] ?
  <subscriber> SubscriberType </subscriber> [1]
</CreateSubscriberRequest>
```

### Example

#### Example 1 - Minimum required fields

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateSubscriberRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <subscriber>
        <credential>
          <networkId>uniqueIdentifier</networkId>
          <password>password</password>
        </credential>
        <status>ACTIVE</status>
      </subscriber>
    </CreateSubscriberRequest>
  </se:Body>
</se:Envelope>
```

#### Example 2 - A more complete subscriber with multiple credentials and a service and some additional fields including AVP examples

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateSubscriberRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <subscriber>
        <name>
          <fullName>Test User</fullName>
        </name>
        <credential>
          <networkId>uniqueIdentifier</networkId>
          <password>password</password>
          <expirationDate>2011-12-31T23:59:59Z</expirationDate>
        </credential>
        <credential>
          <networkId>uniqueIdentifier</networkId>
          <password>password</password>
        </credential>
        <service>
          <code>SERVICE_CODE</code>
          <enabled>true</enabled>
          <avp>
            <code>AVP_CODE</code>
          </avp>
        </service>
      </subscriber>
    </CreateSubscriberRequest>
  </se:Body>
</se:Envelope>
```



```

        <value>AVP_VALUE</value>
      </avp>
    </avp>
    <code>AVP_CODE_2</code>
    <value>AVP_VALUE</value>
  </avp>
</service>
<status>ACTIVE</status>
<startDate>2011-01-01T00:00:00+00:00</startDate>
<endDate>2031-01-01T00:00:00+00:00</endDate>
<billingInfo>
  <chargingId>123456789</chargingId>
</billingInfo>
<avp>
  <code>AVP_CODE</code>
  <value>AVP_VALUE</value>
</avp>
<avp>
  <code>AVP_CODE_2</code>
  <value>AVP_VALUE</value>
</avp>
</subscriber>
</CreateSubscriberRequest>
</se:Body>
</se:Envelope>

```

Example 3 - CreateBalance xml can be added to the request to provision quota when the subscriber is created

```

<createBalance>
  <code>DATA</code>
  <quotaCode>Recurring</quotaCode>
  <startDate>2011-01-01T00:00:00Z</startDate>
  <expirationDate>2012-01-01T00:00:00Z</expirationDate>
  <initialAmount>500</initialAmount>
  <lastRecurringRefresh>2011-01-01T00:00:00Z</lastRecurringRefresh>
</createBalance>

```

## CreateSubscriberResponse

### Schema

```

<CreateSubscriberResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</CreateSubscriberResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </CreateSubscriberResponse>
  </se:Body>
</se:Envelope>

```

## CreateSubscribersRequest

This API creates multiple subscribers in the USuM database. It also allows provisioning initial quota/balance using the CreateBalance element.

### Schema

```
<CreateSubscribersRequest>
  <audit> AuditType </audit> [0..1] ?
  <subscriber> SubscriberType </subscriber> [1..1000]
</CreateSubscribersRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateSubscribersRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <subscriber>
        ...
      </subscriber>
      <subscriber>
        ...
      </subscriber>
    </CreateSubscribersRequest>
  </se:Body>
</se:Envelope>
```

## CreateSubscribersResponse

### Schema

```
<CreateSubscribersResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</CreateSubscribersResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </CreateSubscribersResponse>
  </se:Body>
</se:Envelope>
```

## CreateVoucherRequest

This API adds a voucher to the system.

### Schema

```
<CreateVoucherRequest>
  <audit> AuditType </audit> [0..1] ?
  <voucher> VoucherType </voucher> [1]
```

```
</CreateVoucherRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateVoucherRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <voucher>
        <code>voucher</code>
        <pin>pin</pin>
        <maxConcurrentSessions>5</maxConcurrentSessions>
        <duration>10</duration>
        <durationMeasure>Minutes</durationMeasure>
        <serviceCode>serviceCode</serviceCode>
        <expirationDate>2012-01-01T00:00:00Z</expirationDate>
        <locationCode>location</locationCode>
        <timeQuota>10</timeQuota>
        <timeMeasure>timemeasure</timeMeasure>
      </voucher>
    </CreateVoucherRequest>
  </se:Body>
</se:Envelope>
```

## CreateVoucherResponse

### Schema

```
<CreateVoucherResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</CreateVoucherResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateVoucherResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </CreateVoucherResponse>
  </se:Body>
</se:Envelope>
```

## CreateVouchersRequest

This API adds multiple vouchers to the system.

### Schema

```
<CreateVouchersRequest>
  <audit> AuditType </audit> [0..1] ?
  <voucher> VoucherType </voucher> [1..100]
</CreateVouchersRequest>
```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateVouchersRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <voucher>
        ...
      </voucher>
      <voucher>
        ...
      </voucher>
    </CreateVouchersRequest>
  </se:Body>
</se:Envelope>

```

## CreateVouchersResponse

### Schema

```

<CreateVouchersResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</CreateVouchersResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateVouchersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </CreateVouchersResponse>
  </se:Body>
</se:Envelope>

```

## CreditRequest

This API credits quota for a subscriber.

### Schema

```

<CreditRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <balanceCode> xsd:string </balanceCode> [1] ?
  <quotaCode> xsd:string </quotaCode> [1] ?
  <amount> xsd:long </amount> [1] ?
  <startDate> xsd:dateTime </startDate> [1] ?
  <expirationDate> xsd:dateTime </expirationDate> [1] ?
</CreditRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreditRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>

```

```

    </audit>
    <networkId>uniqueIdentifier</networkId>
    <balanceCode>DATA</balanceCode>
    <quotaCode>RECURRING</quotaCode>
    <amount>100</amount>
    <startDate>2011-01-01T00:00:00Z</startDate>
    <expirationDate>2012-01-01T00:00:00Z</expirationDate>
  </CreditRequest>
</se:Body>
</se:Envelope>

```

## CreditResponse

### Schema

```

<CreditResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <returnCredit> ReturnCreditType </returnCredit> [1]
</CreditResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreditResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <returnCredit>
        <creditId>wertDSD1234sfsdge5657yfc</creditId>
        <nextRefreshDate>2012-11-28T00:00:00Z</nextRefreshDate>
        <balanceRemaining>250</balanceRemaining>
        <amountCredited>50</amountCredited>
        <callbackValidityTime>2012-11-28T00:00:00Z</callbackValidityTime>
      </returnCredit>
    </CreditResponse>
  </se:Body>
</se:Envelope>

```

## DebitRequest

This API debits quota for a subscriber.

### Schema

```

<DebitRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <balanceCode> xsd:string </balanceCode> [1] ?
  <quotaCode> xsd:string </quotaCode> [0..1] ?
  <amount> xsd:long </amount> [1] ?
</DebitRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DebitRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>

```

```

    </audit>
    <networkId>networkId</networkId>
    <balanceCode>DATA</balanceCode>
    <quotaCode>Recurring</quotaCode>
    <amount>200</amount>
  </DebitRequest>
</se:Body>
</se:Envelope>

```

## DebitResponse

### Schema

```

<DebitResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <returnDebit> ReturnDebitType </returnDebit> [1]
</DebitResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DebitResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <returnDebit>
        <nextRefreshDate>2012-11-28T00:00:00Z</nextRefreshDate>
        <balanceRemaining>250</balanceRemaining>
        <amountDebited>50</amountDebited>
        <callbackValidityTime>2012-11-28T00:00:00Z</callbackValidityTime>
        <exhausted>>false</exhausted>
      </returnDebit>
    </DebitResponse>
  </se:Body>
</se:Envelope>

```

## DeleteBalanceRequest

This API removes balance from a subscriber. This API does not deprovision a balance. To deprovision a balance, use UpdateBalance.

### Schema

```

<DeleteBalanceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <code> xsd:string </code> [1] ?
  <hardDelete> xsd:boolean </hardDelete> [0..1] ?
</DeleteBalanceRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteBalanceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
    </DeleteBalanceRequest>
  </se:Body>
</se:Envelope>

```

```

        <networkId>uniqueIdentifier</networkId>
        <code>DATA</code>
        <hardDelete>>false</hardDelete>
    </DeleteBalanceRequest>
</se:Body>
</se:Envelope>

```

## DeleteBalanceResponse

### Schema

```

<DeleteBalanceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteBalanceResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteBalanceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteBalanceResponse>
  </se:Body>
</se:Envelope>

```

## DeleteCredentialRequest

This API deletes a credential from a subscriber.

### Schema

```

<DeleteCredentialRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
</DeleteCredentialRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteCredentialRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
    </DeleteCredentialRequest>
  </se:Body>
</se:Envelope>

```

## DeleteCredentialResponse

### Schema

```

<DeleteCredentialResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?

```

```
</DeleteCredentialResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteCredentialResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteCredentialResponse>
  </se:Body>
</se:Envelope>
```

## DeleteCredentialsRequest

This API deletes a set of credentials from a subscriber.




---

### Note Minimum Number of Credentials

A subscriber must always have at least 1 credential. Therefore, as the API removes credentials, once there is only one left, it stops processing.




---

### Note Delete From One Subscriber

This API is only supposed to delete from one subscriber at a time. The API uses the credentials passed in to find a subscriber or sub account, and the first match is used. Therefore, the API should only send in a set of credentials from the same subscriber or sub account. Once a subscriber has been found, then each credential is checked for a match. Any credentials not matched are ignored.

---

### Schema

```
<DeleteCredentialsRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1..5]
</DeleteCredentialsRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteCredentialsRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <networkId>uniqueIdentifier1</networkId>
      <networkId>uniqueIdentifier2</networkId>
      <networkId>uniqueIdentifier3</networkId>
    </DeleteCredentialsRequest>
  </se:Body>
</se:Envelope>
```



## DeleteCreditRequest

### Schema

```
<DeleteCreditRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <balanceCode> xsd:string </balanceCode> [1] ?
  <quotaCode> xsd:string </quotaCode> [1] ?
  <creditId> xsd:string </creditId> [1] ?
</DeleteCreditRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteCreditRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <balanceCode>DATA</balanceCode>
      <quotaCode>RECURRING</quotaCode>
      <creditId>1</creditId>
    </DeleteCreditRequest>
  </se:Body>
</se:Envelope>
```

## DeleteCreditResponse

### Schema

```
<DeleteCreditResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteCreditResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteCreditResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteCreditResponse>
  </se:Body>
</se:Envelope>
```

## DeleteQuotaRequest

This API removes a quota from a balance. This API replaces the old MsBM API RemoveQuota. The hardDelete element has been added to provide operational consistency with DeleteBalance.

### Schema

```
<DeleteQuotaRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
```

```

    <balanceCode> xsd:string </balanceCode> [1] ?
    <code> xsd:string </code> [1] ?
    <hardDelete> xsd:boolean </hardDelete> [0..1] ?
  </DeleteQuotaRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteQuotaRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <balanceCode>DATA</balanceCode>
      <code>RECURRING</code>
      <hardDelete>true</hardDelete>
    </DeleteQuotaRequest>
  </se:Body>
</se:Envelope>

```

## DeleteQuotaResponse

### Schema

```

<DeleteQuotaResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteQuotaResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteQuotaResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteQuotaResponse>
  </se:Body>
</se:Envelope>

```

## DeleteServiceRequest

This API deletes services and corresponding Balance entries for a given subscriber.



### Note Deleting Old Balances

The deleteOldBalance field is set to false by default. If it is set to true and no balanceCodes are sent in the request, then ALL balances will be removed from the account.

**Note Automatic Balance Provisioning**

This feature allows Service Options to be configured so that when the Service is applied to the Subscriber, Balance is automatically provisioned. (It also removes Balance when a Service is removed.) If the system is setup for auto-provisioning, do not include the Balance object when using this API. That is, only send in the Balance object when the system is not setup for auto-provisioning.

**Schema**

```
<DeleteServiceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <serviceCode> xsd:string </serviceCode> [1..10]
  <balance> DeleteBalanceType </balance> [0..10] ?
  <hardDelete> xsd:boolean </hardDelete> [0..1] ?
  <deleteOldBalance> xsd:boolean </deleteOldBalance> [0..1] ?
</DeleteServiceRequest>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteServiceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <serviceCode>SERVICE_CODE</serviceCode>
      <balance>
        <code>BALANCE_CODE</code>
        <quotaCode>QUOTA_CODE</quotaCode>
      </balance>
      <hardDelete>true</hardDelete>
    </DeleteServiceRequest>
  </se:Body>
</se:Envelope>
```

## DeleteServiceResponse

**Schema**

```
<DeleteServiceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteServiceResponse>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteServiceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteServiceResponse>
  </se:Body>
</se:Envelope>
```

## DeleteSubscriberRequest

This API deletes a subscriber from the USuM database.



### Note Renamed

The API has been renamed. It is DeleteSubscriberByKey in v1.0 API.

### Schema

```
<DeleteSubscriberRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <hardDelete> xsd:boolean </hardDelete> [0..1] ?
</DeleteSubscriberRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteSubscriberRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <hardDelete>>false</hardDelete>
    </DeleteSubscriberRequest>
  </se:Body>
</se:Envelope>
```

## DeleteSubscriberResponse

### Schema

```
<DeleteSubscriberResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteSubscriberResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteSubscriberResponse>
  </se:Body>
</se:Envelope>
```

## DeleteSubscriberSsidRequest

This API deletes all the SSIDs for the given subscriber from the SSID extension collection.

### Schema

```
<DeleteSubscriberSsidRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
</DeleteSubscriberSsidRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteSubscriberSsidRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>the credential id of the provisioned subscriber whose records have to be
deleted from Subscriber SSID extension table</networkId>
    </DeleteSubscriberSsidRequest>
  </se:Body>
</se:Envelope>
```

## DeleteSubscriberSsidResponse

### Schema

```
<DeleteSubscriberSsidResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteSubscriberSsidResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteSubscriberSsidResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteSubscriberSsidResponse>
  </se:Body>
</se:Envelope>
```

## DeleteVoucherBatchRequest

This API deletes a batch of vouchers from the system.

The API uses a code and/or pin value with maskChars to match the vouchers to delete. The code and pin values are 'mask' values which define a range of values to match.

See the code and pin and maskChars fields in the VoucherBatchKeyType for more information.

