



Requests, Responses, and Schema

- [API Requests and Responses, on page 1](#)

API Requests and Responses



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

Single API Request

CreateSubscriberRequest

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

Schema

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

Example

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
-H "Content-Type: text/xml; charset=utf-8" \
-H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/CreateSubscriberRequest\"" \
-d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv=
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:CreateSubscriberRequest>
      <subscriber>
        <name>
          <fullName>LaasUser_41</fullName>
```

```

        </name>
        <credential>
            <networkId>uniqueIdentifier</networkId>
            <password>password</password>
        </credential>
        <service>
            <code>Plan0</code>
            <enabled>true</enabled>
        </service>
        <avp>
            <code>Called-Station-Id</code>
            <value>LaasUser_41</value>
        </avp>
        <avp>
            <code>CIRCLE_CODE</code>
            <value>WB</value>
        </avp>
        <avp>
            <code>PINCODE</code>
            <value>405041</value>
        </avp>
        <status>ACTIVE</status>
    </subscriber>
</typ:CreateSubscriberRequest>
</soapenv:Body>
</soapenv:Envelope>'

```

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>

```

GetSubscriberRequest

This API retrieves a subscriber from the USuM database.



Note Success Response

By default, the GetSubscriber API returns error code 15 (failure) if no subscriber is found. The `success.response.nonexistent.user` property has been introduced in Ops-Center. This property allows cnAAA can to control the behavior of the GetSubscriber API response. If the value is set to true, the GetSubscriber API returns error code 0 even when 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:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
-H "Content-Type: text/xml; charset=utf-8" \
-H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/GetSubscriberRequest\"" \
-d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:GetSubscriberRequest>
      <typ:networkId>SVI_Subscriber_3300424</typ:networkId>
    </typ:GetSubscriberRequest>
  </soapenv:Body>
</soapenv:Envelope>'
```



Note This existing GetSubscriberRequest is used in a new Dual-Write Proxy logic. The UpdateSubscriber logic is enhanced to require specific <id> and <version> injection for data consistency.

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><id>00cd69d02edd246467628f40</id><credential>
<networkId></networkId><password></password></credential><credential><networkId>
]]></networkId><type>Called-Station-Id</type></credential><service><code>Plan0</code><enabled>true</enabled>
</service><status>ACTIVE</status><avp><code>Called-Station-Id</code><value>IN001_3300424</value></avp><avp><code>CIRCLE_CODE
</code><value>DL</value></avp><avp><code>PINCODE</code><value>395004</value></avp><avp><code>SprKeyHash</code><value>2</value>
</avp><version>0</version>
</subscriber></GetSubscriberResponse></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.



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:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
-H "Content-Type: text/xml; charset=utf-8" \
-H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/UpdateSubscriberRequest\"" \
-d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:UpdateSubscriberRequest>
      <subscriber>
        <id>001000001ca50459681090ee</id>
        <name>
          <fullName>uniqueidentifier</fullName>
        </name>
        <credential>
          <networkId>uniqueidentifier</networkId>
          <password>password</password>
        </credential>
        <service>
          <code>Plan0-FTTX</code>
          <enabled>true</enabled>
        </service>
        <status>ACTIVE</status>
        <avp>
          <code>CIRCLE_CODE</code>
          <value>WB</value>
        </avp>
        <version>0</version>
      </subscriber>
    </typ:UpdateSubscriberRequest>
  </soapenv:Body>
</soapenv:Envelope>'
```



Note This existing UpdateSubscriberRequest is used in a new Dual-Write Proxy logic. The UpdateSubscriber logic is enhanced to require specific <id> and <version> injection for data consistency.

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>
```

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

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
  -H "Content-Type: text/xml; charset=utf-8" \
  -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/DeleteSubscriberRequest\"" \
  -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:DeleteSubscriberRequest>
      <networkId>uniqueidentifier</networkId>
      <hardDelete>true</hardDelete>
    </typ:DeleteSubscriberRequest>
  </soapenv:Body>
</soapenv:Envelope>'
```

```

        </typ:DeleteSubscriberRequest>
    </soapenv:Body>
</soapenv: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>

```

Bulk API Request

CreateBulkSubscribersRequest

The CreateBulkSubscribers API enables the creation of multiple subscribers in one request. It includes subscriber details like credentials and services, ensuring correct provisioning.

Schema:

```

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

```

Example:

```

curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
  -H "Content-Type: text/xml; charset=utf-8" \
  -H "SOAPAction:
  \"http://broadhop.com/unifiedapi/soap/types/CreateBulkSubscribersRequest\" \" \" \
  -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:CreateBulkSubscribersRequest>
      <subscriber>
        <name>
          <fullName>uniqueIdentifier</fullName>
        </name>
        <credential>
          <networkId>uniqueIdentifier</networkId>
          <password>password</password>
        </credential>
        <service>
          <code>Plan0</code>

```

```

        <enabled>true</enabled>
      </service>
    <avp>
      <code>Called-Station-Id</code>
      <value>LaasUser_31</value>
    </avp>
    <avp>
      <code>CIRCLE_CODE</code>
      <value>WB</value>
    </avp>
    <avp>
      <code>PINCODE</code>
      <value>405061</value>
    </avp>
    <status>ACTIVE</status>
  </subscriber>
</subscriber>
<subscriber>
  <name>
    <fullName>uniqueIdentifier</fullName>
  </name>
  <credential>
    <networkId>uniqueIdentifier</networkId>
    <password>password</password>
  </credential>
  <service>
    <code>Plan0</code>
    <enabled>true</enabled>
  </service>
  <avp>
    <code>Called-Station-Id</code>
    <value>LaasUser_32</value>
  </avp>
  <avp>
    <code>CIRCLE_CODE</code>
    <value>WB</value>
  </avp>
  <avp>
    <code>PINCODE</code>
    <value>405061</value>
  </avp>
  <status>ACTIVE</status>
</subscriber>
</subscriber>
<subscriber>
  <name>
    <fullName>uniqueIdentifier</fullName>
  </name>
  <credential>
    <networkId>uniqueIdentifier</networkId>
    <password>password</password>
  </credential>
  <service>
    <code>Plan0</code>
    <enabled>true</enabled>
  </service>
  <avp>
    <code>Called-Station-Id</code>
    <value>LaasUser_33</value>
  </avp>
  <avp>
    <code>CIRCLE_CODE</code>
    <value>WB</value>
  </avp>
  <avp>
    <code>PINCODE</code>

```

```

        <value>405061</value>
      </avp>
      <status>ACTIVE</status>
    </subscriber>
  </typ:CreateBulkSubscribersRequest>
</soapenv:Body>
</soapenv:Envelope>'

```

CreateBulkSubscribersResponse

Schema:

```

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

```

Example:

```

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

```

GetSubscribersRequest

The GetSubscribers API retrieves subscriber details based on specified network IDs. It allows for querying multiple subscribers in a single request to access the information.

Schema:

```

<GetSubscribersRequest>
  <audit> AuditType </audit> [0..1] ?
  <networkId> xsd:string </networkId> [1..1000] ?
  <returnSessions> xsd:boolean </returnSessions> [0..1] ?
  <includeExpiredData> xsd:boolean </includeExpiredData> [0..1] ?
  <excludeReservationsFromCreditTotal> xsd:boolean </excludeReservationsFromCreditTotal> [0..1] ?
</GetSubscribersRequest>

```

Example:

```

curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
  -H "Content-Type: text/xml; charset=utf-8" \
  -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/GetSubscribersRequest\"" \
  -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>

```

```

<soapenv:Body>
  <typ:GetSubscribersRequest>
    <typ:networkId>SVI_Subscriber_3300422</typ:networkId>
    <typ:networkId>SVI_Subscriber_3300423</typ:networkId>
    <typ:networkId>SVI_Subscriber_3300424</typ:networkId>
  </typ:GetSubscribersRequest>
</soapenv:Body>
</soapenv:Envelope>'