### Schema

```
<DeleteVoucherBatchRequest>
  <audit> AuditType </audit> [0..1] ?
  <key> VoucherKeyType </key> [1]
</DeleteVoucherBatchRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
```

```

<DeleteVoucherBatchRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
  <audit>
    <id>username</id>
    <comment>comment</comment>
  </audit>
  <key>
    <voucherBatchKey>
      <code>codeMask****</code>
      <pin>pinMask####</pin>
      <maskChars>abcd1234</maskChars>
    </voucherBatchKey>
  </key>
</DeleteVoucherBatchRequest>
</se:Body>
</se:Envelope>

```

## DeleteVoucherBatchResponse

### Schema

```

<DeleteVoucherBatchResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteVoucherBatchResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteVoucherBatchResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteVoucherBatchResponse>
  </se:Body>
</se:Envelope>

```

## DeleteVoucherRequest

This API deletes a voucher from the system.

### Schema

```

<DeleteVoucherRequest>
  <audit> AuditType </audit> [0..1] ?
  <code> xsd:string </code> [1]
</DeleteVoucherRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteVoucherRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>voucher</code>
    </DeleteVoucherRequest>
  </se:Body>
</se:Envelope>

```

## DeleteVoucherResponse

### Schema

```
<DeleteVoucherResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteVoucherResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteVoucherResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteVoucherResponse>
  </se:Body>
</se:Envelope>
```

## ExecuteActionRequest

This API provides a mechanism to run commands against the Policy Engine.

### Location Query

The Location Query ExecuteActionRequest replaces the original Policy Engine getCompleteId request and returns location data plus AVPs from the relevant network device.

#### Parameters

- code - location-query
- args
  - ip-address - the framed ip address of the subscriber's device.
  - port - the port of the subscriber's device.
  - type - indicates the type of network device to check. If the value is set to the string literal 'null', then no device is checked. e.g. isg, car, asr9k or null

### Network Device Query

The Network Device Query ExecuteActionRequest returns AVPs from the relevant network device like the getCompleteId request.

#### Parameters

- code - network\_device\_query
- args
  - ip-address - the framed ip address of the subscriber's device.
  - port - the port of the subscriber's device.

- type - indicates the type of network device to check. If the value is set to the string literal 'null', then no device is checked. e.g. isg, car, asr9k or null

### One Time Code

The One Time Code process is 2-step process (Create and Validate). The first request, Create One Time Code, establishes a unique code and dummy subscriber in the database. The code value is then sent to the user via SMS or email. That user enters that code which is then used in the 2nd step, Validate One Time Code. If validation of the code against the dummy subscriber in the database is successful, the dummy subscriber is removed and the API returns success.

### Policy Builder Custom Configuration

There are built in extension points in the Policy Builder blueprints that allow for additional custom actions to be performed when these 2 API requests are made. The most common thing is to have additional checks against 3rd party datastores.

### Create One Time Code

Parameters

- code - create-onetimecode-request
- args
  - validity-time-min - the time in minutes for how long the code is valid e.g. 10
  - code-length - the length of the generated random code value e.g. 5
  - code-chars - the set of characters to use for the generated code e.g. abcd1234
  - type - specifies the type of identity used for validating the user. e.g. email or mobile
  - identity - specifies a credential used for validating the user against the external system. e.g. test@gmail.com or 13035551212
  - destination - specifies the destination address to send the generated code. This is generally the same as the identity field.
  - dest-type - specifies the notification mechanism used to send the generated code to the user user. Note this is different then the type field since it specifies the transport used to send the code to the user.
    - A dest-type of native-email will use the CPS SMTP interface with a notification template of OTC-EMAIL
    - A dest-type of native-sms will use the CPS SMSC interface with a notification template of OTC-SMS
    - Any other value for dest-type will fall through the Non-Native policy portion of the configuration. In the Non-Native scenario, all notification templates are available.

### Validate One Time Code

Parameters



- code - validate-onetimecode-request
- args
  - code - the random unique code generated in the Create One Time Code request and sent to the user via sms or email. e.g. ser45tfd
  - identity - specifies a credential used for validating the user against the external system. e.g. test@gmail.com or 13035551212



---

**Note** **Passcode and Identity Variables**

The OTC-EMAIL and OTC-SMS notification templates are pre-configured with \$passcode and \$identity keywords on which CPS can perform variable substitution. This is the mechanism of how the sms or email message gets the generated random code.

---

**Password Reset with One Time Password**

The Password Reset ExecuteActionRequest provides a way for subscribers to reset their passwords which is a common function of most websites. The Password Reset with One Time Password functionality piggy-backs on the Create One Time Code request.

**Parameters**

- code - create-onetimecode-request
- args
  - validity-time-min - the time in minutes for how long the code is valid e.g. 10
  - code-length - the length of the generated random code value e.g. 5
  - code-chars - the set of characters to use for the generated code e.g. abcd1234
  - type - specifies the type of identity used for validating the user. e.g. email or mobile
  - identity - specifies a credential used for validating the user against the external system. e.g. test@gmail.com#!PWR#! or 13035551212#!PWR#!
  - destination - specifies the destination address to send the generated code. This is generally the same as the identity field.
  - dest-type - specifies the notification mechanism used to send the generated code to the user user. Note this is different then the type field since it specifies the transport used to send the code to the user.
    - A dest-type of native-email will use the CPS SMTP interface with a notification template of OTC-EMAIL
    - A dest-type of native-sms will use the CPS SMSC interface with a notification template of OTC-SMS
    - Any other value for dest-type will fall through the Non-Native policy portion of the configuration. In the Non-Native scenario, all notification templates are available.

**Note Append #!PWR#!**

You must append #!PWR#! to the identity value. The reason for this is that unlike the One Time Code flow which assumes the subscriber is not in the SPR database, the Password Reset flow is used with a subscriber that is already in the SPR database. However, the Create One Time Code action fails if a subscriber profile is found for the given username. Therefore, if you append a unique value such as #!PWR#!, the action will not find a subscriber and allow the process to continue.

**Validate Registration**

The Validate Registration ExecuteActionRequest provides a mechanism to validate subscribers against external systems. It is similar to the One Time Code APIs but it only works against the 3rd party datastores. Therefore, it will always require additional Policy Builder configuration via the Validate Registration extension point.

## Parameters

- code - validate-registration-request
- args
  - type - specifies the type of identity used for validating the user. e.g. email or mobile
  - identity - unique identifier used for validation
  - validation-identity - additional unique identifier used for validation - some systems require 2 values
  - password

**Start Session**

The Start Session ExecuteActionRequest replaces the original Policy Engine setupSubscriberProfile request. It initiates a session on a network device. Currently, supports ISG, CAR, ASR5K, and ASR9K.

## Parameters

- code - start-session
- args
  - ISG\_IP - the framed ip address of the subscriber's device. e.g. 172.12.9.1 NOTE: The L4 redirect will provide this information to the portal.
  - PORT\_NUMBER - the port of the subscriber's device. e.g. 1234 NOTE: The L4 redirect will provide this information to the portal.
  - type - indicates the type of network device that will be used. e.g. isg, car, or asr9k. If the value is set to the string literal 'null', the API will fall through and do generic setup based on blueprints and policy configuration.
  - USER\_NAME - username of the subscriber.
  - PASSWORD - password of the subscriber.

## Refresh Device Usage

The Refresh Device Usage Query ExecuteActionRequest returns is an asynchronous request that triggers the network device such as ISG or SCE to update the accounting usage data for a given subscriber. This causes CPS to update the data as well.

### Parameters

- code - refresh-device-usage
- key
  - code - USuMSubscriberIdKey or USuMCredentialKey
  - keyField code - usumSubscriberId or networkId: if you are using the USuMSubscriberIdKey:usumSubscriberId then the value is the database id value of the subscriber e.g. 4f5e4cc0e4b027c556fd0d7c. If you are using the USuMCredentialKey:networkId then the value is the subscriber's username e.g. username@gmail.com
- args
  - code - bucket-id. The name of the quota bucket to refresh if this is supported by the network device.

### Schema

```
<ExecuteActionRequest>
  <audit> AuditType </audit> [0..1] ?
  <key> SessionKeyType </key> [0..1]
  <code> xsd:string </code> [1]
  <arg> AvpType </arg> [0..100]
</ExecuteActionRequest>
```

### Example

#### Location Query

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>location-query</code>
      <arg>
        <code>ip-address</code>
        <value>0.0.0.0</value>
      </arg>
      <arg>
        <code>port</code>
        <value>PBHK</value>
      </arg>
      <arg>
        <code>type</code>
        <value>isg|car|asr9k|null</value>
      </arg>
      <arg>
        <code>USERID_QUERY</code>
        <value>userId</value>
      </arg>
    </ExecuteActionRequest>
  </se:Body>
</se:Envelope>
```

```

        <code>MSISDN_QUERY</code>
        <value>msisdn</value>
    </arg>
    <arg>
        <code>IMSI_QUERY</code>
        <value>imsi</value>
    </arg>
    <arg>
        <code>MAC_QUERY</code>
        <value>mac address</value>
    </arg>
    <arg>
        <code>CIRCUIT_ID_QUERY</code>
        <value>circuit id</value>
    </arg>
</ExecuteActionRequest>
</se:Body>
</se:Envelope>

```

#### Network Device Query

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>network_device_query</code>
      <arg>
        <code>ip-address</code>
        <value>0.0.0.0</value>
      </arg>
      <arg>
        <code>port</code>
        <value>PBHK</value>
      </arg>
      <arg>
        <code>type</code>
        <value>isg|car|asr9k|null</value>
      </arg>
    </ExecuteActionRequest>
  </se:Body>
</se:Envelope>

```

#### Create One Time Code

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>create-onetimecode-request</code>
      <arg>
        <code>type</code>
        <value>email</value>
      </arg>
      <arg>
        <code>dest-type</code>
        <value>native-sms</value>
      </arg>
    </ExecuteActionRequest>
  </se:Body>
</se:Envelope>

```

```

        <code>destination</code>
        <value>3035551212</value>
    </arg>
    <arg>
        <code>identity</code>
        <value>uniqueIdentifier</value>
    </arg>
    <arg>
        <code>validity-time-min</code>
        <value>10</value>
    </arg>
    <arg>
        <code>code-length</code>
        <value>5</value>
    </arg>
    <arg>
        <code>code-chars</code>
        <value>abcdefgh</value>
    </arg>
</ExecuteActionRequest>
</se:Body>
</se:Envelope>

```

Validate One Time Code

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>validate-onetimecode-request</code>
      <arg>
        <code>code</code>
        <value>sw34edft5</value>
      </arg>
      <arg>
        <code>identity</code>
        <value>uniqueIdentifier</value>
      </arg>
    </ExecuteActionRequest>
  </se:Body>
</se:Envelope>

```

Password Reset with One Time Password

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>create-onetimecode-request</code>
      <arg>
        <code>type</code>
        <value>email</value>
      </arg>
      <arg>
        <code>dest-type</code>
        <value>native-sms</value>
      </arg>
      <arg>

```

```

        <code>destination</code>
        <value>3035551212</value>
    </arg>
    <arg>
        <code>identity</code>
        <value>uniqueIdentifier#!PWR#!</value>
    </arg>
    <arg>
        <code>validity-time-min</code>
        <value>10</value>
    </arg>
    <arg>
        <code>code-length</code>
        <value>5</value>
    </arg>
    <arg>
        <code>code-chars</code>
        <value>abcdefgh</value>
    </arg>
</ExecuteActionRequest>
</se:Body>
</se:Envelope>

```

#### Validate Registration

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>validate-registration-request</code>
      <arg>
        <code>type</code>
        <value>email</value>
      </arg>
      <arg>
        <code>identity</code>
        <value>uniqueIdentifier</value>
      </arg>
      <arg>
        <code>validation-identity</code>
        <value>1234567</value>
      </arg>
      <arg>
        <code>password</code>
        <value>Password</value>
      </arg>
    </ExecuteActionRequest>
  </se:Body>
</se:Envelope>

```

#### Start Session

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <code>start-session</code>
      <arg>

```

```

        <code>ISG_IP</code>
        <value>10.10.10.10</value>
    </arg>
    <arg>
        <code>PORT_NUMBER</code>
        <value>PBHK</value>
    </arg>
    <arg>
        <code>type</code>
        <value>isg|car|asr9k|null</value>
    </arg>
    <arg>
        <code>USER_NAME</code>
        <value>123456789</value>
    </arg>
    <arg>
        <code>PASSWORD</code>
        <value>password</value>
    </arg>
</ExecuteActionRequest>
</se:Body>
</se:Envelope>

```

Refresh Device Usage No Buckets with USuMSubscriberIdKey

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <code>USuMSubscriberIdKey</code>
        <primary>false</primary>
        <keyField>
          <code>usumSubscriberId</code>
          <value>4f5e4cc0e4b027c556fd0d7c</value>
        </keyField>
      </key>
      <code>refresh-device-usage</code>
    </ExecuteActionRequest>
  </se:Body>
</se:Envelope>

```

Refresh Device Usage For A Specific Bucket with USuMCredentialKey

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <code>USuMCredentialKey</code>
        <primary>false</primary>
        <keyField>
          <code>networkId</code>
          <value>username@gmail.com</value>
        </keyField>
      </key>
      <code>refresh-device-usage</code>
      <arg>
        <code>bucket-id</code>
        <value>GENERIC_USAGE</value>
      </arg>
    </ExecuteActionRequest>
  </se:Body>
</se:Envelope>

```

## ExecuteActionResponse

### Schema

```
<ExecuteActionResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <executeActionResponse> ExecuteActionResponseType </executeActionResponse> [0..500]
</ExecuteActionResponse>
```

### Example

Location Query type=null

```
<se:Envelope
xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"><se:Body><ExecuteActionResponse
xmlns="http://broadhop.com/unifiedapi/soap/types">
  <errorCode>0</errorCode>
  <errorMessage>Request completed successfully</errorMessage>
  <executeActionResponse>
    <responseObject>
      <entry>
        <string>_type</string>
        <string>QueryResult</string>
      </entry>
      <entry>
        <string>queryAvps</string>
        <list>
          <map>
            <entry>
              <string>_type</string>
              <string>QueryAvp</string>
            </entry>
            <entry>
              <string>value</string>
              <string>BHOP</string>
            </entry>
            <entry>
              <string>code</string>
              <string>locationCode</string>
            </entry>
          </map>
          <map>
            <entry>
              <string>_type</string>
              <string>QueryAvp</string>
            </entry>
            <entry>
              <string>value</string>
              <string>USuM Authorization</string>
            </entry>
            <entry>
              <string>code</string>
              <string>domainCode</string>
            </entry>
          </map>
          <map>
            <entry>
              <string>_type</string>
              <string>QueryAvp</string>
            </entry>
            <entry>
```



```

        <string>value</string>
        <string>>false</string>
    </entry>
    <entry>
        <string>code</string>
        <string>loggedIn</string>
    </entry>
</map>
</list>
</entry>
</responseObject>
</executeActionResponse>
</ExecuteActionResponse></se:Body></se:Envelope>

```

Location Query type=isg

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <executeActionResponse>
        <responseObject>
          <entry>
            <string>_type</string>
            <string>QueryResult</string>
          </entry>
          <entry>
            <string>queryAvps</string>
            <list>
              <map>
                <entry>
                  <string>_type</string>
                  <string>QueryAvp</string>
                </entry>
                <entry>
                  <string>value</string>
                  <string>0</string>
                </entry>
                <entry>
                  <string>code</string>
                  <string>NAS-PORT</string>
                </entry>
              </map>
            </list>
          </entry>
          <entry>
            <string>_type</string>
            <string>QueryAvp</string>
          </entry>
          <entry>
            <string>value</string>
            <string>No valid Session</string>
          </entry>
          <entry>
            <string>code</string>
            <string>REPLY-MESSAGE</string>
          </entry>
        </map>
      </executeActionResponse>
    </ExecuteActionResponse>
  </se:Body>
</se:Envelope>

```

```

        <string>value</string>
        <string>sg-version=1.0</string>
    </entry>
</entry>
    <string>code</string>
    <string>CISCO-AVPAIR</string>
</entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>503</string>
    </entry>
    <entry>
        <string>code</string>
        <string>ERROR-CAUSE</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>BHOP</string>
    </entry>
    <entry>
        <string>code</string>
        <string>locationCode</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>S10.10.12.10:68</string>
    </entry>
    <entry>
        <string>code</string>
        <string>CISCO-ACCOUNT-INFO</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>sg-version=1.0</string>
    </entry>
    <entry>
        <string>code</string>
        <string>CISCO-AVPAIR</string>
    </entry>
</map>

```

```

<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>No valid Session</string>
  </entry>
  <entry>
    <string>code</string>
    <string>REPLY-MESSAGE</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>sg-version=1.0</string>
  </entry>
  <entry>
    <string>code</string>
    <string>CISCO-AVPAIR</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>S10.10.12.10</string>
  </entry>
  <entry>
    <string>code</string>
    <string>CISCO-ACCOUNT-INFO</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>sg-version=1.0</string>
  </entry>
  <entry>
    <string>code</string>
    <string>CISCO-AVPAIR</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>172.17.1.11</string>
  </entry>
</map>

```

```

    </entry>
  <entry>
    <string>code</string>
    <string>FRAMED-IP-ADDRESS</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>S10.10.12.10</string>
  </entry>
  <entry>
    <string>code</string>
    <string>CISCO-ACCOUNT-INFO</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>172.17.1.11</string>
  </entry>
  <entry>
    <string>code</string>
    <string>FRAMED-IP-ADDRESS</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>USuM Authorization</string>
  </entry>
  <entry>
    <string>code</string>
    <string>domainCode</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>QueryAvp</string>
  </entry>
  <entry>
    <string>value</string>
    <string>>false</string>
  </entry>
  <entry>
    <string>code</string>
    <string>loggedIn</string>
  </entry>
</map>
<map>
  <entry>

```

```

        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>nas-port:0.0.0.0:0/0/0</string>
    </entry>
    <entry>
        <string>code</string>
        <string>NAS-PORT-ID</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>S10.10.12.10:68</string>
    </entry>
    <entry>
        <string>code</string>
        <string>CISCO-ACCOUNT-INFO</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>$MA00c.29fl.d050</string>
    </entry>
    <entry>
        <string>code</string>
        <string>CISCO-ACCOUNT-INFO</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>503</string>
    </entry>
    <entry>
        <string>code</string>
        <string>ERROR-CAUSE</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>503</string>
    </entry>
    <entry>

```

```

        <string>code</string>
        <string>ERROR-CAUSE</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>$MA000c.29f1.d050</string>
    </entry>
    <entry>
        <string>code</string>
        <string>CISCO-ACCOUNT-INFO</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>sg-version=1.0</string>
    </entry>
    <entry>
        <string>code</string>
        <string>CISCO-AVPAIR</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>No valid Session</string>
    </entry>
    <entry>
        <string>code</string>
        <string>REPLY-MESSAGE</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>nas-port:0.0.0.0:0/0/0</string>
    </entry>
    <entry>
        <string>code</string>
        <string>NAS-PORT-ID</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>

```

```

        </entry>
        <entry>
            <string>value</string>
            <string>0</string>
        </entry>
        <entry>
            <string>code</string>
            <string>NAS-PORT</string>
        </entry>
    </map>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>QueryAvp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>S10.10.12.10</string>
    </entry>
    <entry>
        <string>code</string>
        <string>CISCO-ACCOUNT-INFO</string>
    </entry>
</map>
</list>
</entry>
</responseObject>
</executeActionResponse>
</ExecuteActionResponse>
</se:Body>
</se:Envelope>

```

Create One Time Code Success

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ExecuteActionResponse>
  </se:Body>
</se:Envelope>

```

Create One Time Code Failure

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>2</errorCode>
      <errorMessage>com.broadhop.exception.BroadhopException: OTC001: User is already
registered. One time code generation is disabled
com.broadhop.spr.impl.actions.CreateOneTimeCode.execute(CreateOneTimeCode.java36)
com.broadhop.policy.impl.RulesPolicyService.processSyncAction(RulesPolicyService.java486)
com.broadhop.policy.Rule_Validate_Code_247e55a8_1da4_4bf9_b095_d826bf53748d_yHv0IaBQFeGrGauRX01s5g_0.consequence(Unknown
Source)
com.broadhop.policy.Rule_Validate_Code_247e55a8_1da4_4bf9_b095_d826bf53748d_yHv0IaBQFeGrGauRX01s5g_0.consequenceInvoker.evaluate(Unknown
Source)
org.drools.common.DefaultAgenda.fireActivation(DefaultAgenda.java554)
org.drools.common.DefaultAgenda.fireNextItem(DefaultAgenda.java518)
org.drools.common.AbstractWorkingMemory.fireAllRules(AbstractWorkingMemory.java475)
org.drools.common.AbstractWorkingMemory.fireAllRules(AbstractWorkingMemory.java439)
com.broadhop.policy.impl.RulesPolicyService.internalProcessRules(RulesPolicyService.java358)
com.broadhop.policy.impl.RulesPolicyService.process(RulesPolicyService.java210)
com.broadhop.policy.impl.RulesPolicyService$PolicyExecutionRunnable.run(RulesPolicyService.java793)

```

```

java.util.concurrent.Executors$RunnableAdapter.call(Unknown Source)
java.util.concurrent.FutureTask$Sync.innerRun(Unknown Source)
java.util.concurrent.FutureTask.run(Unknown Source)
java.util.concurrent.ThreadPoolExecutor$Worker.runTask(Unknown Source)
java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
java.lang.Thread.run(Unknown Source)</errorMessage>
</ExecuteActionResponse>
</se:Body>
</se:Envelope>

```

Validate One Time Code Success

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ExecuteActionResponse>
  </se:Body>
</se:Envelope>

```

Validate One Time Code Failure

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExecuteActionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>2</errorCode>
      <errorMessage>com.broadhop.exception.BroadhopException: OTC001: One time code
does not exist for user or code is invalid
com.broadhop.spr.impl.actions.ValidateOneTimeCode.execute (ValidateOneTimeCode.java57)
com.broadhop.policy.impl.RulesPolicyService.processSyncAction (RulesPolicyService.java486)
com.broadhop.policy.Rule_Validate_Code_247e55a8_1da4_4bf9_b095_d826bf53748d_yHv0LaBQeGrGeuFX0ls5g_0.consequence(Unknown
Source)
com.broadhop.policy.Rule_Validate_Code_247e55a8_1da4_4bf9_b095_d826bf53748d_yHv0LaBQeGrGeuFX0ls5g_0ConsequenceInvoker.evaluate(Unknown
Source)
org.drools.common.DefaultAgenda.fireActivation (DefaultAgenda.java554)
org.drools.common.DefaultAgenda.fireNextItem (DefaultAgenda.java518)
org.drools.common.AbstractWorkingMemory.fireAllRules (AbstractWorkingMemory.java475)
org.drools.common.AbstractWorkingMemory.fireAllRules (AbstractWorkingMemory.java439)
com.broadhop.policy.impl.RulesPolicyService.internalProcessRules (RulesPolicyService.java358)
com.broadhop.policy.impl.RulesPolicyService.process (RulesPolicyService.java210)
com.broadhop.policy.impl.RulesPolicyService$PolicyExecutionRunnable.run (RulesPolicyService.java793)
java.util.concurrent.Executors$RunnableAdapter.call (Unknown Source)
java.util.concurrent.FutureTask$Sync.innerRun (Unknown Source)
java.util.concurrent.FutureTask.run (Unknown Source)
java.util.concurrent.ThreadPoolExecutor$Worker.runTask (Unknown Source)
java.util.concurrent.ThreadPoolExecutor$Worker.run (Unknown Source)
java.lang.Thread.run (Unknown Source)</errorMessage>
</ExecuteActionResponse>
</se:Body>
</se:Envelope>

```

## ExtendCreditRequest

This API extends (modifies) a credit by moving the expiration date into the future and/or adding to the amount of credit available.

The resulting modified credit does not account for any debits. For example in the case of add = true, if the original credit amount is 100 and 50 is debited, and it is then extended by 100, the resulting credit amount will be 200, and the available amount will be 150. For example in the case of add = false (meaning replace),



if the original credit amount is 100 and 50 is debited, and it is changed with a replacement amount of 300, the resulting credit amount will be 300, and the available amount will be 250.




---

**Note One Time Quota Only!**

ExtendCredit API only works for One Time Quota NOT Recurring.

---




---

**Note What This API Does Not Do**

This API does not modify the start date, move the end date to an earlier date, or reduce of the amount of quota. Changes of these types will cause system instability and will almost always invalidate active reservations.

---

**Schema**

```
<ExtendCreditRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <extendCredit> ExtendCreditType </extendCredit> [1..10]
</ExtendCreditRequest>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExtendCreditRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <extendCredit>
        <balanceCode>DATA</balanceCode>
        <quotaCode>QUOTA</quotaCode>
        <creditId>_Dsdggsdw01282323jd</creditId>
        <add>false</add>
        <endDate>2013-01-31T00:00:00Z</endDate>
        <amount>500</amount>
      </extendCredit>
    </ExtendCreditRequest>
  </se:Body>
</se:Envelope>
```

## ExtendCreditResponse




---

**Note Amount Credited**

The amountCredited field is overloaded because the API can do an add or a replace. If the API has performed a replace the amountCredited will show the new total amount of credit which should match the passed in value. If the API has performed an add the amountCredited will show the new total amount of credit which is not the amount that was added (the value passed into the API).

---

## Schema

```
<ExtendCreditResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <returnCredit> ReturnCreditType </returnCredit> [1..10]
</ExtendCreditResponse>
```

## Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ExtendCreditResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <returnCredit>
        <id>_wertDSD1234sfsdge5657yfc</id>
        <nextRefreshDate>2012-11-28T00:00:00Z</nextRefreshDate>
        <balanceRemaining>250</balanceRemaining>
        <amountCredited>50</amountCredited>
        <callbackValidityTime>2012-11-28T00:00:00Z</callbackValidityTime>
      </returnCredit>
    </ExtendCreditResponse>
  </se:Body>
</se:Envelope>
```

# GenerateVoucherBatchRequest

This API generates a batch of vouchers and adds them to the system.

The API takes a sample voucher object and generates a set of voucher instances based on that sample. The code and pin values are 'mask' values which define a range of values that can be used for each instance.

See the code and pin fields in the VoucherType for more information.

## Schema

```
<GenerateVoucherBatchRequest>
  <audit> AuditType </audit> [0..1] ?
  <voucher> VoucherType </voucher> [1]
  <batchAmount> xsd:integer (1 >= value <= 1000) </batchAmount> [1]
  <maskChars> xsd:string </maskChars> [0..1] ?
</GenerateVoucherBatchRequest>
```

## Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GenerateVoucherBatchRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <voucher>
        <code>codeMask****</code>
        <pin>pinMask####</pin>
        <maxConcurrentSessions>5</maxConcurrentSessions>
        <duration>10</duration>
        <durationMeasure>Minutes</durationMeasure>
        <serviceCode>serviceCode</serviceCode>
        <expirationDate>2012-01-01T00:00:00Z</expirationDate>
        <locationCode>location</locationCode>
      </voucher>
    </GenerateVoucherBatchRequest>
  </se:Body>
</se:Envelope>
```

```

        <timeQuota>10</timeQuota>
        <timeMeasure>timemeasure</timeMeasure>
        <balanceCode>balanceCode</balanceCode>
        <quotaCode>quotaCode</quotaCode>
    </voucher>
    <batchAmount>100</batchAmount>
    <maskChars>abcd1234</maskChars>
</GenerateVoucherBatchRequest>
</se:Body>
</se:Envelope>

```

## GenerateVoucherBatchResponse

### Schema

```

<GenerateVoucherBatchResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</GenerateVoucherBatchResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GenerateVoucherBatchResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </GenerateVoucherBatchResponse>
  </se:Body>
</se:Envelope>

```

## GenericErrorResponse

This response is sent back for all known errors related to the function and health of the API server itself rather than errors returning from the operation of the APIs.

### Schema

```

<GenericErrorResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</GenericErrorResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GenericErrorResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>6</errorCode>
      <errorMessage>Invalid Request: CannotResolveClassException</errorMessage>
    </GenericErrorResponse>
  </se:Body>
</se:Envelope>

```

## GetRefDataBalanceRequest

This API retrieves a list of the balance templates, quota templates, and associated ref data thresholds and the values defined on the templates in the reference data.