```

GetSubscribersResponse

Schema:

```

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

```

Example:

```

<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"><se:Body><list>
  <GetSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types"><errorCode>
    0</errorCode><errorMessage>Request completed successfully</errorMessage><subscriber>
    <subscriber><id>00d69c018cab7b49680ab385</id><credential><networkId>
      <![CDATA[SVI_Subscriber_3300422]]></networkId><password><![CDATA[{SHA}
      Yk6GaDy0tor24Aigh7ts4FirGPY/PwE/Pz91Pw==]]>
    </password></credential><credential><networkId><![CDATA[INJKNORAXXXXTW0013ENBESR001_3300422]]>
    </networkId><type>Called-Station-Id</type></credential><service>
      <code>Plan0</code><enabled>true</enabled>
    </service><status>ACTIVE</status><avp><code>Called-Station-Id</code>
      <value>INJKNORAXXXXTW0013ENBESR001_3300422</value></avp><avp>
      <code>CIRCLE_CODE</code><value>DL</value>
    </avp><avp><code>PINCODE</code><value>385006</value></avp><avp><code>SprKeyHash</code><value>1</value>
    </avp><version>0</version></subscriber><subscriber><id>00e69c018cab7b49680ab385</id><credential><networkId>
      <![CDATA[SVI_Subscriber_3300423]]></networkId><password><![CDATA[{SHA}a6CJLxyyf2h2KzmzgWxCehNixGRIV0gTF3c/Pw==]]>
    </password></credential><credential><networkId><![CDATA[INJKNORAXXXXTW0013ENBESR001_3300423]]>
    </networkId><type>Called-Station-Id</type></credential><service><code>Plan0</code><enabled>true</enabled></service>
    <status>ACTIVE</status><avp><code>Called-Station-Id</code><value>INJKNORAXXXXTW0013ENBESR001_3300423</value></avp>
    <avp><code>CIRCLE_CODE</code><value>KO</value></avp><avp><code>PINCODE</code><value>385007</value></avp><avp><code>
      SprKeyHash</code><value>0</value></avp><version>0</version></subscriber><subscriber><id>00d69c01e2171601680ab385
    </id><credential><networkId><![CDATA[SVI_Subscriber_3300424]]></networkId><password>
      <![CDATA[{SHA}qFkpMzgCrSyVepwSOitG5JnEMDczOj90Pz9XPw==]]></password></credential><credential><networkId>
      <![CDATA[INJKNORAXXXXTW0013ENBESR001_3300424]]></networkId><type>Called-Station-Id</type></credential><service>
      <code>Plan0</code><enabled>true</enabled></service><status>ACTIVE</status><avp><code>Called-Station-Id</code>
      <value>INJKNORAXXXXTW0013ENBESR001_3300424</value></avp><avp><code>CIRCLE_CODE</code><value>AP</value></avp><avp>
      <code>PINCODE</code><value>385008</value></avp><avp><code>SprKeyHash</code><value>2</value></avp><version>0</version>
    </subscriber><subscriber><id>00f69c01e2171601680ab385</id><credential><networkId></GetSubscribersResponse></list>
  </se:Body></se:Envelope>

```

UpdateSubscribersRequest

The UpdateSubscribers API modifies subscriber details, including credentials and services, for active subscribers in the system. It ensures updates are reflected in the subscriber database.

Schema:

```

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

```

Example:

```

curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
  -H "Content-Type: text/xml; charset=utf-8" \
  -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/UpdateSubscribersRequest\"" \
  -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:UpdateSubscribersRequest>
      <subscriber>
        <id>00200000602b083e680f3b6f</id>
        <name>
          <fullName>uniqueIdentifier</fullName>
        </name>
        <credential>
          <networkId>uniqueIdentifier</networkId>
          <password>password</password>
        </credential>
        <service>
          <code>Plan0-FTTX</code>
          <enabled>true</enabled>
        </service>
        <status>ACTIVE</status>
        <avp>
          <code>CIRCLE_CODE</code>
          <value>MU</value>
        </avp>
        <avp>
          <code>SprKeyHash</code>
          <value>0</value>
        </avp>
        <version>0</version>
      </subscriber>
      <subscriber>
        <id>00300000602b083e680f3b6f</id>
        <name>
          <fullName>uniqueIdentifier</fullName>
        </name>
        <credential>
          <networkId>uniqueIdentifier</networkId>
          <password>password</password>
        </credential>
        <service>
          <code>Plan0-FTTX</code>
          <enabled>true</enabled>
        </service>
        <status>ACTIVE</status>
        <avp>
          <code>CIRCLE_CODE</code>
          <value>MU</value>
        </avp>
        <avp>
          <code>SprKeyHash</code>
          <value>1</value>
        </avp>
        <version>0</version>
      </subscriber>
      <subscriber>
        <id>0010000065805166680f3bde</id>
        <name>
          <fullName>uniqueIdentifier</fullName>