**Schema**

```
<GetRefDataBalanceRequest>
  <audit> AuditType </audit> [0..1] ?
</GetRefDataBalanceRequest>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetRefDataBalanceRequest xmlns="http://broadhop.com/unifiedapi/soap/types"/>
  </se:Body>
</se:Envelope>
```

## GetRefDataBalanceResponse

**Schema**

```
<GetRefDataBalanceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <refDataBalanceTemplate> RefDataBalanceTemplateType </refDataBalanceTemplate> [0..1000]
</GetRefDataBalanceResponse>
```

**Example**

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetRefDataBalanceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <refDataBalanceTemplate>
        <code>VOUCHER-TIME</code>
        <description/>
        <quotaUnits>Second</quotaUnits>
        <refDataQuotaTemplate>
          <code>VOUCHER</code>
          <amount>0</amount>
        </refDataQuotaTemplate>
      </refDataBalanceTemplate>
      <refDataBalanceTemplate>
        <code>VOUCHER-DATA</code>
        <description/>
        <quotaUnits>Byte</quotaUnits>
        <refDataQuotaTemplate>
          <code>VOUCHER</code>
          <amount>0</amount>
        </refDataQuotaTemplate>
      </refDataBalanceTemplate>
      <refDataBalanceTemplate>
        <code>QNS_DATA</code>
        <description>Default data balance</description>
        <quotaUnits>Byte</quotaUnits>
        <refDataQuotaTemplate>
          <code>QNS_DATA</code>
          <amount>0</amount>
        </refDataQuotaTemplate>
      </refDataBalanceTemplate>
      <refDataBalanceTemplate>
        <code>QNS_TIME</code>
        <description/>
        <quotaUnits>Second</quotaUnits>
```

```

        <refDataQuotaTemplate>
          <code>QNS_TIME</code>
          <amount>0</amount>
        </refDataQuotaTemplate>
      </refDataBalanceTemplate>
    <refDataBalanceTemplate>
      <code>DATA</code>
      <description/>
      <quotaUnits>Byte</quotaUnits>
      <refDataQuotaTemplate>
        <code>onetime</code>
        <amount>100</amount>
        <refDataThreshold>
          <code>50Percent</code>
          <amount>50</amount>
          <thresholdType>Percentage</thresholdType>
          <triggerOnRemaining>false</triggerOnRemaining>
        </refDataThreshold>
      </refDataQuotaTemplate>
      <refDataQuotaTemplate>
        <description>monthly quota</description>
        <code>monthly</code>
        <amount>100</amount>
        <refDataThreshold>
          <code>25Percent</code>
          <amount>25</amount>
          <thresholdType>Percentage</thresholdType>
          <triggerOnRemaining>true</triggerOnRemaining>
        </refDataThreshold>
      </refDataQuotaTemplate>
    </refDataBalanceTemplate>
  </GetRefDataBalanceResponse>
</se:Body>
</se:Envelope>

```

## GetRefDataServicesRequest

This API retrieves a list of the services and their top level attributes defined in the reference data.

### Schema

```

<GetRefDataServicesRequest>
  <audit> AuditType </audit> [0..1] ?
</GetRefDataServicesRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetRefDataServicesRequest xmlns="http://broadhop.com/unifiedapi/soap/types"/>
  </se:Body>
</se:Envelope>

```

## GetRefDataServicesResponse

### Schema

```

<GetRefDataServicesResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <refDataService> RefDataServiceType </refDataService> [0..100]

```

```
</GetRefDataServicesResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetRefDataServicesResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <refDataService>
        <name>100 Megabit Service</name>
        <code>100Mbit</code>
        <addToSubAccounts>false</addToSubAccounts>
        <balanceService>true</balanceService>
        <enabled>true</enabled>
        <suppressInPortal>true</suppressInPortal>
      </refDataService>
      <refDataService>
        <name>default</name>
        <code>myservice</code>
        <addToSubAccounts>false</addToSubAccounts>
        <balanceService>true</balanceService>
        <enabled>true</enabled>
        <suppressInPortal>true</suppressInPortal>
      </refDataService>
    </GetRefDataServicesResponse>
  </se:Body>
</se:Envelope>
```

## GetSubscriberCountRequest

This API retrieves the number of subscribers in the USuM database.

### Schema

```
<GetSubscriberCountRequest>
  <audit> AuditType </audit> [0..1] ?
</GetSubscriberCountRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetSubscriberCountRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
    </GetSubscriberCountRequest>
  </se:Body>
</se:Envelope>
```

## GetSubscriberCountResponse

### Schema

```
<GetSubscriberCountResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <count> xsd:integer </count> [1]
```

```
</GetSubscriberCountResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetSubscriberCountResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <count>450000</count>
    </GetSubscriberCountResponse>
  </se:Body>
</se:Envelope>
```

## GetSubscriberRequest

This API retrieves a subscriber from the USuM database.



### Note Success Response

The GetSubscriber API returns error code 0 (success) even if no subscriber is found.

See the QueryBalance request for information about the boolean options: includeExpiredData and excludeReservationsFromCreditTotal

### Schema

```
<GetSubscriberRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <returnSessions> xsd:boolean </returnSessions> [0..1] ?
  <returnBalances> xsd:boolean </returnBalances> [0..1] ?
  <includeExpiredData> xsd:boolean </includeExpiredData> [0..1] ?
  <excludeReservationsFromCreditTotal> xsd:boolean </excludeReservationsFromCreditTotal>
  [0..1] ?
</GetSubscriberRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetSubscriberRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <returnSessions>>false</returnSessions>
      <returnBalances>>true</returnBalances>
      <includeExpiredData>>false</includeExpiredData>
      <excludeReservationsFromCreditTotal>>true</excludeReservationsFromCreditTotal>
    </GetSubscriberRequest>
  </se:Body>
</se:Envelope>
```

## GetSubscriberResponse

### Schema

```
<GetSubscriberResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <subscriber> SubscriberType </subscriber> [0..1]
</GetSubscriberResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <subscriber>
        ...
      </subscriber>
    </GetSubscriberResponse>
  </se:Body>
</se:Envelope>
```

## GetSubscriberSsidsRequest

This API retrieves a subscriber from the USuM database.



### Note Success Response

The GetSubscriberSsids API returns error code 0 (success) even if no subscriber is found.

### Schema

```
<GetSubscriberSsidsRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
</GetSubscriberSsidsRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetSubscriberSsidsRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
    </GetSubscriberSsidsRequest>
  </se:Body>
</se:Envelope>
```

## GetSubscriberSsidsResponse

### Schema



```

<GetSubscriberSsidsResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <subscriber> SubscriberSSIDType </subscriber> [0..1]
</GetSubscriberSsidsResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <GetSubscriberSsidsResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <subscriberSsids>
        ...
      </subscriberSsids>
    </GetSubscriberSsidsResponse>
  </se:Body>
</se:Envelope>

```

## KeepAliveRequest

This API verifies that the server is active. It does not require the server to do any processing and therefore does not increase load.

### Schema

```
<KeepAliveRequest/>
```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <KeepAliveRequest xmlns="http://broadhop.com/unifiedapi/soap/types"/>
  </se:Body>
</se:Envelope>

```

## KeepAliveResponse

### Schema

```

<KeepAliveResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</KeepAliveResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <KeepAliveResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </KeepAliveResponse>
  </se:Body>
</se:Envelope>

```

## ProvisionServiceRequest

This API provisions service definitions and corresponding Balance entries for a given subscriber.

The new Balances follow existing Balance rules - meaning if new, then provision, but if the Balance already exists, then just add a new credit.

If the service code already exists for the Subscriber, the service is updated with the new data from the API.




---

### Note Update = Replace

If the service already exists for the Subscriber, the service is updated by replacing the existing data with the new data from the API. It is NOT an incremental update, it is a complete replacement of the object. For example, if the existing service object has an AVP stored on it, and the new service data from the API does not, the resulting service object will not have the AVP.

---




---

### Note Automatic Balance Provisioning

This feature allows Service Options to be configured so that when the Service is applied to the Subscriber, Balance is automatically provisioned. (It also removes Balance when a Service is removed.) If the system is setup for auto-provisioning, do not include the Balance object when using this API. That is, only send in the Balance object when the system is not setup for auto-provisioning.

---

### Schema

```
<ProvisionServiceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <service> ServiceType </service> [1..10]
  <balance> CreateBalanceType </balance> [0..10]
</ProvisionServiceRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ProvisionServiceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <service>
        <code>SERVICE_CODE</code>
        <enabled>true</enabled>
        <avp>
          <code>AVP_CODE</code>
          <value>AVP_VALUE</value>
        </avp>
        <schedule>
          <startDate>2011-01-01T00:00:00Z</startDate>
          <endDate>2012-01-01T00:00:00Z</endDate>
          <state>ON</state>
          <startTime>00:00</startTime>
          <endTime>23:59</endTime>
        </schedule>
      </service>
    </ProvisionServiceRequest>
  </se:Body>
</se:Envelope>
```

```

        <repeat>
          <dayOfMonth>*</dayOfMonth>
          <month>*</month>
          <dayOfWeek>?</dayOfWeek>
          <year>*</year>
        </repeat>
        <enabled>>true</enabled>
      </schedule>
    </service>
  <balance>
    <code>DATA</code>
    <quotaCode>Recurring</quotaCode>
    <startDate>2011-01-01T00:00:00Z</startDate>
    <expirationDate>2012-01-01T00:00:00Z</expirationDate>
    <initialAmount>500</initialAmount>
  </balance>
</ProvisionServiceRequest>
</se:Body>
</se:Envelope>

```

## ProvisionServiceResponse

### Schema

```

<ProvisionServiceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</ProvisionServiceResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <ProvisionServiceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </ProvisionServiceResponse>
  </se:Body>
</se:Envelope>

```

## PurgeAuditHistoryRequest

This API purges the Audit History. See [Audit History](#) for more information.

The query is very flexible - it uses regex automatically for the id and dataid, and only one of the following are required: id, dataid, or request. The dataid element typically will be the networkId (Credential) value of a subscriber.

The id element is the person or application who made the API request. For example, if a CSR logs into Control Center and queries a subscriber balance, the id will be that CSR's username.

The dataid element is typically the subscriber's username. For example, if a CSR logs into Control Center and queries a subscriber, the id will be that CSR's username, and the dataid will be the subscriber's credential (networkId value). For queries, the dataid value is checked for spaces and then tokenized and each word is used as a search parameter. For example, "networkId1 networkId2" is interpreted as two values to check.

The fromDate represents the date in the past from which to start the purge or query. If the date is null, the api starts at the oldest entry in the history.

The toDate represents the date in the past to which the purge or query of data includes. If the date is null, the api includes the most recent entry in the purge or query.



#### Note Size-Capped Database

If the database is capped by size, then the purge request ignores the request key values and drops the entire database due to restrictions of the database software.

#### Schema

```
<PurgeAuditHistoryRequest>
  <key> AuditKeyType </key> [1]
</PurgeAuditHistoryRequest>
```

#### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <PurgeAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <id>username</id>
        <dataid>subscriber</dataid>
        <request>API Name</request>
        <fromDate>2011-01-01T00:00:00Z</fromDate>
        <toDate>2011-01-01T00:00:00Z</toDate>
      </key>
    </PurgeAuditHistoryRequest>
  </se:Body>
</se:Envelope>
```

To purge all CreateSubscriberRequest:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <PurgeAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <request>CreateSubscriberRequest</request>
      </key>
    </PurgeAuditHistoryRequest>
  </se:Body>
</se:Envelope>
```

To purge all CreateSubscriberRequest by CSR:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <PurgeAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <id>csrusername</id>
        <request>CreateSubscriberRequest</request>
      </key>
    </PurgeAuditHistoryRequest>
  </se:Body>
</se:Envelope>
```

To purge all actions by CSR for a given subscriber for a date range:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <PurgeAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
```

```

    <key>
      <id>csrusername</id>
      <dataid>subscriber@gmail.com</dataid>
      <fromDate>2010-01-01T00:00:00Z</fromDate>
      <toDate>2012-11-01T00:00:00Z</toDate>
    </key>
  </PurgeAuditHistoryRequest>
</se:Body>
</se:Envelope>

```

## PurgeAuditHistoryResponse

### Schema

```

<PurgeAuditHistoryResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</PurgeAuditHistoryResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <PurgeAuditHistoryResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </PurgeAuditHistoryResponse>
  </se:Body>
</se:Envelope>

```

## QueryAuditHistoryRequest

This API queries the Audit History. See [Audit History](#) for more information.

The query is very flexible - it uses regex automatically for the id and dataid, and only one of the following are required: id, dataid, or request. The dataid element typically will be the networkId (Credential) value of a subscriber.

The id element is the person or application who made the API request. For example, if a CSR logs into Control Center and queries a subscriber balance, the id will be that CSR's username.

The dataid element is typically the subscriber's username. For example, if a CSR logs into Control Center and queries a subscriber, the id will be that CSR's username, and the dataid will be the subscriber's credential (networkId value). For queries, the dataid value is checked for spaces and then tokenized and each word is used as a search parameter. For example, "networkId1 networkId2" is interpreted as two values to check.

The fromDate represents the date in the past from which to start the purge or query. If the date is null, the api starts at the oldest entry in the history.

The toDate represents the date in the past to which the purge or query of data includes. If the date is null, the api includes the most recent entry in the purge or query.

### Schema

```

<QueryAuditHistoryRequest>
  <key> AuditKeyType </key> [1]
</QueryAuditHistoryRequest>

```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <id>username</id>
        <dataid>subscriber</dataid>
        <request>API Name</request>
        <fromDate>2011-01-01T00:00:00Z</fromDate>
        <toDate>2011-01-01T00:00:00Z</toDate>
      </key>
    </QueryAuditHistoryRequest>
  </se:Body>
</se:Envelope>
```

To find all CreateSubscriberRequest:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <request>CreateSubscriberRequest</request>
      </key>
    </QueryAuditHistoryRequest>
  </se:Body>
</se:Envelope>
```

To find all CreateSubscriberRequest by CSR:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <id>csrusername</id>
        <request>CreateSubscriberRequest</request>
      </key>
    </QueryAuditHistoryRequest>
  </se:Body>
</se:Envelope>
```

To find all actions by CSR for a given subscriber for a date range:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryAuditHistoryRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <key>
        <id>csrusername</id>
        <dataid>subscriber@gmail.com</dataid>
        <fromDate>2010-01-01T00:00:00Z</fromDate>
        <toDate>2012-11-01T00:00:00Z</toDate>
      </key>
    </QueryAuditHistoryRequest>
  </se:Body>
</se:Envelope>
```

## QueryAuditHistoryResponse

### Schema

```
<QueryAuditHistoryResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <audit> AuditType </audit> [0..1000]
```

```
</QueryAuditHistoryResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryAuditHistoryResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <audit>
      </audit>
    </QueryAuditHistoryResponse>
  </se:Body>
</se:Envelope>
```

## QueryBalanceRequest

This API retrieves a subscriber's balance.

This API replaces the old MsBM QuerySubscriber API. The set of boolean options in the old API have been condensed to 2 booleans: includeExpiredData and excludeReservationsFromCreditTotal.

- includeExpireData: if true, returns all account balances whether they are currently active or deprovisioned. Also returns all credits including expired credits.
- excludeReservationsFromCreditTotal: if true, returns credit totals without reservation data being calculated into the totals.

### New Parameters to Old Parameters

current	old
includeExpiredData	showDeprovisionedBalances, showAllCreditsInDetail
excludeReservationsFromCreditTotal	doNotDebitReservedAmtFromCreditElementAmt
	showDetailedInformation, showNextRefreshListInDetail, and showDefinedQuotaCodeListInDetail are always set to true

### Old Parameters Behavior

- showDetailedInformation: if true, it will return lists of the currently valid credits for each account balance.
- showAllCreditsInDetail: if true then all credits, including expired credits will be returned. Only valid when showDetailedInformation is set to true.
- showDefinedQuotaCodeListInDetail: if true, displays of a list of all quota codes whether they have currently valid credits or not. Only valid when showDetailedInformation is set to true.
- showDeprovisionedBalances: if true, displays all account balances of the subscriber whether they are currently active or deprovisioned. Normally only active account balances are displayed.

- `doNotDebitReservedAmtFromCreditElementAmt`: if true, then credits listed do not have the amount of pending reservations deducted from the credit amount, only actually debited amounts are used in the calculation. Reservation amounts are listed separately.
- `showNextRefreshListInDetail`: if true, then a list of the next refresh dates for recurring quotas will be returned. Only valid when `showDetailedInformation` is set to true.

### Auto Rollover

For Recurring Quota that has Auto Rollover setup, there are additional values that get returned with the Rollover Credit objects. These fields are there to assist portals display the proper values for when credit expires. For example, the Recurring Quota refreshes each month, but the Rollover is valid for 2 months from the time of rollover which means that the credit is valid for a total of 3 months. Technically according to the system, there is a credit associated to the Recurring Quota valid for 1 month and then a credit associated to the Rollover Quota that is valid for 2 months. However, for the subscriber, credit is valid for 3 months. To properly display when the rollover credit expires, the response must include the template data.

- `rolloverPeriodAmount`
- `rolloverPeriodUnits`
- `rolloverExpirationDate`
- `rolloverTemplateName`

### Schema

```
<QueryBalanceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <includeExpiredData> xsd:boolean </includeExpiredData> [0..1]
  <excludeReservationsFromCreditTotal> xsd:boolean </excludeReservationsFromCreditTotal>
  [0..1]
</QueryBalanceRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryBalanceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <includeExpiredData>true</includeExpiredData>
      <excludeReservationsFromCreditTotal>true</excludeReservationsFromCreditTotal>
    </QueryBalanceRequest>
  </se:Body>
</se:Envelope>
```

## QueryBalanceResponse

### Schema

```
<QueryBalanceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
```



```

    <balance> BalanceType </balance> [0..100]
  </QueryBalanceResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryBalanceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <balance>
      </balance>
    </QueryBalanceResponse>
  </se:Body>
</se:Envelope>

```

## QuerySessionRequest

This API retrieves session data from active sessions in the Policy Engine session cache. Because of the flexible nature of CPS configuration, session data may differ for every installation/deployment. Therefore the response XML from a QuerySession request simply defines a map that can contain any of the following objects: Map, List, String, Boolean, Long, Integer, and Byte-Array.

In addition to the examples below, there may be the following keys stored on the session:

- FramedIpKey:framedIp:VALUE - for example, FramedIpKey:framedIp:173.251.37.147
- MacAddressKey:macAddress:VALUE - for example, MacAddressKey:macAddress:0A2B.3C4D.5E6F
- ciscoAccountInfo:NASIpVALUE:FramedIpVALUE - for example, ciscoAccountInfo:167.206.20.1:S173.251.37.147

### Schema

```

<QuerySessionRequest>
  <audit> AuditType </audit> [0..1] ?
  <key> SessionKeyType </key> [1]
  <retrieveAll> xsd:boolean </retrieveAll> [0..1] ?
</QuerySessionRequest>

```

### Example

Example - UserIdKey

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QuerySessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <key>
        <code>UserIdKey</code>
        <primary>>false</primary>
        <keyField>
          <code>userId</code>
          <value>username</value>
        </keyField>
      </key>
    </QuerySessionRequest>
  </se:Body>
</se:Envelope>

```

```

    </se:Body>
  </se:Envelope>

```

Example - USuMSubscriberIdKey

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QuerySessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <key>
        <code>USuMSubscriberIdKey</code>
        <primary>>false</primary>
        <keyField>
          <code>usumSubscriberId</code>
          <value>12zzcvasdfqwer234qwfv4fwe</value>
        </keyField>
      </key>
    </QuerySessionRequest>
  </se:Body>
</se:Envelope>

```

Example - USuMCredentialKey

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QuerySessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <key>
        <code>USuMCredentialKey</code>
        <primary>>false</primary>
        <keyField>
          <code>networkId</code>
          <value>mediatest320@optimum.net</value>
        </keyField>
      </key>
    </QuerySessionRequest>
  </se:Body>
</se:Envelope>

```

Example - FramedIpKey

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QuerySessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <key>
        <code>FramedIpKey</code>
        <primary>>false</primary>
        <keyField>
          <code>framedIp</code>
          <value>172.0.0.1</value>
        </keyField>
      </key>
    </QuerySessionRequest>
  </se:Body>
</se:Envelope>

```

```

</se:Body>
</se:Envelope>

```

## QuerySessionResponse



**Note** RADIUS-based policy control is no longer supported in CPS 14.0.0 and later releases as 3GPP Gx Diameter interface has become the industry-standard policy control interface.

### Schema

```

<QuerySessionResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <session> SessionType </session> [0..500]
</QuerySessionResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QuerySessionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <session>
        <sessionKey>
          <code>accountSessionId</code>
          <primary>true</primary>
          <keyField>
            <code>accountSessionId</code>
            <value>167.206.20.1:A7CE14420000163A</value>
          </keyField>
        </sessionKey>
        <sessionObject>
          <entry>
            <string>tags</string>
            <list>
              <string>USuMSubscriberIdKey:usumSubscriberId:50b51183e4b0b2dcee401c47</string>

              <string>USuMCredentialKey:networkId:emediatest302%40optimum.net</string>
              <string>UserIdKey:userId:emediatest302%40optimum.net</string>
              <string>MacAddressKey:macAddress:ac811233519f</string>
              <string>ciscoAccountInfo:167.206.20.1:S173.251.37.147</string>
              <string>ciscoAccountInfo:173.251.37.147:S173.251.37.147</string>
              <string>FramedIpKey:framedIp:173.251.37.147</string>
            </list>
          </entry>
          <entry>
            <string>avps</string>
            <list>
              <map>
                <entry>
                  <string>_type</string>
                  <string>Avp</string>
                </entry>
                <entry>
                  <string>value</string>
                  <string>AWIFILOC1</string>
                </entry>
              </map>
            </list>
          </entry>
        </sessionObject>
      </session>
    </QuerySessionResponse>
  </se:Body>
</se:Envelope>

```

```

        <string>code</string>
        <string>CISCO-ACCOUNT-INFO</string>
    </entry>
</map>
<map>
    <entry>
        <string>_type</string>
        <string>Avp</string>
    </entry>
    <entry>
        <string>value</string>
        <string>75B4EAEF5AAF060585904F3E7FE09C80</string>
    </entry>
    <entry>
        <string>code</string>
        <string>MESSAGE-AUTHENTICATOR</string>
    </entry>
</map>
</list>
</entry>
<entry>
    <string>_transId</string>
    <string>ddb708c9-8138-4499-adbf-73db195fce64-1</string>
</entry>
<entry>
    <string>macAddress</string>
    <string>ac811233519f</string>
</entry>
<entry>
    <string>deviceSessions</string>
    <list>
        <map>
            <entry>
                <string>deviceService</string>
                <string>WIFILOC1</string>
            </entry>
            <entry>
                <string>_type</string>
    <string>com.broadhop.radius.impl.devicemanager.domain.RadiusUsageTracking</string>
    </entry>
    <entry>
        <string>inBytes</string>
        <long>103906</long>
    </entry>
    <entry>
        <string>sessionId</string>
        <string>A7CE1442000163D</string>
    </entry>
    <entry>
        <string>deviceManagerId</string>

<string>com.broadhop.radius.impl.devicemanager.RadiusReportingManager</string>
</entry>
<entry>
    <string>outBytes</string>
    <long>671971</long>
</entry>
</map>
<map>
    <entry>
        <string>locationCode</string>
        <string>AAA Proxy User</string>
    </entry>

```

```

    <entry>
      <string>_type</string>
      <string>com.broadhop.policy.authdomain.SessionAuthDomain</string>
    </entry>
  </entry>
  <entry>
    <string>talSession</string>
    <boolean>>false</boolean>
  </entry>
  <entry>
    <string>deviceManagerId</string>
    <string>com.broadhop.policy.authdomain.impl.DomainDeviceManager</string>
  </entry>
  <entry>
    <string>providerCode</string>
    <string>Cablevision Optimum.Net</string>
  </entry>
  <entry>
    <string>domainCode</string>
    <string>Cablevision</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>com.broadhop.policy.authdomain.impl.domain.AuthorizedSession</string>
  </entry>
  <entry>
    <string>deviceManagerId</string>
    <string>com.broadhop.policy.authdomain.impl.DomainDeviceManager</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>com.broadhop.balance.impl.autowire.BalanceDeviceSession</string>
  </entry>
  <entry>
    <string>deviceManagerId</string>
    <string>com.broadhop.balance.impl.autowire.AutowireBalanceManagerBlueprint</string>
  </entry>
</map>
<map>
  <entry>
    <string>_type</string>
    <string>com.broadhop.radius.impl.devicemanager.domain.IsgDeviceSession</string>
  </entry>
  <entry>
    <string>services</string>
    <list>
      <map>
        <entry>
          <string>_type</string>
          <string>com.broadhop.radius.devices.IsgServiceSession</string>
        </entry>
        <entry>
          <string>sessionId</string>
          <string>A7CE14420000163D</string>
        </entry>
        <entry>
          <string>reactivationAllowed</string>
          <map>

```

```

        <entry>
          <string>_type</string>
          <string>java.util.Date</string>
        </entry>
        <entry>
          <string>time</string>
          <long>1359998144594</long>
        </entry>
      </map>
    </entry>
    <entry>
      <string>minReactWaitTime</string>
      <int>30</int>
    </entry>
    <entry>
      <string>active</string>
      <boolean>>false</boolean>
    </entry>
    <entry>
      <string>serviceCode</string>
      <string>WIFILOC1</string>
    </entry>
  </map>
</list>
</entry>
<entry>
  <string>keyType</string>
  <int>0</int>
</entry>
<entry>
  <string>radiusGroup</string>
  <string>ISG1</string>
</entry>
<entry>
  <string>deviceManagerId</string>
  <string>com.broadhop.radius.impl.devicemanager.IsgNetworkDeviceManager</string>
</entry>
<entry>
  <string>terminated</string>
  <boolean>>false</boolean>
</entry>
<entry>
  <string>accountInfo</string>
  <string>S173.251.37.147</string>
</entry>
<entry>
  <string>loopbackAddress</string>
  <string>173.251.37.147</string>
</entry>
<entry>
  <string>callingStationId</string>
  <string>ac811233519f</string>
</entry>
<entry>
  <string>sessionAvps</string>
  <list/>
</entry>
<entry>
  <string>radiusUserId</string>
  <string>emediatest302@optimum.net</string>
</entry>
<entry>
  <string>accountSessionId</string>

```

```

        <string>A7CE14420000163A</string>
      </entry>
    </entry>
    <string>ipAddress</string>
    <string>167.206.20.1</string>
  </entry>
</map>
</list>
</entry>
<entry>
  <string>credentialId</string>
  <string>emediatest302@optimum.net</string>
</entry>
<entry>
  <string>startTime</string>
  <date>2013-02-04T09:38:21.000-07:00</date>
</entry>
<entry>
  <string>_type</string>
  <string>com.broadhop.session.domain.NetworkSession</string>
</entry>
<entry>
  <string>_id</string>
  <map>
    <entry>
      <string>accountSessionId</string>
      <string>167.206.20.1:A7CE14420000163A</string>
    </entry>
  </map>
</entry>
<entry>
  <string>expirationTime</string>
  <date>2013-02-04T18:15:14.345-07:00</date>
</entry>
<entry>
  <string>framedIp</string>
  <string>173.251.37.147</string>
</entry>
<entry>
  <string>userId</string>
  <string>emediatest302@optimum.net</string>
</entry>
<entry>
  <string>nextEvalTime</string>
  <date>2013-02-04T18:15:14.345-07:00</date>
</entry>
<entry>
  <string>countsAgainstMaxSessions</string>
  <boolean>true</boolean>
</entry>
<entry>
  <string>key</string>
  <map>
    <entry>
      <string>accountSessionId</string>
      <string>167.206.20.1:A7CE14420000163A</string>
    </entry>
  </map>
</entry>
</sessionObject>
</session>
</QuerySessionResponse>
</se:Body>
</se:Envelope>

```

## QueryVoucherRequest

This API retrieves vouchers.

### Schema

```
<QueryVoucherRequest>
  <audit> AuditType </audit> [0..1] ?
  <key> VoucherKeyType </key> [1]
</QueryVoucherRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryVoucherRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <key>
        <voucherLocationKey>
          <locationCode>location</locationCode>
          <active>true</active>
        </voucherLocationKey>
      </key>
    </QueryVoucherRequest>
  </se:Body>
</se:Envelope>
```

## QueryVoucherResponse

### Schema

```
<QueryVoucherResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <voucher> VoucherType </voucher> [0..1000]
</QueryVoucherResponse>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <QueryVoucherResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <voucher>
        <code>voucher</code>
        <pin>pin</pin>
        <maxConcurrentSessions>5</maxConcurrentSessions>
        <duration>10</duration>
        <durationMeasure>Minutes</durationMeasure>
        <serviceCode>serviceCode</serviceCode>
        <expirationDate>2012-01-01T00:00:00Z</expirationDate>
        <locationCode>location</locationCode>
        <timeQuota>10</timeQuota>
        <timeMeasure>timemeasure</timeMeasure>
        <balanceCode>balanceCode</balanceCode>
        <quotaCode>quotaCode</quotaCode>
      </voucher>
    </QueryVoucherResponse>
  </se:Body>
</se:Envelope>
```



```

    <voucher>
    ...
    </voucher>
  </QueryVoucherResponse>
</se:Body>
</se:Envelope>

```

## RedeemVoucherRequest

This API redeems a voucher for a subscriber. It marks the voucher as redeemed and sets up the subscriber with the appropriate services and quota based on the data stored in the voucher. The code and pin are used to match a specific voucher instance.

The code and pin are added as a credential to the subscriber.



### Note No New Subscriber

This API does not create a new subscriber unless the subscriber element is included in the request. This means that the api assumes the subscriber already exists in the SPR database unless there is a subscriber element included in the request. If the subscriber element is included in the request, the API attempts to create the subscriber and will error if the credential already exists in the SPR database.