```

```

</name>
<credential>
  <networkId>uniqueIdentifier</networkId>
  <password>password</password>
</credential>
<service>
  <code>Plan0-FTTX</code>
  <enabled>true</enabled>
</service>
<status>ACTIVE</status>
<avp>
  <code>CIRCLE_CODE</code>
  <value>MU</value>
</avp>
<avp>
  <code>SprKeyHash</code>
  <value>2</value>
</avp>
<version>0</version>
</subscriber>
</typ:UpdateSubscribersRequest>
</soapenv:Body>
</soapenv:Envelope>'

```

UpdateSubscribersResponse

Schema:

```

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

```

Example:

```

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

```

DeleteSubscribersRequest

The DeleteSubscribers API removes subscribers from the system based on specified network IDs. It supports hard deletions, offering flexible option for managing subscriber data.

Schema:

```

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

```

```
<networkId> xsd:string </networkId> [1..1000] ?
<hardDelete> xsd:boolean </hardDelete> [0..1] ?
</DeleteSubscribersRequest>
```

Example:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
-H "Content-Type: text/xml; charset=utf-8" \
-H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/DeleteSubscribersRequest\"" \
-d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:DeleteSubscribersRequest>
      <networkId>uniqueidentifier</networkId>
      <networkId>uniqueidentifier</networkId>
      <networkId>uniqueidentifier</networkId>
      <hardDelete>true</hardDelete>
    </typ:DeleteSubscribersRequest>
  </soapenv:Body>
</soapenv:Envelope>'
```

DeleteSubscribersResponse

Schema:

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

Example:

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

Refresh profile API implementation in cnAAA

RefreshSubscriberProfileRequest

The Refresh Profile API in cnAAA accepts a mac-address. This API triggers the policy engine. The policy engine queries the SPR, fetches the subscriber record, and processes CoA if the session is available in CDL.

Schema:

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

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:Body><RefreshSubscriberProfileRequest xmlns="http://broadhop.com/unifiedapi/soap/types">

  <networkId>uniqueidentifier</networkId>
</RefreshSubscriberProfileRequest>
</se:Body>
</se:Envelope>
```

RefreshSubscriberProfileResponse**Schema:**

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

Example:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:Body>
<RefreshSubscriberProfileResponse>
  <errorCode>0</errorCode>
  <errorMessage>Request completed successfully</errorMessage>
  <sessionId>192.168.175.148:00012</sessionId>
  <sessionId>192.168.175.148:00013</sessionId>
  <sessionId>192.168.175.148:00014</sessionId>
</RefreshSubscriberProfileResponse>
</se:Body>
</se:Envelope>
```

Delete session API implementation in cnAAA

The Delete Session API in cnAAA accepts a mac-address. This API triggers the policy engine to delete the CDL sessions of that subscriber.

DeleteSubscriberSessionRequest**Schema:**

```
<DeleteSubscriberSessionRequest>
  <networkId> xsd:string </networkId> [1] ?
  <subscriberId> xsd:string </subscriberId> [1] ?
</DeleteSubscriberSessionRequest>
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:Body>
<DeleteSubscriberSessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
  <networkId>uniqueidentifier</networkId>
  <subscriberId>00c5000042e99b6068bed7ce</subscriberId>
</DeleteSubscriberSessionRequest>
</se:Body>
</se:Envelope>
```

DeleteSubscriberSessionResponse

Schema:

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

Example:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteSubscriberSessionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
      <sessionCount>3</sessionCount>
      <sessionId>192.168.175.148:00012</sessionId>
      <sessionId>192.168.175.148:00013</sessionId>
      <sessionId>192.168.175.148:00014</sessionId>
    </DeleteSubscriberSessionResponse>
  </se:Body>
</se:Envelope>
```