### Schema

```

<RedeemVoucherRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <code> xsd:string </code> [1]
  <pin> xsd:string </pin> [1]
  <subscriber> SubscriberType </subscriber> [0..1]
</RedeemVoucherRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <RedeemVoucherRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <code>voucher</code>
      <pin>voucher</pin>
      <subscriber>
        ...
      </subscriber>
    </RedeemVoucherRequest>
  </se:Body>
</se:Envelope>

```

## RedeemVoucherResponse

### Schema

```

<RedeemVoucherResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</RedeemVoucherResponse>

```

**Example**

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <RedeemVoucherResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </RedeemVoucherResponse>
  </se:Body>
</se:Envelope>

```

## RemoveSubscriberSsidRequest

This API removes a subscriber SSID from the SPR extension collection.

**Schema**

```

<RemoveSubscriberSsidRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <ssidKey> xsd:string </ssidKey> [0..1]
</RemoveSubscriberSsidRequest>

```

**Example**

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <RemoveSubscriberSsidRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <ssidKey>keyOfSSID</ssidKey>
    </RemoveSubscriberSsidRequest>
  </se:Body>
</se:Envelope>

```

## RemoveSubscriberSsidRequest

**Schema**

```

<RemoveSubscriberSsidResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</RemoveSubscriberSsidResponse>

```

**Example**

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <RemoveSubscriberSsidResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </RemoveSubscriberSsidResponse>
  </se:Body>
</se:Envelope>

```

```

</se:Body>
</se:Envelope>

```

## RolloverCreditRequest

This API operates the same as the old MsBM API. It marks a current credit to be rolled over on the next refresh date, and immediately rolls over an expired credit. It can do this even for recurring quota types that are not specifically set to rollover quota for other subscribers.



### Note Recurring Quota

Please note that a Rollover Quota must be specified on the Recurring Quota template in order for this API to work.

### Schema

```

<RolloverCreditRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <balanceCode> xsd:string </balanceCode> [1] ?
  <quotaCode> xsd:string </quotaCode> [1] ?
  <creditId> xsd:string </creditId> [1] ?
  <rollover> xsd:boolean </rollover> [1] ?
</RolloverCreditRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <RolloverCreditRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <balanceCode>DATA</balanceCode>
      <quotaCode>Recurring</quotaCode>
      <creditId>1</creditId>
      <rollover>true</rollover>
    </RolloverCreditRequest>
  </se:Body>
</se:Envelope>

```

## RolloverCreditResponse

### Schema

```

<RolloverCreditResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</RolloverCreditResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <RolloverCreditResponse xmlns="http://broadhop.com/unifiedapi/soap/types">

```

```

        <errorCode>0</errorCode>
        <errorMessage>Request completed successfully</errorMessage>
    </RolloverCreditResponse>
</se:Body>
</se:Envelope>

```

## SearchSubscribersRequest

This API retrieves all Subscribers that match the given search criteria (filter element).

Searches using the pre-defined filter options are "fuzzy" by default, meaning that the search will return values that are like matches and not just exact matches.

See [Custom Search Params](#) for information about using custom search params rather than the pre-defined filter options.




---

### Note Plugin Configuration

Fuzzy or Regex searching is enabled by default in the Unified API Plugin Configuration. AVPs used in the custom search params can also use the same regex algorithm as the pre-defined filters and must be enabled in the plugin configuration. (AVP regex is disabled by default.) Be careful because AVP regex searching can impact performance.




---

### Note Fuzzy Matching Caveats

Do not include regular expressions in the search values. The code is written to do a case-insensitive regex match in the following manner: `/^value/` This regex means to match values that start (^) with the same characters. For example, if you send 'rob' as the credential value in the request, the API will match rob, Robert or ROBERTA. If you send '123' as the credential value in the request, the API will match 123, 1234, etc.

---

### Schema

```

<SearchSubscribersRequest>
  <audit> AuditType </audit> [0..1] ?
  <filter> SearchType </filter> [1]
  <returnSessions> xsd:boolean </returnSessions> [0..1]
  <returnBalances> xsd:boolean </returnBalances> [0..1]
</SearchSubscribersRequest>

```

### Example

Example 1 - Name Search

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <SearchSubscribersRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <filter>
        <name>
          <fullName>Test Subscriber</fullName>
        </name>
      </filter>
    </SearchSubscribersRequest>
  </se:Body>
</se:Envelope>

```

```

        </filter>
    </SearchSubscribersRequest>
</se:Body>
</se:Envelope>

```

Example 2 - Name Search with balances and sessions

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <SearchSubscribersRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <filter>
        <name>
          <fullName>Test Subscriber</fullName>
        </name>
      </filter>
      <returnSessions>true</returnSessions>
      <returnBalances>true</returnBalances>
    </SearchSubscribersRequest>
  </se:Body>
</se:Envelope>

```

Example 3 - NetworkId Search

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <SearchSubscribersRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <filter>
        <networkId>uniqueId</networkId>
      </filter>
    </SearchSubscribersRequest>
  </se:Body>
</se:Envelope>

```

## SearchSubscribersResponse

### Schema

```

<SearchSubscribersResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <subscriber> SubscriberType </subscriber> [0..1000]
</SearchSubscribersResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <SearchSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <subscriber>
        ...
      </subscriber>
      <subscriber>

```

```

    ...
  </subscriber>
  ...
</SearchSubscribersResponse>
</se:Body>
</se:Envelope>

```

## StopSessionRequest

This API stops an active session on the network and discards the data from the Policy Engine session cache.

This API replaces the version 5.2 Policy Engine RemoveSession API.

Please see QuerySession for more information about keys to use.

### Schema

```

<StopSessionRequest>
  <audit> AuditType </audit> [0..1] ?
  <key> SessionKeyType </key> [1]
</StopSessionRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <StopSessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <key>
        <code>UserIdKey</code>
        <primary>true</primary>
        <keyField>
          <code>userId</code>
          <value>uniqueIndentifier</value>
        </keyField>
      </key>
    </StopSessionRequest>
  </se:Body>
</se:Envelope>

```

## StopSessionResponse

### Schema

```

<StopSessionResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</StopSessionResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <StopSessionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </StopSessionResponse>
  </se:Body>
</se:Envelope>

```

```
</se:Body>
</se:Envelope>
```

## SwitchServiceRequest

This API changes a service definition and corresponding Balance entry for a given subscriber to new ones.

The oldServiceCode value is used to find the oldBalanceCode if oldBalanceCode is not sent in. If no balance is found, the API then checks for the number of balances. If 1 balance is found, then it is deleted, if more than 1, then an error is thrown.

This API works with One Time or Recurring Quotas. The new quota is provisioned per the createBalance values. The lastRecurringRefresh date of the new quota is set per normal new provision rules. Old dates are not migrated and the validity period for the reservation stays the same.




---

### Note Automatic Balance Provisioning

This feature allows Service Options to be configured so that when the Service is applied to the Subscriber, Balance is automatically provisioned. (It also removes Balance when a Service is removed.) If the system is setup for auto-provisioning, do not include the Balance object when using this API. That is, only send in the Balance object when the system is not setup for auto-provisioning.

---




---

### Note Reservations are not migrated

Reservations are not migrated from the old Balance or Quota because Autowire code handles any latent charges.

---




---

### Note Quota Capacity

The new quota must have the same or more capacity than the old quota currently has available.

---

### Schema

```
<SwitchServiceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <oldServiceCode> xsd:string </oldServiceCode> [1] ?
  <service> ServiceType </service> [1]
  <oldBalanceCode> xsd:string </oldBalanceCode> [0..1]
  <oldQuotaCode> xsd:string </oldQuotaCode> [0..1] ?
  <balance> CreateBalanceType </balance> [0..1] ?
</SwitchServiceRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <SwitchServiceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
```

```

</audit>
<networkId>uniqueIdentifier</networkId>
<oldServiceCode>SERVICE_ORIGINAL</oldServiceCode>
<service>
  <code>SERVICE_CODE</code>
  <enabled>true</enabled>
  <avp>
    <code>AVP_CODE</code>
    <value>AVP_VALUE</value>
  </avp>
  <schedule>
    <startDate>2011-01-01T00:00:00Z</startDate>
    <endDate>2012-01-01T00:00:00Z</endDate>
    <state>ON</state>
    <startTime>00:00</startTime>
    <endTime>23:59</endTime>
    <repeat>
      <dayOfMonth>*</dayOfMonth>
      <month>*</month>
      <dayOfWeek>?</dayOfWeek>
      <year>*</year>
    </repeat>
    <enabled>true</enabled>
  </schedule>
</service>
<oldBalanceCode>DATA</oldBalanceCode>
<oldQuotaCode>OLD_QUOTA</oldQuotaCode>
<balance>
  <code>DATA</code>
  <quotaCode>RECURRING</quotaCode>
  <startDate>2011-01-01T00:00:00Z</startDate>
  <expirationDate>2012-01-01T00:00:00Z</expirationDate>
  <initialAmount>500</initialAmount>
</balance>
</SwitchServiceRequest>
</se:Body>
</se:Envelope>

```

## SwitchServiceResponse

### Schema

```

<SwitchServiceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</SwitchServiceResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <SwitchServiceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </SwitchServiceResponse>
  </se:Body>
</se:Envelope>

```



## UpdateBalanceRequest

This API updates non-calculated data fields on a subscriber's balance. Similar to [UpdateSubscriberRequest](#), on page 86 this API is not incremental; meaning an entire balance object must be sent.

This API replaces several old MsBM APIs.

- DeProvisionBalance
- ModifyAccountBalanceProvisionStatus
- CreateSubscriberThreshold
- RemoveSubscriberThreshold
- UpdateSubscriberThreshold

Only subscriber specific thresholds can be managed via this API. Any global thresholds configured in Policy Builder must be updated through the Policy Builder.




---

**Note Breached and Amount**

The amount field for a threshold is the amount at which the threshold gets triggered. If the threshold is in a breached state according to the Balance engine, the amount field cannot be updated!

---




---

**Note What This API Does NOT Do!**

This API will not handle ChangeRecurringRefresh or ChangeBillCycle or ChangeBalanceSubscriberId or DeleteQuota or DeleteCredit because UpdateBalance does not allow users to adjust Quota values manually. This would have dire consequences on the integrity of the data.

---




---

**Note Removed Elements**

The nextRefreshDate and avp elements in a threshold have been removed because they are not used by the Balance module.

---




---

**Note The proper steps to use this API:**

1. Call GetSubscriber or QueryBalance API to get the balance object.
  2. Update returned balance object as required.
  3. Call UpdateBalance API with updated balance object.
- 

**Schema**

```
<UpdateBalanceRequest>
  <audit> AuditType </audit> [0..1] ?
```

```

    <networkId> xsd:string </networkId> [1] ?
    <balance> BalanceType </balance> [1..100] ?
</UpdateBalanceRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateBalanceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <balance>
        <code>DATA</code>
        <deprovisioned>true</deprovisioned>
        <depleted>true</depleted>
        <startDate>2011-01-01T00:00:00Z</startDate>
        <expirationDate>2012-01-01T00:00:00Z</expirationDate>
        <quota>
          <code>TopUp</code>
          <threshold>
            <code>THRESHUPDATE</code>
            <amount>70</amount>
            <quotaCode>TopUp</quotaCode>
            <type>Percentage</type>
            <breached>>false</breached>
            <subscriberSpecific>>true</subscriberSpecific>
          </threshold>
        </quota>
        <threshold>
          <code>THRESHUPDATE</code>
          <amount>70</amount>
          <type>Percentage</type>
          <breached>>false</breached>
          <subscriberSpecific>>true</subscriberSpecific>
        </threshold>
      </balance>
    </UpdateBalanceRequest>
  </se:Body>
</se:Envelope>

```

## UpdateBalanceResponse

### Schema

```

<UpdateBalanceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</UpdateBalanceResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateBalanceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </UpdateBalanceResponse>
  </se:Body>
</se:Envelope>

```

## UpdateServiceRequest

This API updates a service definition and corresponding Balance entries for a given subscriber.

The order of operation for the AVPs is delete, modify, add (new). The modify avps only modify the value.




---

### Note Schedules

Because it is not possible to match an individual schedule object, the list is simply a replacement. If no list is present in the request, then nothing happens. However, if you send an empty element (<schedule/> or <schedule></schedule>), the list will get nulled out.

---



### Note Automatic Balance Provisioning

This feature allows Service Options to be configured so that when the Service is applied to the Subscriber, Balance is automatically provisioned. (It also removes Balance when a Service is removed.) If the system is setup for auto-provisioning, do not include the Balance object when using this API. That is, only send in the Balance object when the system is not setup for auto-provisioning.

---

### Schema

```
<UpdateServiceRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
  <code> xsd:string </code> [1]
  <enabled> xsd:boolean </enabled> [0..1]
  <deletedAvp> AvpType </deletedAvp> [0..100]
  <modifiedAvp> AvpType </modifiedAvp> [0..100]
  <newAvp> AvpType </newAvp> [0..100]
  <schedule> ScheduleType </schedule> [0..10]
  <balance> CreateBalanceType </balance> [0..10]
  <extendCredit> ExtendCreditType </extendCredit> [0..10]
</UpdateServiceRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateServiceRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <networkId>uniqueIdentifier</networkId>
      <code>SERVICE_CODE</code>
      <enabled>true</enabled>
      <deletedAvp>
        <code>AVP_CODE</code>
        <value>AVP_VALUE</value>
      </deletedAvp>
      <modifiedAvp>
        <code>AVP_CODE_2</code>
        <value>AVP_VALUE</value>
        <newValue>NEW_AVP_VALUE</newValue>
      </modifiedAvp>
```

```

<newAvp>
  <code>NEW_AVP_CODE</code>
  <value>NEW_AVF_VALUE</value>
</newAvp>
<schedule>
  <startDate>2011-01-01T00:00:00Z</startDate>
  <endDate>2012-01-01T00:00:00Z</endDate>
  <state>ON</state>
  <startTime>00:00</startTime>
  <endTime>23:59</endTime>
  <repeat>
    <dayOfMonth>*</dayOfMonth>
    <month>*</month>
    <dayOfWeek>?</dayOfWeek>
    <year>*</year>
  </repeat>
  <enabled>true</enabled>
</schedule>
<balance>
  <code>DATA</code>
  <quotaCode>RECURRING</quotaCode>
  <startDate>2011-01-01T00:00:00Z</startDate>
  <expirationDate>2012-01-01T00:00:00Z</expirationDate>
  <initialAmount>500</initialAmount>
</balance>
<extendCredit>
  <balanceCode>DATA</balanceCode>
  <quotaCode>QUOTA</quotaCode>
  <creditId>_Dsdggsdw01282323jd</creditId>
  <add>false</add>
  <endDate>2013-01-31T00:00:00Z</endDate>
  <amount>500</amount>
</extendCredit>
</UpdateServiceRequest>
</se:Body>
</se:Envelope>

```

## UpdateServiceResponse

### Schema

```

<UpdateServiceResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
  <returnCredit> ReturnCreditType </returnCredit> [0..10]
</UpdateServiceResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateServiceResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <returnCredit>
        <id>_wertDSD1234sfsdgs5657yfc</id>
        <nextRefreshDate>2012-11-28T00:00:00Z</nextRefreshDate>
        <balanceRemaining>250</balanceRemaining>
        <amountCredited>50</amountCredited>
        <callbackValidityTime>2012-11-28T00:00:00Z</callbackValidityTime>
      </returnCredit>
    </UpdateServiceResponse>
  </se:Body>
</se:Envelope>

```

```

</se:Body>
</se:Envelope>

```

## UpdateSessionRequest

This API updates an active session on the network and in the Policy Engine session cache. It removes or adds AVPs.

Please see QuerySession for more information about keys to use.

### Schema

```

<UpdateSessionRequest>
  <audit> AuditType </audit> [0..1] ?
  <key> SessionKeyType </key> [1]
  <newAvp> AvpType </newAvp> [1..100]
  <deletedAvp> AvpType </deletedAvp> [0..100]
</UpdateSessionRequest>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateSessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
      <audit>
        <id>username</id>
        <comment>comment</comment>
      </audit>
      <key>
        <code>UserIdKey</code>
        <primary>true</primary>
        <keyField>
          <code>userId</code>
          <value>uniqueIdentifier</value>
        </keyField>
      </key>
      <newAvp>
        <code>cisco</code>
        <value>sharedsecret</value>
      </newAvp>
      <deletedAvp>
        <code>key</code>
        <value>{}</value>
      </deletedAvp>
    </UpdateSessionRequest>
  </se:Body>
</se:Envelope>

```

## UpdateSessionResponse

### Schema

```

<UpdateSessionResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</UpdateSessionResponse>

```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateSessionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </UpdateSessionResponse>
  </se:Body>
</se:Envelope>
```

## UpdateSubscriberRequest

This API updates a subscriber. An entire subscriber object must be sent. This request also updates credentials. It also allows provisioning initial quota/balance using the CreateBalance element.

### Charging Id

The chargingId (subscriber.getBillingInfo().getChargingId()) is used as a correlation id value between Balance and USuM. The default correlation is the USuM id (generated by the database). If a chargingId is set on the subscriber, the chargingId value becomes the correlation id.




---

#### Note Cannot Change the Charging Id

The UpdateSubscriber API does not/cannot change the chargingId/Balance SubscriberId relationship. To make such a change you must use ChangeBalanceSubscriberId.

---




---

#### Note Balance

The UpdateSubscriber API does allow Balance updates and it can create new Balance. Please see UpdateBalance and CreateBalance for more information.

---




---

#### Note The proper steps to use this API:

1. Call GetSubscriber API to get the subscriber object.
  2. Update returned subscriber object as required.
  3. Call UpdateSubscriber API with updated subscriber object.
- 

### Schema

```
<UpdateSubscriberRequest>
  <audit> AuditType </audit> [0..1] ?
  <subscriber> SubscriberType </subscriber> [1] ?
</UpdateSubscriberRequest>
```

### Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
```

```

<UpdateSubscriberRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
  <audit>
    <id>username</id>
    <comment>comment</comment>
  </audit>
  <subscriber>
    <id>4d4dc3c0f2e1f8ca3871bbd8</id>
    <parentId></parentId>
    <name>
      <fullName>Test User</fullName>
    </name>
    <credential>
      <networkId>70.1.128.7</networkId>
      <password>testuser</password>
      <expirationDate>2011-12-31T23:59:59Z</expirationDate>
    </credential>
    <credential>
      <networkId>70.1.128.9</networkId>
      <password>testuser</password>
    </credential>
    <service>
      <code>SERVICE_CODE</code>
      <enabled>true</enabled>
      <avp>
        <code>AVP_CODE</code>
        <value>AVP_VALUE</value>
      </avp>
      <avp>
        <code>AVP_CODE_2</code>
        <value>AVP_VALUE</value>
      </avp>
    </service>
    <status>ACTIVE</status>
    <version>0</version>
  </subscriber>
</UpdateSubscriberRequest>
</se:Body>
</se:Envelope>

```

## UpdateSubscriberResponse

### Schema

```

<UpdateSubscriberResponse>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</UpdateSubscriberResponse>

```

### Example

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </UpdateSubscriberResponse>
  </se:Body>
</se:Envelope>

```

# Schema Object Definitions



**Note** By default, schema validation is disabled to avoid performance impact. When the schema validation is disabled, the number of AVPs values is not restricted.

To enable the schema validation, set `-Dua.validate.xml=true` in `/etc/broadhop/qns.conf` file.

## AuditKeyType

Super-types	None
Sub-types	None

Name	AuditKeyType
Abstract	no

### Schema

```
<...>
  <id> xsd:string </id> [0..1] ?
  <dataid> xsd:string </dataid> [0..1] ?
  <request> xsd:string </request> [0..1]
  <fromDate> xsd:dateTime </fromDate> [0..1] ?
  <toDate> xsd:dateTime </toDate> [0..1] ?
</...>
```

## AuditType

Super-types	None
Sub-types	None

Name	AuditType
Abstract	no

### Schema

```
<...>
  <id> xsd:string </id> [1] ?
  <comment> xsd:string </comment> [0..1] ?
  <timestamp> xsd:dateTime </timestamp> [0..1]
  <request> xsd:string </request> [0..1] ?
  <dataid> xsd:string </dataid> [0..1] ?
  <data> xsd:string </data> [0..1] ?
  <avp> AvpType </avp> [0..100]
</...>
```



## AvpType

Super-types	None
Sub-types	None

Name	AvpType
Abstract	no

### Schema

```
<...>
  <code> xsd:string </code> [1]
  <value> xsd:string </value> [1]
  <newValue> xsd:string </newValue> [0..1] ?
</...>
```

## BalanceType

Super-types	None
Sub-types	None

Name	BalanceType
Abstract	no
Documentation	A balance is the top-level object that represents a subscriber's balance (amount of time, data, or bandwidth) that can be used.

### Schema

```
<...>
  <code> xsd:string </code> [1]
  <deprovisioned> xsd:boolean </deprovisioned> [0..1] ?
  <depleted> xsd:boolean </depleted> [0..1]
  <billCycle> xsd:integer (1 >= value <= 31) </billCycle> [0..1] ?
  <quota> QuotaType </quota> [0..1000]
  <threshold> ThresholdType </threshold> [0..10]
  <reservation> ReservationType </reservation> [0..1000]
  <startDate> xsd:dateTime </startDate> [0..1]
  <expirationDate> xsd:dateTime </expirationDate> [0..1]
  <totals> TotalsType </totals> [0..1]
  <avp> AvpType </avp> [0..100]
</...>
```

## BaseRequestAuditType

Super-types:	None
Sub-types:	BaseRequestNetworkIdType (by extension)

Name	BaseRequestAuditType
------	----------------------

Abstract	no
----------	----

**Schema**

```
<...>
  <audit> AuditType </audit> [0..1] ?
</...>
```

**BaseRequestNetworkIdType**

Super-types:	BaseRequestAuditType < BaseRequestNetworkIdType (by extension)
Sub-types:	None

Name	BaseRequestNetworkIdType
Abstract	no

**Schema**

```
<...>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1] ?
</...>
```

**BillingInfoType**

Super-types:	None
Sub-types:	None

Name	BillingInfoType
Abstract	no

**Schema**

```
<...>
  <chargingId> xsd:string </chargingId> [0..1] ?
  <ratePlanCode> xsd:string </ratePlanCode> [0..1]
  <avp> AvpType </avp> [0..100]
</...>
```

**CreateBalanceType**

Super-types:	None
Sub-types:	None

Name	CreateBalanceType
Abstract	no

**Schema**

```

<...>
  <code> xsd:string </code> [1] ?
  <quotaCode> xsd:string </quotaCode> [1] ?
  <startDate> xsd:dateTime </startDate> [0..1] ?
  <expirationDate> xsd:dateTime </expirationDate> [0..1] ?
  <initialAmount> xsd:long </initialAmount> [0..1]
  <threshold> ThresholdType </threshold> [0..10]
  <lastRecurringRefresh> xsd:dateTime </lastRecurringRefresh> [0..1] ?
  <billCycle> xsd:integer (1 >= value <= 31) </billCycle> [0..1] ?
  <recurrenceLimit> xsd:integer (1 >= value <= 1000) </recurrenceLimit> [0..1] ?
  <quotaExpirationDate> xsd:dateTime </quotaExpirationDate> [0..1] ?
</...>

```

**CredentialType**

Super-types:	None
Sub-types:	None

Name	CredentialType
Abstract	no

**Schema**

```

<...>
  <networkId> xsd:string </networkId> [1] ?
  <password> xsd:string </password> [0..1]
  <type> xsd:string </type> [0..1] ?
  <description> xsd:string </description> [0..1] ?
  <expirationDate> xsd:dateTime </expirationDate> [0..1] ?
  <avp> AvpType </avp> [0..100]
</...>

```

**CreditType**

Super-types:	None
Sub-types:	None

Name	CreditType
Abstract	no

**Schema**

```

<...>
  <id> xsd:string </id> [1]
  <initialAmount> xsd:long </initialAmount> [1]
  <amount> xsd:long </amount> [1]
  <reservedAmount> xsd:long </reservedAmount> [1]
  <startDate> xsd:dateTime </startDate> [0..1]
  <expirationDate> xsd:dateTime </expirationDate> [0..1]
  <valid> xsd:boolean </valid> [0..1]
  <rolloverPeriodAmount> xsd:integer </rolloverPeriodAmount> [0..1]
  <rolloverPeriodUnits> xsd:string (value comes from list:

```

```
{'Minute'|'Hour'|'Day'|'Week'|'Month'}) </rolloverPeriodUnits> [0..1]
  <rolloverExpirationDate> xsd:dateTime </rolloverExpirationDate> [0..1] ?
  <rolloverTemplateName> xsd:string </rolloverTemplateName> [0..1]
  <avp> AvpType </avp> [0..100]
</...>
```

## DeleteBalanceType

Super-types:	None
Sub-types:	None

Name	DeleteBalanceType
Abstract	no

### Schema

```
<...>
  <code> xsd:string </code> [1] ?
  <quotaCode> xsd:string </quotaCode> [0..1] ?
</...>
```

## EntryType

Super-types:	None
Sub-types:	None

Name	EntryType
Abstract	no

### Schema

```
<...> Start Choice [1]
  <string> xsd:string </string> [1]
  <key> xsd:string </key> [1]
  End Choice
  Start Choice [1]
  <list> ListType </list> [1]
  <linked-list> ListType </linked-list> [1]
  <map> MapType </map> [1]
  <string> xsd:string </string> [1]
  <boolean> xsd:boolean </boolean> [1]
  <int> xsd:integer </int> [1]
  <long> xsd:long </long> [1]
  <date> xsd:dateTime </date> [1]
  <byte-array> xsd:base64Binary </byte-array> [1]
  <value> xsd:anyType </value> [1]
  End Choice
</...>
```

## ExecuteActionResponseType

Super-types:	None
Sub-types:	None

Name	ExecuteActionResponseType
Abstract	no
Documentation	The ExecuteActionResponseType is a catch all object that translates data into a map of key:value pairs. Therefore, from deployment to deployment, an ExecuteActionResponseType can look different.

### Schema

```
<...>
  <responseObject> MapType </responseObject> [0..*]
</...>
```

## ExtendCreditType

Super-types:	None
Sub-types:	None

Name	ExtendCreditType
Abstract	no

### Schema

```
<...>
  <balanceCode> xsd:string </balanceCode> [1]
  <quotaCode> xsd:string </quotaCode> [0..1]
  <creditId> xsd:string </creditId> [0..1] ?
  <add> xsd:boolean </add> [0..1] ?
  <endDate> xsd:dateTime </endDate> [0..1] ?
  <minutes> xsd:integer (value >= 1) </minutes> [0..1] ?
  <amount> xsd:long (value >= 1) </amount> [0..1] ?
</...>
```

## KeyFieldType

Super-types:	None
Sub-types:	None

Name	KeyFieldType
Abstract	no

### Schema

```

<...>
  <code> xsd:string </code> [1]
  <value> xsd:string </value> [1]
</...>

```

## ListType

Super-types:	None
Sub-types:	None

Name	ListType
Abstract	no

### Schema

```

<...> Start Choice [1..*]
  <map> MapType </map> [1]
  <string> xsd:string </string> [1]
  End Choice
</...>

```

## MapType

Super-types:	None
Sub-types:	None

Name	MapType
Abstract	no

### Schema

```

<...>
  <entry> EntryType </entry> [0..*]
</...>

```

## NameType

Super-types:	None
Sub-types:	None

Name	NameType
Abstract	no

### Schema

```

<...>
  <fullName> xsd:string (pattern = [^$&<]*) </fullName> [1] ?
</...>

```

## NotificationType

Super-types:	None
Sub-types:	None

Name	NotificationType
Abstract	no
Documentation	A notification is a holder for subscriber messaging data.

### Schema

```
<...>
  <destination> xsd:string </destination> [1]
  <enabled> xsd:boolean </enabled> [0..1]
  <transport> xsd:string (value comes from list: {'SMS'|'EMAIL'|'APPLE_PUSH'|'GCM'})
</transport> [1]
  <avp> AvpType </avp> [0..100]
</...>
```

## QuotaType

Super-types:	None
Sub-types:	None

Name	QuotaType
Abstract	no

### Schema

```
<...>
  <code> xsd:string </code> [1]
  <credit> CreditType </credit> [0..*]
  <threshold> ThresholdType </threshold> [0..10]
  <nextRefreshDate> xsd:dateTime </nextRefreshDate> [0..1]
  <recurrenceLimit> xsd:integer </recurrenceLimit> [0..1]
  <quotaExpirationDate> xsd:dateTime </quotaExpirationDate> [0..1]
  <totals> TotalsType </totals> [0..1]
  <avp> AvpType </avp> [0..100]
</...>
```

## RefDataBalanceTemplateType

Super-types:	None
Sub-types:	None

Name	RefDataBalanceTemplateType
Abstract	no

Documentation	A ref data balance template is a reference data object defined in Policy Builder that defines an Account Balance for MsBM and also contains any associated Quota and Threshold definitions.
---------------	---

**Schema**

```
<...>
  <code> xsd:string </code> [1]
  <description> xsd:string </description> [0..1]
  <quotaUnits> xsd:string </quotaUnits> [1]
  <refDataQuotaTemplate> RefDataQuotaTemplateType </refDataQuotaTemplate> [0..100]
  <refDataThreshold> RefDataThresholdType </refDataThreshold> [0..100]
</...>
```

## RefDataQuotaTemplateType

Super-types:	None
Sub-types:	None

Name	RefDataQuotaTemplateType
Abstract	no
Documentation	A ref data quota template is a reference data object defined in Policy Builder that defines an Quota for MsBM and also contains any associated Threshold definitions.

**Schema**

```
<...>
  <code> xsd:string </code> [1]
  <description> xsd:string </description> [0..1]
  <amount> xsd:string </amount> [1]
  <priority> xsd:integer </priority> [0..1]
  <refDataThreshold> RefDataThresholdType </refDataThreshold> [0..100]
</...>
```

## RefDataServiceType

Super-types:	None
Sub-types:	None

Name	RefDataServiceType
Abstract	no
Documentation	A ref data service is a reference data object defined in Policy Builder that defines the top level values defined for a service.

**Schema**

```
<...>
  <name> xsd:string </name> [1]
  <code> xsd:string </code> [1]
```



```

<addToSubAccounts> xsd:boolean </addToSubAccounts> [1] ?
<balanceService> xsd:boolean </balanceService> [1] ?
<enabled> xsd:boolean </enabled> [1] ?
<suppressInPortal> xsd:boolean </suppressInPortal> [1] ?
</...>

```

## RefDataThresholdType

Super-types:	None
Sub-types:	None

Name	RefDataThresholdType
Abstract	no
Documentation	A ref data threshold is a reference data object defined in Policy Builder that defines a threshold on either a Balance or Quota in MsBM.

### Schema

```

<...>
  <code> xsd:string </code> [1]
  <amount> xsd:string </amount> [1]
  <thresholdType> xsd:string </thresholdType> [1]
  <group> xsd:string </group> [0..1]
  <triggerOnRemaining> xsd:boolean </triggerOnRemaining> [1]
</...>

```

## RepeatType

Super-types:	None
Sub-types:	None

Name	RepeatType
Abstract	no

### Schema

```

<...>
  <dayOfMonth> xsd:string (pattern = [\-,0-9\*\?LW/]*) </dayOfMonth> [0..1]
  <month> xsd:string (pattern = [\-,0-9\*\?A-Z/]*) </month> [0..1]
  <dayOfWeek> xsd:string (pattern = [\-,0-9\*\?L#/]*) </dayOfWeek> [0..1]
  <year> xsd:string (pattern = [\-,0-9\*/]*) </year> [0..1]
</...>

```

## ReservationType

Super-types:	None
Sub-types:	None

Name	ReservationType
Abstract	no

**Schema**

```
<...>
  <id> xsd:string </id> [1]
  <amount> xsd:long </amount> [1]
  <expirationDate> xsd:dateTime </expirationDate> [1]
  <avp> AvpType </avp> [0..100]
</...>
```

**Response Type**

Super-types:	None
Sub-types:	None

Name	ResponseType
Abstract	no

**Schema**

```
<...>
  <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
  <errorMessage> xsd:string </errorMessage> [1] ?
</...>
```

**ReturnCreditType**

Super-types:	None
Sub-types:	None

Name	ReturnCreditType
Abstract	no

**Schema**

```
<...>
  <id> xsd:string </id> [1]
  <nextRefreshDate> xsd:dateTime </nextRefreshDate> [0..1]
  <balanceRemaining> xsd:long </balanceRemaining> [1]
  <amountCredited> xsd:long </amountCredited> [1]
  <callbackValidityTime> xsd:dateTime </callbackValidityTime> [1]
</...>
```

**ReturnDebitType**

Super-types:	None
Sub-types:	None

Name	ReturnDebitType
Abstract	no

**Schema**

```
<...>
  <nextRefreshDate> xsd:dateTime </nextRefreshDate> [0..1]
  <balanceRemaining> xsd:long </balanceRemaining> [1]
  <amountDebited> xsd:long </amountDebited> [1]
  <callbackValidityTime> xsd:dateTime </callbackValidityTime> [1]
  <exhausted> xsd:boolean </exhausted> [1]
</...>
```

## ScheduleType

Super-types:	None
Sub-types:	None

Name	ScheduleType
Abstract	no
Documentation	A schedule is a holder for a cron-like time structure that customizes service (de)activation. See <a href="#doc_ServicesAndServiceSchedules">Services and Service Schedules</a> for more information.

**Schema**

```
<...>
  <startDate> xsd:dateTime </startDate> [0..1]
  <endDate> xsd:dateTime </endDate> [0..1]
  <state> xsd:string (value comes from list: {'ON'|'OFF'}) </state> [0..1]
  <startTime> xsd:string (pattern = [0-9]{2,2}:[0-9]{2,2}) </startTime> [0..1] ?
  <endTime> xsd:string (pattern = [0-9]{2,2}:[0-9]{2,2}) </endTime> [0..1] ?
  <repeat> RepeatType </repeat> [0..1]
  <enabled> xsd:boolean </enabled> [0..1]
  <avp> AvpType </avp> [0..100]
</...>
```

## SearchType

Super-types:	None
Sub-types:	None

Name	SearchType
Abstract	no

**Schema**

```
<...> Start Choice [0..1]
  <name> NameType </name> [1]
```

```

    <networkId> xsd:string </networkId> [1]
  End Choice
  <avp> AvpType </avp> [0..100]
</...>

```

## ServiceType

Super-types:	None
Sub-types:	None

Name	ServiceType
Abstract	no
Documentation	A service indicates what service CPS should provision to a subscriber and contains any special custom data that helps CPS determine what to provision.

### Schema

```

<...>
  <code> xsd:string </code> [1] ?
  <enabled> xsd:boolean </enabled> [0..1]
  <avp> AvpType </avp> [0..100] ?
  <schedule> ScheduleType </schedule> [0..10] ?
</...>

```

## SessionKeyType

Super-types:	None
Sub-types:	None

Name	SessionKeyType
Abstract	no
Documentation	A sessionKey is used to find sessions in the session cache.

### Schema

```

<...>
  <code> xsd:string </code> [1] ?
  <primary> xsd:boolean </primary> [0..1]
  <keyField> KeyFieldType </keyField> [0..20]
</...>

```

## SessionType

Super-types:	None
Sub-types:	None

Name	SessionType
------	-------------

Abstract	no
Documentation	Similar to the <a href="#">ExecuteActionResponseType</a> a SessionType is a map of key:value pairs. CPS is very flexible and allows for many session configurations. Therefore, all data is structured as a map of key:value pairs. Therefore, from deployment to deployment, a session can look different.

**Schema**

```
<...>
  <sessionKey> SessionKeyType </sessionKey> [1]
  <sessionObject> MapType </sessionObject> [0..*]
</...>
```

## SsidType

Super-types:	None
Sub-types:	None

Name	SsidType
Abstract	no

**Schema**

```
<...>
  <ssidKey> xsd:string </ssidKey> [1]
  <ssid> xsd:string </ssid> [0..1]
  <accessType> xsd:string </accessType> [0..1]
  <authType> xsd:string </authType> [0..1]
  <username> xsd:string </username> [0..1]
  <password> xsd:string </password> [0..1]
  <loginUrl> xsd:string </loginUrl> [0..1]
  <configUrl> xsd:string </configUrl> [0..1]
  <verificationCertUrl> xsd:string </verificationCertUrl> [0..1]
  <configMessage> xsd:string </configMessage> [0..1]
  <portalFailMessage> xsd:string </portalFailMessage> [0..1]
  <unmanagedVpnConnectPrompt> xsd:string </unmanagedVpnConnectPrompt> [0..1]
  <unmanagedVpnDisconnectPrompt> xsd:string </unmanagedVpnDisconnectPrompt> [0..1]
</...>
```

## SubscriberSSIDType

Super-types:	None
Sub-types:	None

Name	SubscriberSSIDType
Abstract	no

**Schema**

```

<...>
  <subscriberId> xsd:string </subscriberId> [1]
  <ssidList> SsidType </ssidList> [0..40]
</...>

```

## SubscriberType

Super-types:	UserType < SubscriberType (by extension)
Sub-types:	None

Name	SubscriberType
Abstract	no

### Schema

```

<...>
  <id> xsd:string </id> [0..1] ?
  <parentId> xsd:string </parentId> [0..1] ?
  <name> NameType </name> [0..1]
  <authType> xsd:string (value comes from list: {'NONE'|'BASIC'|'DIGEST'|'SIMPLE'})
</authType> [0..1]
  <authUsername> xsd:string </authUsername> [0..1]
  <authPassword> xsd:string </authPassword> [0..1]
  <credential> CredentialType </credential> [1..20]
  <service> ServiceType </service> [0..20]
  <notification> NotificationType </notification> [0..10]
  <session> SessionType </session> [0..100]
  <balance> BalanceType </balance> [0..100]
  <status> StatusType </status> [1]
  <startDate> xsd:dateTime </startDate> [0..1]
  <endDate> xsd:dateTime </endDate> [0..1]
  <role> xsd:string (value comes from list:
{'READ_ALL'|'READ_SELF'|'WRITE_ALL'|'WRITE_SELF'}) </role> [0..1]
  <externalId> xsd:string (pattern = .{0,100}) </externalId> [0..1] ?
  <billingInfo> BillingInfoType </billingInfo> [0..1]
  <avp> AvpType </avp> [0..100]
  <version> xsd:integer </version> [0..1] ?
  <realm> xsd:string </realm> [0..1] ?
  <subAccount> UserType </subAccount> [0..1000] ?
  <createBalance> CreateBalanceType </createBalance> [0..10]
</...>

```

## ThresholdType

Super-types:	None
Sub-types:	None

Name	ThresholdType
Abstract	no

### Schema

```

<...>
  <code> xsd:string </code> [1]

```

```

    <amount> xsd:long </amount> [1]
    <quotaCode> xsd:string </quotaCode> [0..1]
    <type> xsd:string (value comes from list:
    {'Percentage'|'Bytes'|'Kilobytes'|'Megabytes'|'Gigabytes'|'Other'}) </type> [1]
    <group> xsd:string </group> [0..1]
    <breached> xsd:boolean </breached> [0..1] ?
    <subscriberSpecific> xsd:boolean </subscriberSpecific> [1]
  </...>

```

## TotalsType

Super-types:	None
Sub-types:	None

Name	TotalsType
Abstract	no
Documentation	The totals object is a convenience field showing a quick summary of the debited, reserved, and available amounts for the balance.

### Schema

```

<...>
  <balance> xsd:decimal </balance> [0..1]
  <reserved> xsd:decimal </reserved> [0..1]
  <debited> xsd:decimal </debited> [0..1]
</...>

```

## UserType

Super-types:	None
Sub-types:	SubscriberType (by extension)

Name	UserType
Abstract	no

### Schema

```

<...>
  <id> xsd:string </id> [0..1] ?
  <parentId> xsd:string </parentId> [0..1] ?
  <name> NameType </name> [0..1]
  <authType> xsd:string (value comes from list: {'NONE'|'BASIC'|'DIGEST'|'SIMPLE'})
</authType> [0..1]
  <authUsername> xsd:string </authUsername> [0..1]
  <authPassword> xsd:string </authPassword> [0..1]
  <credential> CredentialType </credential> [1..20]
  <service> ServiceType </service> [0..20]
  <notification> NotificationType </notification> [0..10]
  <session> SessionType </session> [0..100]
  <balance> BalanceType </balance> [0..100]
  <status> StatusType </status> [1]
  <startDate> xsd:dateTime </startDate> [0..1]

```

```

    <endDate> xsd:dateTime </endDate> [0..1]
    <role> xsd:string (value comes from list:
{'READ_ALL'|'READ_SELF'|'WRITE_ALL'|'WRITE_SELF'}) </role> [0..1]
    <externalId> xsd:string (pattern = .{0,100}) </externalId> [0..1] ?
    <billingInfo> BillingInfoType </billingInfo> [0..1]
    <avp> AvpType </avp> [0..100]
</...>

```

## VoucherBatchKeyType

Super-types:	None
Sub-types:	None

Name	VoucherBatchKeyType
Abstract	no

### Schema

```

<...>
  <code> xsd:string </code> [1] ?
  <pin> xsd:string </pin> [0..1] ?
  <maskChars> xsd:string </maskChars> [0..1] ?
</...>

```

## VoucherKeyType

Super-types:	None
Sub-types:	None

Name	VoucherKeyType
Abstract	no

### Schema

```

<...> Start Choice [1]
  <voucherCode> xsd:string </voucherCode> [1]
  <networkId> xsd:string </networkId> [1] ?
  <voucherLocationKey> VoucherLocationKeyType </voucherLocationKey> [1]
  <voucherBatchKey> VoucherBatchKeyType </voucherBatchKey> [1]
End Choice
  <skip> xsd:integer (value >= 0) </skip> [0..1] ?
  <limit> xsd:integer (1 >= value <= 1000) </limit> [0..1] ?
</...>

```

## VoucherLocationKeyType

Super-types:	None
Sub-types:	None



Name	VoucherLocationKeyType
Abstract	no

**Schema**

```
<...>
  <locationCode> xsd:string </locationCode> [1] ?
  <active> xsd:boolean </active> [0..1]
</...>
```

## VoucherType

Super-types:	None
Sub-types:	None

Name	VoucherType
Abstract	no

**Schema**

```
<...>
  <code> xsd:string </code> [1] ?
  <pin> xsd:string </pin> [0..1] ?
  <subscriberId> xsd:string </subscriberId> [0..1] ?
  <maxConcurrentSessions> xsd:integer </maxConcurrentSessions> [1] ?
  <duration> xsd:long </duration> [1] ?
  <durationMeasure> xsd:string (value comes from list:
{'Minutes'|'Hours'|'Days'|'Weeks'|'Months'}) </durationMeasure> [1] ?
  <serviceCode> xsd:string </serviceCode> [1] ?
  <expirationDate> xsd:dateTime </expirationDate> [0..1] ?
  <locationCode> xsd:string </locationCode> [0..1] ?
  <timeQuota> xsd:long </timeQuota> [0..1] ?
  <timeMeasure> xsd:string </timeMeasure> [0..1] ?
  <volumeQuota> xsd:string </volumeQuota> [0..1] ?
  <volumeMeasure> xsd:string </volumeMeasure> [0..1] ?
  <balanceCode> xsd:string </balanceCode> [0..1] ?
  <quotaCode> xsd:string </quotaCode> [0..1] ?
</...>
```

## StatusType

Super-types:	xsd:string < StatusType (by restriction)
Sub-types:	None

Name	StatusType
Content	<ul style="list-style-type: none"> <li>'string' super type was not found in this schema. Its facets could not be printed out.</li> <li>value comes from list: {'ACTIVE' 'DELETED' 'SUSPENDED' 'INACTIVE'}</li> </ul>

